Griffith School of Engineering

Griffith University
6007ENG – Industry Affiliates Program

Development of a Low Cost EEG Based System for Enhancing Communication in Patients with Locked-In Syndrome

Thomas Ireland
S2783962

Trimester 2, 2018

Griffith University Internal Project

Andrew Seagar

A report submitted in partial fulfilment of the degree of Bachelor of Engineering with Honours.

Griffith UNIVERSITY

I - Executive Summary

This project was developed as an internal project for the Industry Affiliates Program at Griffith University. It documents the process of developing a low-cost brain computer interface which allows people the ability to have basic communication solely via their brain activity.

To do this, background research was performed on brain activity, brain-computer interfaces (BCI), and the resources and methodology required to develop a working prototype. A literature review was also performed to identify works previously done, and to determine the relevance of this project. The Muse Headband was chosen as a low-cost consumer device which measures the electrical activity of a person's brain (Electroencephalography/EEG), and transmits the data via bluetooth to a computer. It was identified that different mental activities produce identifiable EEG data characteristics, which can be fed into a machine classifier in order to predict the classification of future data sets. The project objectives were defined as developing a working prototype system to allow a 'Yes' or 'No' classification from EEG, to determine the subject's emotional and mental states in real from their EEG characteristics, and also to compare the performance and capabilities with other high cost systems that have been developed.

Testing verified that the Muse Headband could accurately measure EEG data, and with filtering and processing, the data fed into a classifier could accurately predict future data. A software prototype was built around this, visually and auditorily displaying this communication upon another user prompting the software. Then, testing was performed to identify the EEG feature correlates with certain mental and emotional states. Definite correlations were found, which were then used to implement state classification into the prototype software.

An analysis of the capabilities and performance of this low-cost system in comparison to high-cost systems revealed that the Muse Headband was capable of accurately measuring data, and being used for research purposes or BCI applications. The high-cost systems featured much more extensive measurement capabilities, and also implemented complex signal processing and filtering, which allowed for highly sophisticated BCI functionality. One system could allow a user to spell sentences with a high accuracy and speed, and another could predict a subject's limb positions from their EEG.

The project was largely a success, however a main limitation was due to the focus on the low-level filtering and processing of the EEG data. If instead, already developed tool kits and coding libraries were used, more time could have been spent developing further BCI functionality. The project confirms that research and development applications are no longer limited to EEG measurement systems costing in excess of tens of thousands of dollars. EEG

measurement can be performed with devices costing just hundreds of dollars, allowing accessibility to hobby and independent researchers.

II - Acknowledgements

I would like to take this opportunity to acknowledge and thank the following people, without whom I would not have been able to complete this project:

My girlfriend, Dell, for putting up with me while I've been glued to my computer working the last couple of months. Helping me with the household chores, cooking me food when I've been too worn out to cook, and supporting me in everything I choose to do.

My mother, father and grandparents, for always being there for me, believing in me and supporting me 100% in every way they can. Also for pushing me to go back to Uni when I thought I wanted to pave my way through life and succeed without needing a degree.

My awesome group of friends, for respecting the time that I've been so focused on working hard, and for helping me celebrate after I'm finished like I know they will.

My fellow peers at Griffith for putting up with my distractions when I haven't been able to concentrate and just wanted to chat, inspiring me with their work and helping whenever I needed.

All the academic staff at Griffith who have ever taught me and been there to answer any questions I had, and for educating me well enough to be able to complete a decent project.

Finally, my academic supervisor Dr Andrew Seagar, for supervising my project and providing feedback and support every step of the way.

III - Table of Contents

1 - Introduction

1.1 - Report Structure

This report is organised as follows. Initially, relevant background information is presented and the aims, objectives and outcomes of the project are outlined. Second, a review of the published literature identifies previous works performed and the relation of this study. Next, the design specifications and process to produce the final prototype is detailed. Results from testing of the prototype and discussions follow, with an analysis of its performance and capabilities in comparison to other systems. Recommendations and conclusions are then made, followed by the references and appendices.

1.2 - Project Background

1.2.1 - Locked-in Syndrome

Locked-in syndrome (LIS) is a rare condition characterized by complete paralysis of the body, with the exception of vertical eye movements and blinking [1]. Some patients may retain the ability to move certain facial muscles, whereas those with complete locked-in syndrome (CLIS) also suffer paralysis of the eyes. The cause of LIS is due to damage to the lower portion of the brain and the brainstem, with the upper portion of the brain remaining intact. Possible specific causes can include poisoning, stroke, circulatory disease, nerve damage and trauma. [2]

Typically, LIS patients never recover motor function, and remain in full paralysis for their entire lifetime. Issues can arise with difficulties in prognosis, when the patient is assumed to be in a vegetative state. LIS patients however retain cognitive function, awareness and their ability to perceive the world through their sensory inputs [3]. As reliance on caregivers is required for human needs such as personal hygiene, meals and entertainment, the quality of life for a LIS patient is limited by their ability to communicate with those around them.

1.2.2 - Communication Methods for LIS Patients

The traditional methods for allowing paralysis patients to communicate are via assistive computer interface technologies. These allow patients to form words and control a computer cursor with very limited bodily movement, such as with eye tracking technologies. Recently, brain-computer interfaces (BCI) have allowed patients to interact with software purely based on thought processes alone, with testing proving successful attempts for answering questions and operating computer software. [4]

1.2.3 - Electroencephalography (EEG)

The most researched and utilized BCI systems measure the brain's activity through electroencephalography (EEG). The brain operates by firing neurons which receive, process and transmit information via electrical signals. When a large group of neurons fire simultaneously, the resultant flow of electric charge excites other neurons [5]. This ionic current causes voltage fluctuations, which can be measured via EEG. EEG is performed non-invasively, by placing electrodes on the surface of the scalp and measuring the voltage differences between a reference and target[3], [6].

1.2.4 - EEG Measurement

EEG measures brain activity closest to the scalp the most accurately due to the placement of the electrodes directly on the scalp surface. The data collected is a measure of the voltage oscillations over a period of time. The magnitude of a scalp measured EEG signal is typically within the range of 10uV to 100 uV [7]. Thus, due to the required sensitivity of the measurement, electromagnetic noise can be introduced from outside sources such as mains voltage.

EEG, when compared to other methods of measuring brain activity such as Positron Emission Tomography (PET) and Magnetic Resonance Imagery (MRI), has many advantages. These include a lower cost of hardware due to the simpler measuring method [8], data measurement resolution of up to 20,000 Hz [9], silent operation, non-invasive measurement and the ability to detect brain process without requiring a linked motor response or reaction [10].

1.2.5 - EEG Devices

The main limitation for current BCI's arise from the high cost of medical grade EEG equipment, limiting use to those who have the means to afford it and also limiting development and research teams to those with enough resources to fund equipment cost. Consumer grade EEG devices have recently been introduced to the market, mainly for recreational purposes. The Muse Headband (Figure 1) for example is primarily sold as a relaxation monitor to aid with meditation. However, the developers of these devices also include fully featured and backed support for research and development purposes. This allows independent and hobby researchers and developers to access EEG data and expand the scope of EEG applications. As the Muse Headband is non-invasive, meaning that electrode pads can be placed on the surface of the scalp, and as it runs from an internal battery with a wireless communication protocol, EEG data can be monitored and viewed with very easy setup and portability.

Figure 1. The Muse Headband by Interaxon. [11]

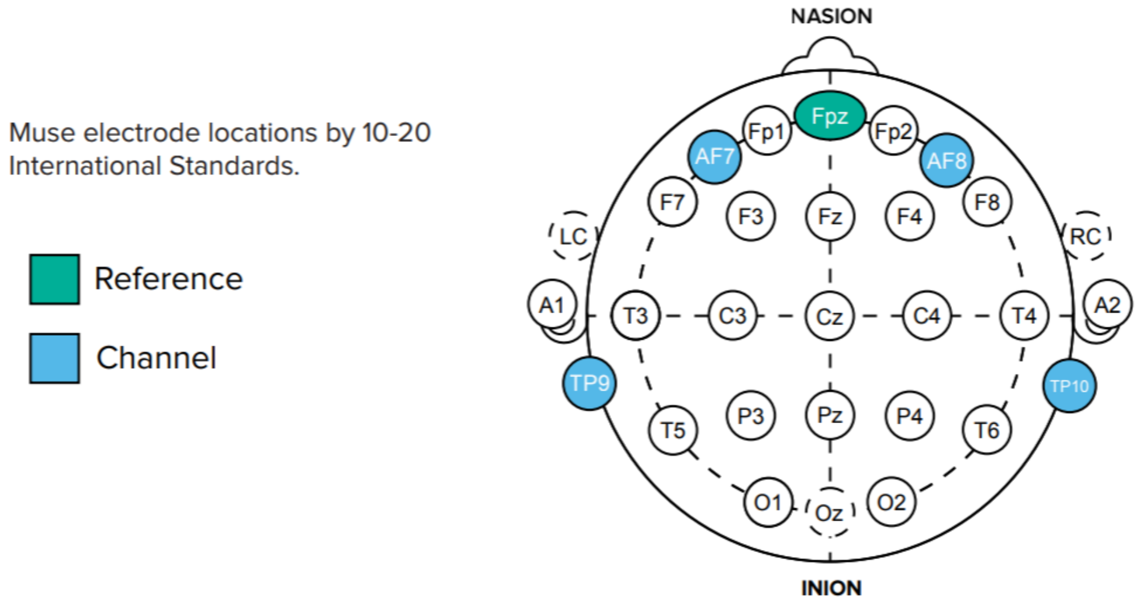


Figure 2. Muse Headband Electrode Locations. [11]

The Muse Headband contains seven total sensors: Three reference sensors, two forehead sensors and two sensors behind the ears. The sensor locations are shown above in Figure 2. The 10-20 International Standards for EEG electrode placements were developed in order to standardise testing and research methods. The AF designation corresponds to the area between the prefrontal and frontal lobes, while the TP designation corresponds to the area between the temporal and parietal lobes. The frontal lobe of the brain is responsible for various functions such as higher level cognition, reasoning, motor skills and language expression. The parietal lobe processes tactile sensory function. The somatosensory cortex is also located in this lobe and is responsible for the processing of this sensory data. The temporal lobe processes auditory signals, short-term memory and also emotional association [12]. Figure 3 below displays the anatomy and functional areas of the brain, with the location and functions of these lobes displayed.

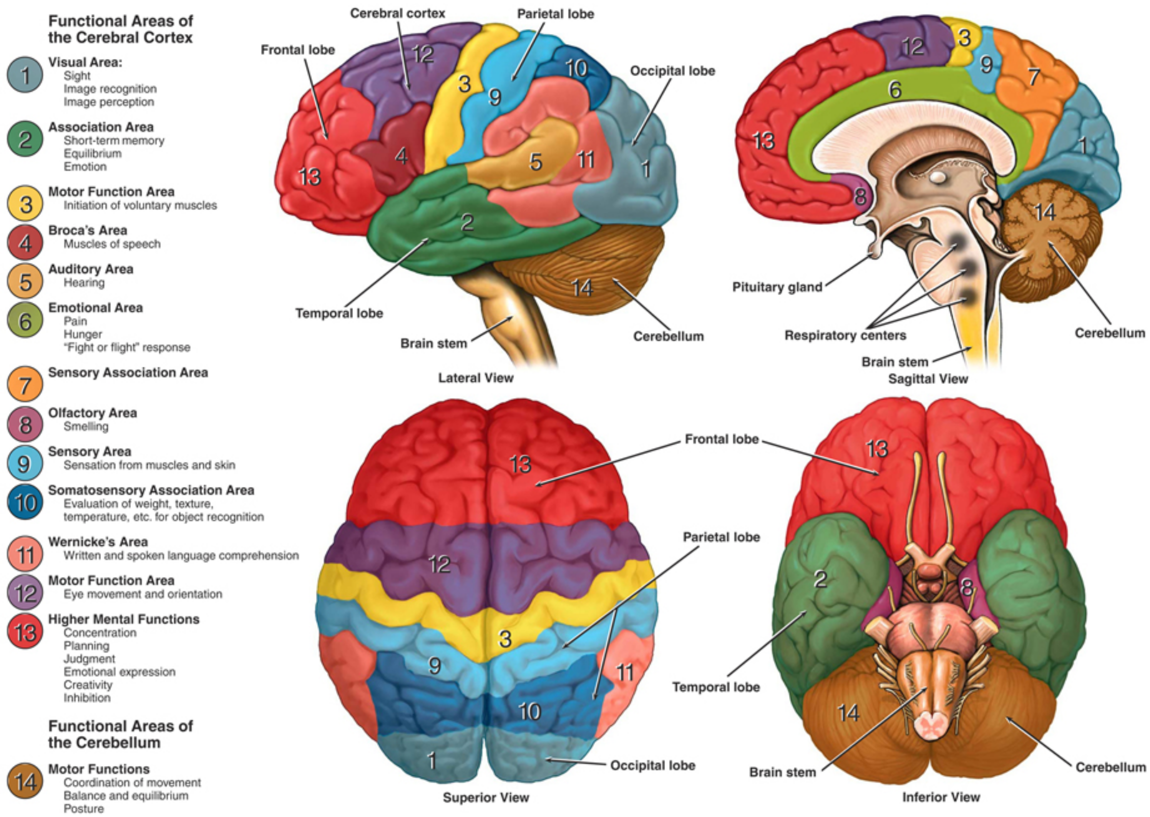


Figure 3. Anatomy and Functional Areas of the Brain. [13]

The headband connects via Bluetooth to either a mobile device or computer and can transmit real-time, four channel EEG data for recording, analysis and processing. Using the software development kit provided by Interaxon, events such as jaw clenching, eye blinks, and in-built signal processing to determine dominant frequency domain states can also be measured.

1.2.6 - EEG Limitations

Artifacts are disturbances in a signal not relevant to the important data. Physiological artifacts are those caused by obscurities of the brain EEG due to electrical impulses from the body. "Physiological artifacts may include cardiac, pulse, respiratory, sweat, glossokinetic, eye movement (blink, lateral rectus spikes from lateral eye movement), and muscle and movement artifacts." [14]

Figure 4 below is an example of artifacting due to rapid eye movement. Spiked discharges can be visually observed across all channels during the eye movements.

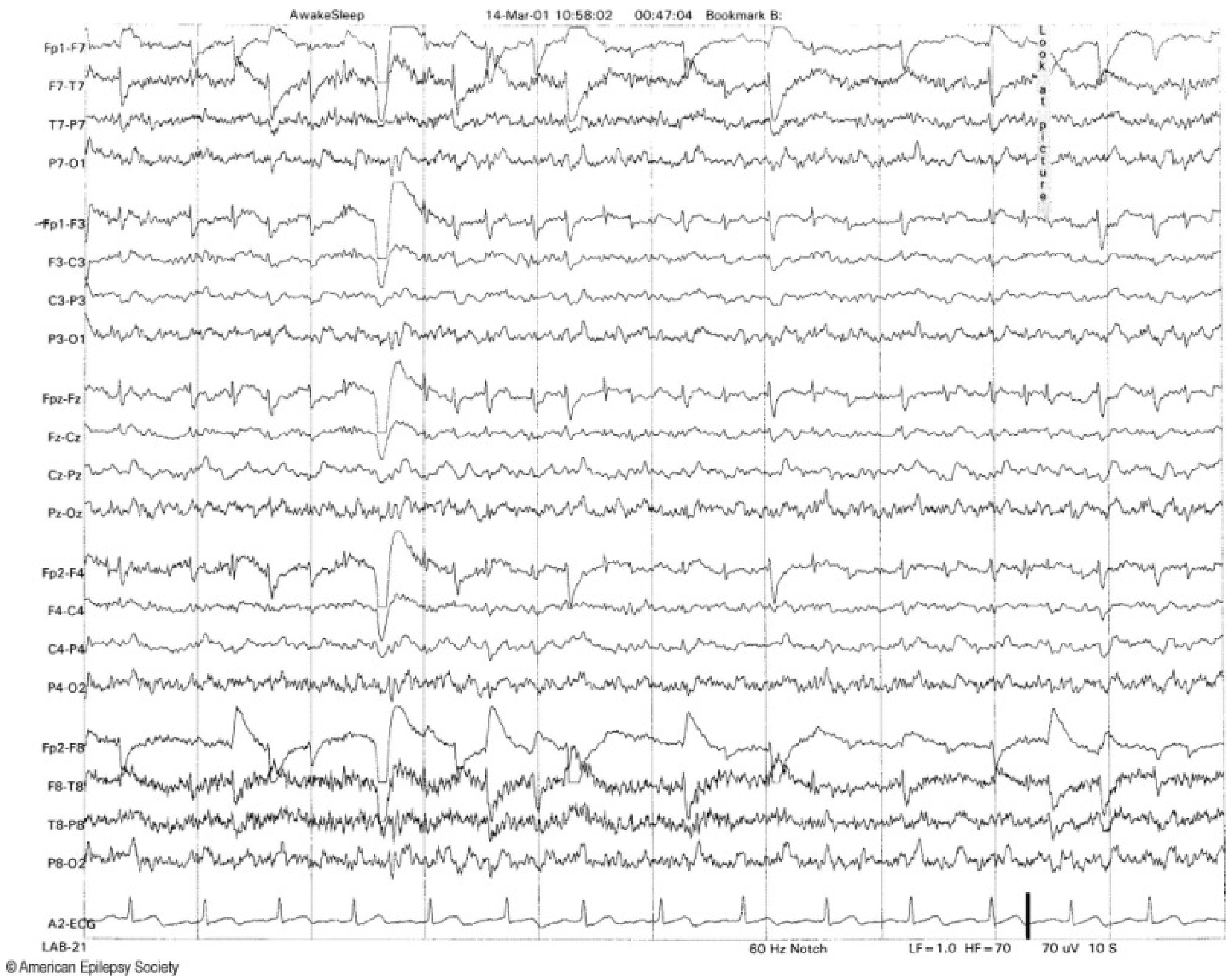


Figure 4. EEG Artifacts due to Rapid Eye Movement. [14]

Non-physiological artifacts are those due to electrical phenomena, devices in the recording environment or movement between the sensors and skin. Special consideration must be made to limit the cause and effect of these artifacts when measuring and processing EEG data.

1.2.7 - Characteristics of EEG

When EEG data is recorded it is a function of voltage over a period of time. Typically in this form, not much can be observed or utilised. Spectral data refers to the amplitude of certain frequencies which comprise a signal, and by converting the EEG data from its time domain to a frequency domain allows the observation of the features of the signal. The fast fourier transform is one method used to do this, by sampling a signal over a period of time and extracting its frequency components. It has been observed and studied that certain frequency ranges of EEG correspond to different mental states [15][16]. Table 1 below displays studied characteristics of certain EEG frequency bands in Adults.

Table 1. Adult EEG Frequency Band Characteristics.

Band Name	Frequency Range (Hz)	Location Present on Brain	State of Occurence in Adults
Delta	1-4	Frontally	Slow-Wave Sleep Satisfaction of basic biological needs Dopamine Production
Theta	4-8	Locations unrelated to task being performed	Drowsiness Repression of Responses Meditation, Creativity Emotional reactions and learning Frontal power increase related to positive emotion
Alpha	8-14	Posterior	Relaxation, Closed Eyes Attenuates with open eyes mental exertion Left Prefrontal power associated with motivations toward pleasure, Right prefrontal increase with negative emotion
Beta	14-32	Frontally, both sides	Open eyes Alert, anxious or focused thinking Imagined movement
Gamma	32-45	Somatosensory Cortex	Binding of neurons to carry out motor or cognitive function Multi-sensory perception Short term memory matching
Mu	8-12	Sensorimotor Cortex	Motor Neurons At Rest Attenuates with Imagined or real movement

It can be noted that in children, the location of the activity and occurrences can differ. For example, in babies and children, the lower frequency ranges have been shown to have higher activity [17]. Thus for the purposes of this study, only responses in adults will be referred to. Also, the majority of usable EEG data for processing falls into the frequency ranges of 1-20 Hz, as the activity in frequency ranges outside of this are usually due to artifacting. Thus, only the frequency bands delta, theta, alpha and beta are typically measured in the 1-20 Hz range.

1.2.8 - Integrated Development Environment (IDE) and Python

An IDE is a software package designed for program developers to design, test and debug software. Spyder 3 is an IDE for the software language Python. Python is one of the most popular and prevalent software languages in modern use, known for its ease of use and readability. Python is highly extensible and has many open source packages and libraries available to use for many types of development and prototyping applications.

1.2.9 - Machine Learning and Support Vector Machines

Machine learning involves the use of statistical techniques by computer systems to improve on the ability to perform a certain task without being explicitly programmed to do so. Sample inputs are fed into the system, which is then analysed and modelled in order to make future predictions. Support Vector Machines (SVM) are a model of Machine Learning allowing for regression analysis and binary classification. Data is fed into the SVM algorithm, labelled with one of two classifications. A complex model is then produced allowing the classification of new data samples. The open source *scikit-learn* python machine learning library contains an SVM algorithm.

1.3 - Purpose of the Project

The main purpose of this project was defined by the following research question: How effective is a low-cost EEG based system for the purpose of enhancing communication in LIS patients in comparison to existing high cost systems? By exploring the capabilities of a low-cost system, the ability for hobby and independent researchers to objectively collect data on brain activity is explored. If research is not limited to those with funding large enough to afford medical-grade, expensive EEG equipment, accessibility for research and development becomes available to a much larger demographic. With more studies and testing performed in this field, greater advancements in the understanding of human brain function can be made. Furthermore, this can lead to consequent enhancements in the treatment of neurological disorders.

Measuring human EEG can be beneficial in the research of many different fields, from business and marketing applications through to clinical and psychological treatment. For example, neuromarketing is the study of the brain's activity during the decision-making process of a consumer. A low-cost, portable EEG measurement system can allow for consumers to be monitored in a natural environment. Human psychology can also be observed by analysing brain activity during interactions between human subjects in different scenarios. In a clinical sense, emotional responses to various stimuli can be studied to help further understand trauma and abuse in patients.

The two aims for this project are listed as below:

1. To develop a low-cost brain-computer interface system, enhancing the ability for those with temporary or permanent loss of motor function to interact with the environment around them.

2. To explore the possibilities for the low-cost, portable and objective analysis of human brain function for research and engineering purposes.

To achieve the aims, the following objectives were determined:

1. Develop a prototype model for allowing a patient with locked-in syndrome to communicate and interact with the environment around them.

2. Develop a system for determining a subject's emotional and mental state with EEG data.

3. Analyze the effectiveness for a low-cost EEG based system for measurement and BCI in comparison to a high-cost system.

2 - Review of the Published Literature

This literature review analyses the works already performed with relation to EEG based communication methods and emotional classification. Furthermore, the relevance of this study with relation to other studies performed is discussed.

Lotte et al. in 2018 [18] performed a review of classification algorithms for EEG-based brain-computer interfaces. They identified two main phases during BCI operation - an offline phase during which calibration is performed, classifying mental commands and an online phase during which the BCI translates commands from the subject to a computer. The system for the online phase of the BCI consists of the following processes:

1. The subject produces a specific EEG pattern which is measured.
2. The EEG signals are filtered and preprocessed, extracting the spectral features.
3. The features are classified, then translated into a command.
4. The subject is given feedback on a recognised command.

Due to the large difference of EEG signals between individuals, successful non-calibrated BCI function has not yet been developed. Thus, the offline calibration phase where a classifier is trained with the spectral features is necessary. This involves the subject performing a desired mental task repeatedly whilst the classifier develops the classification model. These phases are both displayed in Figure 5 below.

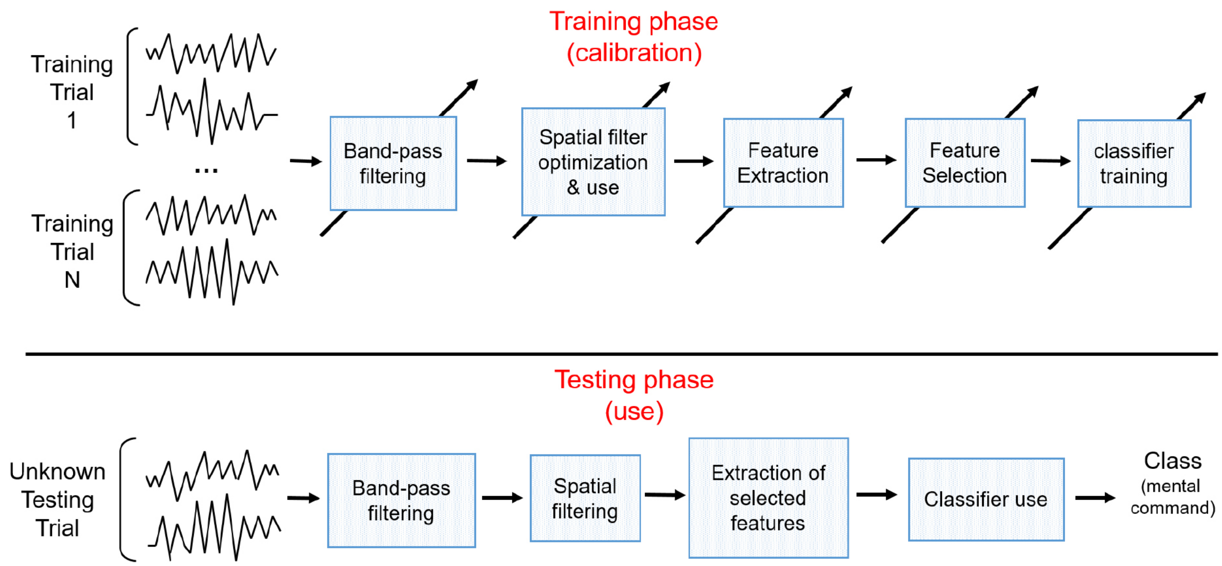


Figure 5. Phases of EEG-based BCI Function. [18]

Kim et al. in 2013 [16] reviewed the methods for emotional estimation from human EEG. To find the emotional correlates, first definitions of emotional state spaces need to be made. The discrete space separates emotional experiences to a set of core positive and negative states, such as happiness, love, fear, sadness and anger. The continuous space classifies emotions based on an arousal (magnitude) and valence (positive/negative) vector space. Emotional stimuli used to elicit states in the subject need to be used based on the choice of emotional state classification space.

The spectral power of specific frequency bands have been found to have correlation with emotional state. Alpha band power has been found to vary with valence, or discrete emotional state shifts. Right-frontal activity showed an increase with negative emotional valence, whilst the asymmetry of band power in the frontal regions has been observed to vary with valence and approach/avoidance aspects of emotion. Theta band power in the frontal region increased with positive emotion. Table 2 below compiles various EEG studies performed and their findings with relation to emotional state correlates.

Table 2. Observed Emotional State Correlation with EEG Powers. [16]

Authors	Year	#subjects	Stimulus (duration)	#EEG channels	Channel location	Emotional state	EEG features	Effects
Davidson et al. [48]	1990	37	Emotional film clips (60 sec)	8	F3, F4, C3, C4, T3, T4, P3, and P4	Happiness, disgust	Alpha power	Left-frontal: happiness < disgust Right-frontal: happiness > disgust Left-anterior temporal: happiness < disgust Right-anterior temporal: happiness > disgust
Aftanas et al. [54]	2001	22	IAPS (7 sec)	128	IS	Valence (+/−)	Theta power (ERD/ERS)	Anterior temporal region (left-hemisphere) Negative: left < right Positive: left > right Parietotemporal region Negative: left < right Positive: left < right
Keil et al. [51]	2001	10	IAPS (1 sec)	128	IS	Arousal	Gamma power	Gamma power (46–65 Hz, 500 ms): arousing ↑
Kemp et al. [104]	2002	16	IAPS (13 Hz)	64	IS	Valence (+/−)	SSVEP amplitude	Negative: SSVEP ↓ at the bilateral anterior frontal area
Pollatos et al. [105]	2005	44	IAPS (6 sec)	61	IS	Arousal	ERP/ECG	Good heartbeat perceivers show higher P300 peak
Balconi and Lucchiari [18]	2006	20	Ekman's picture set (500 ms)	14	Fz, Cz, Pz, Oz, F3, F4, C3, C4, T3, T4, P3, P4, O1, and O2	Neutral versus emotions (happy, sad, angry, and, fearful)	ERD (alpha, beta, delta, and theta)	ERD% of theta (150–250 ms) at the anterior regions Emotional state > neutral state
Baumgartner et al. [106]	2006	24	IAPS + music (16 pictures/4.375 s)	16	F7, F3, FT7, FC3, F4, F8, FC4, FT8, TP7, CP3, P7, P3, CP4, TP8, P4, and P8	Happiness, sadness, and fear	Alpha power	Combining music with pictures evokes more intensive emotional experience
Sammler et al. [55]	2007	18	Music (22–44 sec)	63	IS	Valence (+/−)	Theta power	Frontal midline theta power is increased by positive emotion
Balconi and Mazza [47]	2009	19	Ekman's picture set (30 ms, 200 ms)	32	IS	Anger, fear, surprise, disgust, and happiness	Alpha power (ERD)	Right-frontal activity increase for negative emotions
Li and Lu [107]	2009	10	Picture (6 sec)	62	IS	Happiness, sadness	Gamma ERD	Gamma ERD for emotional stimuli
Petrantonakis and Hadjileontiadis [70]	2010	16	Ekman's picture set	3	Fp1, Fp2, and F3/F4 (bipolar)	Happiness, sadness, surprise, anger, fear, and disgust	Alpha and beta power	Higher-order crossing index improves performance
Lithari et al. [108]	2010	28	IAPS (2 sec)	19	IS	Arousal	ERP (N100, N200)	ERP peaks increase for unpleasant stimuli in female
Petrantonakis [69]	2011	16	IAPS (5 sec)	8	F3, F4, C3, C4, T3, T4, P3, P4	Arousal/valence	Alpha and beta power	Asymmetry index can detect arousal levels

The two methods of classifying emotional state are based on feature extraction and classification methods. Feature extraction utilises neuropsychological knowledge and subject trials in order to correlate feature band powers at certain EEG sensor locations with emotional states. The use of feature band power from parts of the brain irrelevant to the emotional state can cause inconsistencies and error in this method. Machine learning involves the use of statistical signal processing and machine learning methods for the emotional classification. The limitation of this method involves the use of linear classifiers for a non-linear feature structure.

In a study performed to examine self-initiation of EEG based communication in paralysis patients [19], it was found humans can control EEG parameters in the 8-12Hz mu rhythm frequency band and slow cortical potentials (SCP). By learning to control these EEG variables, they can perform "select" and "reject" interface commands. Training was required in the patients in order to make this system functional. The system involved initially starting the input, then having the patients visually observe a computer monitor. They would perform a ‘select’ command when a horizontally moving cursor reached its desired destination, and then another select command when the vertically moving cursor reached its final destination. A ‘select’ or ‘reject’ command could perform or reject the selected prompt.

Another group [20] observed a large amount of documented studies on the correlation between EEG and negative emotion, however a lack with regards to positive emotions. They developed a model classifying the intensity and type of emotional responses of encouragement (awe, gratitude, hope, inspiration, pride), playfulness (amusement, joy) and harmony (love, serenity). This model was tested to be accurate 80% of the time, which determines that it is possible to determine emotional state via EEG. The EEG in this study was measured from 32 electrodes, at a sampling rate of 250Hz. Large weights were placed on the frontal electrodes in order to remove artifacting from electrode movement, and eye movement/muscle activity related artifacting was filtered from the signals using machine-learning filtering techniques. This study indicates that the accuracy of this project’s prototype in determining emotional state may be limited due to the lesser number of sensors in the Muse Headband, and the integration of the complex signal filtering techniques.

Most BCI systems developed involve the use of a high-cost EEG measurement device with a much greater number of sensors than the Muse Headband and better signal-noise ratio. They also involve complex signal filtering and machine-learning classification techniques. This study serves to document the design process and considerations of developing a low-cost system with a short project time-frame, and thus compare the performance and capabilities with other systems.

3 - Design Process

3.1 - Methodology

The methodology to achieve the objectives and outcomes for this project was divided into the 13 steps listed below in Table 3.

Table 3. Methodology to Achieve Project Objectives.

Step	Process
1	Perform literature research on EEG data. Determine what is measurable, the classifications of frequency bands and their relation to human brain function.
2	Perform a literature review on methods of EEG based communication. Investigate previous works, their methodology and effectiveness.
3	Initiate the interface allowing the software to monitor and record data from the EEG device.
4	Perform a test determining the effectiveness of the EEG device in reading blinks, jaw clenches, and different frequency domain brain states.
5	Measure EEG data at different thought processes and emotional states.
6	Identify frequency bands containing important information.
7	Determine filtering and processing necessary to utilise EEG data.
8	Derive communication protocols based on reliable and measurable EEG data characteristics.
9	Develop software to allow visual and auditory outputs of determined communication protocols.
10	Integrate EEG data with software to visually and auditorily output EEG communication protocols.
11	Add functionality to software to display classified emotional and mental states.
12	Perform testing to determine reliability and accuracy.
13	Compare performance with reference to existing systems.

This methodology was developed with reference to the previous works as studied in the literature review. The study by Lotte et al. [18] provided the framework for the classification of mental commands via machine learning. The relevant features used for state classification were based off findings from the study performed by Kim et al. [16] .

3.2 - Design Specifications

The software prototype was designed with the following specifications.

3.2.1 - System Diagrams

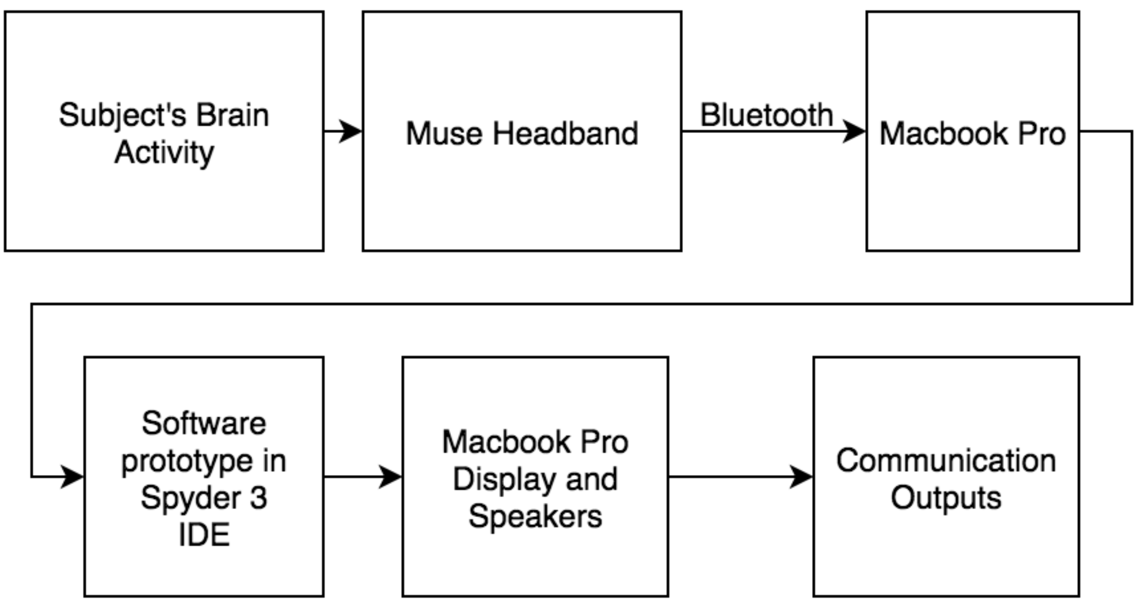


Figure 6. Hardware System Diagram.

Figure 6 above represents the hardware system diagram of the prototype. The subject's brain emits EEG signals, measured by the Muse Headband. These EEG signals are transmitted via bluetooth to the Macbook Pro. The signals are recorded and processed by code produced within the Spyder 3 IDE, and outputs communication through the Macbook Pro display and speakers.

3.2.2 - State Classification Table

Table 4 below lists mental states and their corresponding EEG based classification. As every EEG frequency band has an associated mental state, the magnitude of these frequency bands can be used to determine the subject's state. Machine learning was used for the determination of the YES/NO communication commands.

Table 4. Mental States and Corresponding EEG Classifications.

State	Classification
YES Command	Machine Learning
NO Command	Machine Learning
Asleep	EEG Features determined by testing
Drowsy/Meditative	“
Relaxed	“
Focused/Anxious	“
Positive Valence	“
Negative Valence	“

3.2.3 - Data Processing

In order to utilise the measured raw EEG data, various methods of data processing are required as discussed below.

Fourier Transform

EEG data in its measured format as a function of time is difficult to analyse and process. The Fast Fourier Transform (FFT) is a method to calculate the frequency components that comprise a signal, and the magnitudes of such components. The frequency spectrum for EEG data is much more useful in analysis and processing. The FFT takes a signal comprised of a certain number of samples and outputs a transform with the same number of components, mirrored around 0 Hz.

For non-imaginary data, the negative frequency components contain no unique data and therefore can be discarded. In order to normalise the FFT data, it must be divided by the number of samples in the original data, and multiplied by two to account for the mirrored negative frequency components.

When performing the FFT, a few considerations have to be made in order to ensure the most accurate processing. The FFT is most effective when calculated using a sample window which is a power of two, and least effective with a prime number. Thus, data can be filled with trailing zeros to create a power of two sample window before performing the FFT. Secondly, the FFT is accurate only when the data contains signals with an integer number of periods. As the FFT calculates the transform assuming the data window repeats infinitely, if the signal components do not align then the abrupt changes at the start and end of the window are viewed as spectral leakage in the transform. Windowing is used to improve the effectiveness of a transform by attenuating the data set at the start and end points. The hamming window is a raised cosine wave which is particularly effective for performing the FFT with EEG data. Applying the hamming window to data before performing the FFT limits the spectral leakage, and is necessary in accurate processing.

Frequency Band Powers

The power spectral density of a data set is calculated by taking the absolute values of the normalised FFT spectrum. The absolute band power for a frequency range is the logarithm of the mean of the power spectral densities within that range. The relative band power for a frequency range is the ratio between the linear band power for a range and the sum of the linear band powers over all ranges. Both the absolute and relative band powers are useful characteristics in the analysis of EEG data.

Filtering

Raw signals can contain unwanted noise due to interference from external sources and the measuring equipment. Filters can be used to eliminate or at least attenuate unwanted frequency components in a signal. A bandstop is one kind of filter which attenuates frequencies in a specified range. A bandpass filter allows signals within a certain frequency range to pass through, whilst attenuating others. In the measured EEG data, it was expected that a large contributor to signal noise would be due to the 50 Hz mains power, and to a lesser extent the consequent mains hum frequency of 100 Hz. These components can be eliminated, along with other unnecessary components with a band pass filter, as the desired frequency range to measure and analyse is within 1-20 Hz. If any unwanted components remains after the use of a bandpass filter, a bandstop filter can be used for further attenuation.

The DC bias for a signal refers to the mean amplitude of the waveform. A signal in theory should be centred around 0V, however can be biased due to imperfect measuring equipment. If the EEG data contains a DC bias, it will be reflected in the frequency spectrum as a peak at 0 Hz. This can be eliminated by subtracting the mean of the signal from itself.

Epochs

Epoching is the extraction of specific time-windows from a continuous signal, for the purpose of processing. These epochs can be used to study certain time specific events, or to construct multiple trials with which to use to improve classifier accuracy. When the data is divided into epochs, it creates a three dimensional matrix of shape (time, channel, epochs) where time is the continuous EEG signal for the duration of the epoch length, channel is the EEG electrode and epochs is the number of different epochs formed from the original signal.

Classifier

The Support Vector Machines classifier included in the *scikit-learn* python library takes two arrays as its input. The first contains a set of samples, each with a set of features. The second contains the binary class labels for the set of samples. After being fitted, the classifier then can take new sample data to predict the binary classification.The classifier can also produce a score of the classifier model which represents the prediction accuracy.

3.2.4 - Design Resources

The Muse Headband requires connection with the Bluetooth Low Energy (BLE) Protocol. Due to the limitations of the Windows and OSX operating systems in connecting to the Muse Headband, Linux was chosen as the operating system to use. The Python programming language was chosen to develop the prototype software in due to the availability of pre-developed libraries allowing the connection to the Muse Headband, and also the receiving of live-streamed data.

Table 5 below lists the hardware and software resources required for the project, with their costs.

Table 5. Required Project Resources.

Resource	Cost	Function
Muse EEG Headband	$249 USD	EEG Measuring Device capable of transmitting data via bluetooth.
2014 Macbook Pro 13” Retina	Already Owned	Computer used to connect to the Muse Headband, develop the software prototype and output visual and auditory commands.
Arch Linux	$0	Operating system used on Macbook Pro.
Spyder 3	$0	Interactive Development Environment for the Python Programming Language.

3.3 - Software Design

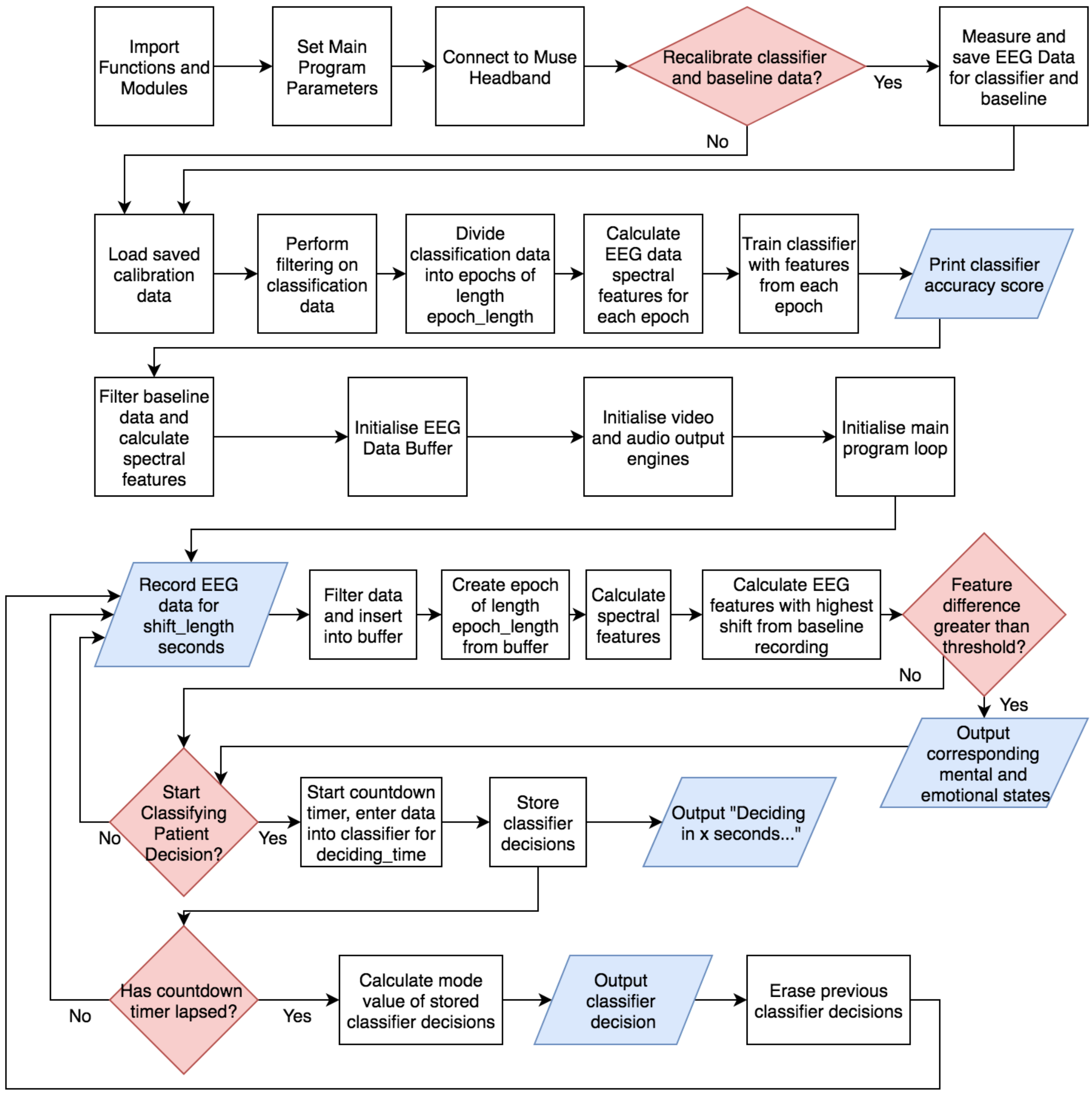


Figure 7. Software Block Diagram.

Figure 7 above displays the block diagram for the prototype software.

Table 6 below details the step-by-step process of the software.

Table 6. Software Processing Steps.

Step	Process
1	Additional python functions and modules are imported, and main program parameters set.
2	The bluetooth connection to the Muse Headband is made, and information regarding the stream is collected.
3	The user is asked whether or not new calibration data is to be recorded - if yes, the new data is saved. If no, the program skips to the next step.
4	The calibration data is loaded, consisting of the classifier training data and the baseline state data.
5	Classification data is filtered then divided into epochs of length epoch_length, and the spectral features calculated for each epoch.
6	The classifier is trained with the features of each epoch, and the classifier score printed.
7	State baseline data is filtered and spectral features calculated.
8	The EEG data buffer is initialized with zeros.
9	The video and audio output engines are initialised.
10	The main program loop starts.
11	The EEG data for length shift_length is recorded, filtered then inserted into the buffer.
12	An epoch of length epoch_length is created from the buffer, and its spectral features calculated.
13	The EEG features with the highest difference from the baseline recording are calculated, and if they surpass a threshold value, the corresponding mental and emotional states are output.
14	If the user starts the decision making process, the epoch is fed into the classifier, with the classifier output stored. If the decision countdown timer hasn't lapsed, the main program loop begins again. If it has lapsed, the mode score of the classifier outputs is used to determine the final Yes/No output.
15	The Yes/No command is visually and auditorily output. The stored classifier outputs are erased, and the loop begins again.

The design considerations for each program step is explained as follows.

Import Functions and Modules

External functions and modules to use in the main program script are imported as listed in Table 7 below.

Table 7. External Python Packages Used.

Python Package	Function
numpy	Module for performing high-level mathematical functions on large arrays and matrices
matplotlib	Library for displaying graphical plots
pylsl	Python interface to the Lab Streaming Layer. The LSL is a network allowing the real-time exchange of time-series data between applications
muse-lsl	A package allowing the connection, streaming, and plotting of EEG data from the 2016 Muse Headband
pyttsx3	Text to speech
time	A module allowing functions related to time
pygame	Python modules for outputting graphical displays
threading	Module allowing python code to run in multiple threads
sklearn.svm	Support Vector Modules machine learning python library
scipy	Library offering various maths, science and engineering functions

Set Program Parameters

The program parameters are set, as listed and described below in Table 8.

Table 8. Program Parameters.

Program Parameter	Description	Unit
display_width	Width of graphical output display	pixels
display_height	Height of graphical output display	pixels
buffer_length	Length of the EEG data buffer	seconds
epoch_length	Length of epoch sample for classifier input	seconds
overlap_length	Length of overlap between epochs	seconds
training_time	Length the calibration data is recorded for	seconds
decision_time	Time over which the classifier decisions are collected before making a final decision	seconds
emotion_threshold	Threshold at which to display emotional state	%
mental_threshold	Threshold at which to display mental state	%

Connect to Muse Headband

Pylsl is the python library allowing the input of EEG data from the Muse Headband. From the pylsl instructions, initially all data streams on the network need to be found. The *resolve_byprop* function does this, and the Muse EEG stream is found by inputting the stream type to be 'EEG' with a timeout of two seconds if the stream isn't found. A StreamInfo object is returned, from which can be used to create an inlet data stream. This stream has time correction applied to it in order to sync with the computer's clock.

Record New Calibration Data

The user is asked whether or not they want to train new calibration data, consisting of the two data sets to train the classifier, and the baseline state data from which to determine the subject's mental and emotional states. If yes, then the program prompts the user by informing which data is being recorded, and waiting for an input before starting the EEG data measurement. The data is measured for the time specified by *training_time*. A countdown timer is printed also, to inform the user of the duration of data collection. This is repeated until all three data sets are saved. The

EEG data is recorded as a two dimensional matrix of shape (time,channels), where time consists of the EEG data samples and channels is the number of EEG electrodes.

Classifier Training

To train the classifier with the data saved previously, the two sets of classifier data are first loaded. Bandpass filtering between the relevant frequencies of 1-20 Hz is performed, along with a bandstop filter between 49-51 Hz to attenuate the 50 Hz mains frequency. As the classifier is trained with data of length epoch_length, the classification data is divided into epochs. The spectral features for each epoch is calculated by applying the hamming window to the data, performing the FFT then calculating the absolute band powers. This matrix of spectral features is then concatenated, to fit the classifier input requirements, and also normalised by subtracting the mean, then dividing by the standard deviation. The mean and standard deviation for normalisation are saved for later use. A label vector is created to label each set of features with the binary 0 or 1 classification. This, along with the feature data is fed into the classifier, which returns a the classifier object and an accuracy score to be printed for the user to observe.

Further Initialisations

To complete the initialisations before the main software loop, first the baseline state data is filtered and then processed to calculate its spectral features. Then, an EEG data buffer is created with zeros of size *buffer_length* seconds multiplied by the sampling frequency, with four channels. Finally, the engines used to output display and audio from the main program code are initiated from the modules *pygame* and *pyttsx3* respectively.

Main Program Loop

EEG data is acquired for the duration of *shift_length*, which is obtained by the difference of *epoch_length* and *overlap_length*. This data is filtered, then inserted into the data buffer. From the buffer, an epoch of length *epoch_length* is created. This epoch has its spectral features calculated, and the EEG features with the highest percentage difference compared to the baseline data features are used to determine the mental and emotional states of the subject. These states are visually output if the percentage difference exceeds the threshold values of *emotion_threshold* and *mental_threshold*. Another program thread waits for a user input, in which the user input sets the deciding state to be true. In the main program thread, if the deciding state is true, a countdown timer is started and visually displayed, whilst the eeg features for each loop iteration are entered into the classifier. The classifier outputs are stored in an array, until the countdown timer elapses. At this point, the mode of the classifier outputs is used to determine the final decision to be visually and auditorily output. The classifier outputs are erased, and the deciding state set to be false until the next user input.

The full program code can be viewed in the Appendix, Section 7.2.

4 - Prototype Testing

4.1 - Testing Methodology and Criteria

4.1.1 - Test One: Data Plotting, Fast Fourier Transform and Filtering

The software's ability to plot signals, perform the fast fourier transform and apply filtering to the EEG data were tested first with basic signals in order to ensure that they function as expected. To do this, a signal created by the summation of five sine waves, each with specified frequencies of 14 Hz, 20 Hz, 50 Hz, 90 Hz and 150 Hz was created and plotted. The fast fourier transform was then performed on this signal and plotted. This transform should display peak magnitudes of one at each of the frequencies of the individual sine waves comprising the created signal. Next, the band pass filter was applied between 10-70 Hz and the bandstop filter applied between 49-51 Hz. This was then plotted along with the transform. The filtered time series data should appear to be visibly cleaner, with the filtered transform showing a reduction of the 50 Hz frequency to verify the bandstop filter operation, and a reduction of the 90 Hz and 150 Hz frequencies to verify the band pass filter operation.

4.1.2 - Test Two: EEG Data Collection

Next, testing was performed to ensure the software could successfully receive EEG data from the Muse Headband. This was performed by recording two series of data sets, with the subject performing different tasks as detailed in Table 9 below. The data was collected for a total of twenty seconds for each test.

Table 9. EEG Data Acquisition Tests with Associated Tasks Performed.

Test	Task Performed	Criteria
Blinks	Both eyes blinked ten times	Ten distinct artifacts in the time-series data plot
Jaw Clenches	Jaw clenched ten times, without blinking	Ten distinct artifacts in the time-series data plot

The data was plotted and expected to display four channels of EEG data, with ten artifacts at most or all of the electrode sites during each test.

4.1.3 - Test Three: EEG Filtering and Feature Calculation

The software's ability to filter out sources of interference and calculate the EEG features was tested next. Ten seconds of transmitted EEG data was recorded whilst the subject wore the headband. This data was plotted on all four electrode channels to observe the time-series and frequency transform. A hamming window was applied to the original signal to perform the FFT. A lowpass filter was applied at 45 Hz, and bandstop filter applied between 49-51 Hz to further attenuate the 50 Hz component. It was expected to have peaks in the transform from sources of noise such as the 50 Hz mains and the 0 Hz DC Component. After filtering the data, the time-series was expected to have visibly less noise with the hamming window applied and for the spectral transform to display reduced peaks from the undesired noise components.

4.1.4 - Test Four: Classifier Mental Tasks

In order to determine the mental tasks to be performed by the subject during classifying, two sets of contrasting mental tasks were performed and analysed to determine their effectiveness in classifier differentiation. Table 10 below displays the tasks used for testing. The tasks were chosen with reference to Table 1, being easy to reproduce and expected to have the greatest difference in EEG spectral features.

Table 10. Possible Mental Tasks for Classification.

Task Set	0 (No) Classification Mental Task	Expected EEG Feature	1 (Yes) Classification Mental Task	Expected EEG Feature
1	No Imagined Movement	High Alpha due to relaxed state with no imagined movement.	Imagined Movement of Limbs	High Beta due to imagined movement
2	Create mental image of imaginary animal with features from different species	High Theta due to high creativity required	Perform mental arithmetic by calculating multiplication table of arbitrary number	High Beta due to high level of focus and concentration required

The absolute and relative band powers were plotted, with their values from each task tabulated. The mental tasks to be used was determined by first determining which task set had the greatest difference in band powers between tasks, and second from the modelling score the classifier function returns.

4.1.5 - Test Five: Classifier Accuracy

Table 11. Test Five Commands.

Order	Attempted Command to Produce
1	Yes
2	No
3	Yes
4	No
5	Yes
6	Yes
7	No
8	No
9	Yes
10	No

The classifier was tested to determine its ability to produce the desired output for the subject. The subject was given a list of ten ‘Yes’ or ‘No’ commands as above in Table 11, with the purpose of producing those commands in order. The test was performed a total of three times, with the final score calculated by taking the mean of the percentage of correct produced outputs.

4.1.6 - Test Six: Visual and Auditory Communication Outputs

The software’s ability to produce visual and auditory outputs was tested. The desired output from the program consisted of the following:

1. Classifier YES/NO Command. When the software receives input to start the decision making process, a countdown will appear visually on the screen with the following text: “Deciding in X seconds…”. When the countdown reaches 0 seconds, the words ‘Yes’ or ‘No’ will appear on screen visually and also be spoken by the text to speech module.

2. Emotional State Classification. The software must be able to visually display the mental states as listed in Table 4 - Asleep, Drowsy, Relaxed and Focused, along with the emotional valences of negative and positive.

4.1.7 - Test Seven: EEG Feature Analysis for State Classification

To determine if it was viable to determine the subject’s state with the use of EEG features, the subject performed various mental tasks and observed various emotional stimulus whilst their EEG data was recorded. Next, the percentage difference in EEG features between a baseline recording, where the subject was to be in a neutral state, and the elicited states were calculated. The differences Table 12 below lists the five data recordings and the respective tasks performed by the subject. The data was collected for a total of thirty seconds for each test. The tasks performed were formulated with reference to Table 1. Note: the delta and theta bands were not tested due to the need for the subject to be in a sleep or drowsy state, which was not effectively reproducible in the test environment.

Table 12. Tasks to Determine Effectiveness of Prototype in State Classification.

Subject State	Task to be Performed	Expected EEG Features
Baseline	Eyes open whilst keeping a mental time counter	Baseline determines reference for other state EEG data
Relaxed	Eyes closed with deep slow breathing to calm the mind. Meditative music played in the background.	Increase in Theta and Alpha, Decrease in Beta
Focused	Eyes open while a series of provided mental arithmetic was performed.	Increase in Beta, Decrease in Alpha
Positive Valence	Observe video eliciting positive valence	Increase in left prefrontal Alpha
Negative Valence	Observe video eliciting negative valence	Increase in right prefrontal Alpha

4.2 - Test Results

4.2.1 - Test One: Data Plotting, Fast Fourier Transform and Filtering Results

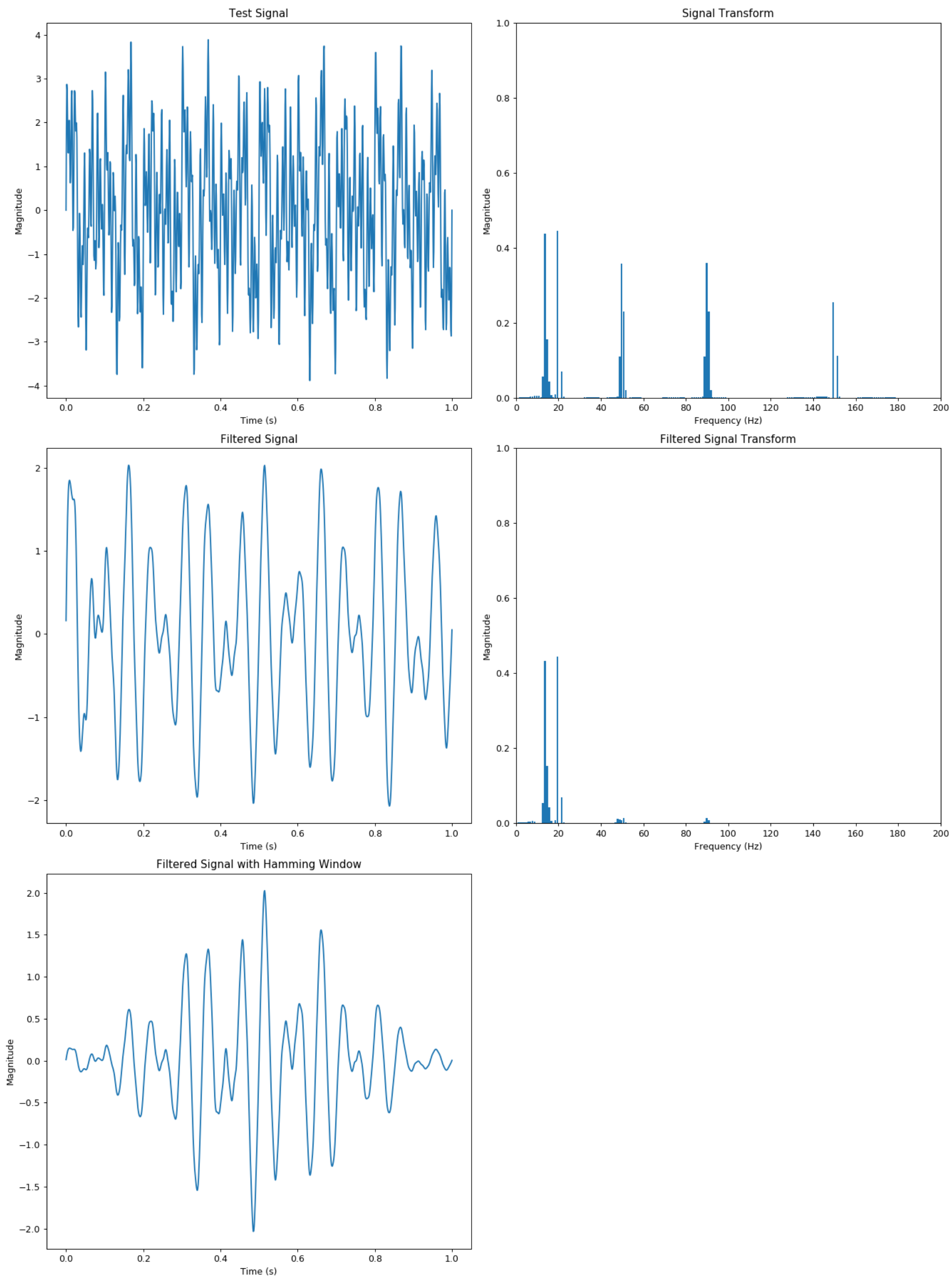


Figure 8. Test One Signal Outputs.

Figure 8 above displays the plotted signals from Test One. The Test Signal plot displays the created signal, a summation of five different sine waves. The Signal Transform plot displays the fast fourier transform of the test signal, resulting in the peaks at 14 Hz, 20 Hz, 50 Hz, 90 Hz and 150 Hz as expected. The filtered signal plot displays the signal after the bandstop and bandpass filters were applied. The signal was visibly less distorted, which indicates successful filtering. The filtered signal transform verifies this, as the 50 Hz, 90 Hz and 150 Hz signal components were nearly eliminated. The signal with hamming plot verifies the hamming window functionality, as the signal was attenuated at the start and end of the sample window. As the test results were as expected, the program was concluded to have a functional plotting, fast fourier transform, filtering and hamming window capabilities.

4.2.2 - Test Two: EEG Data Collection Results

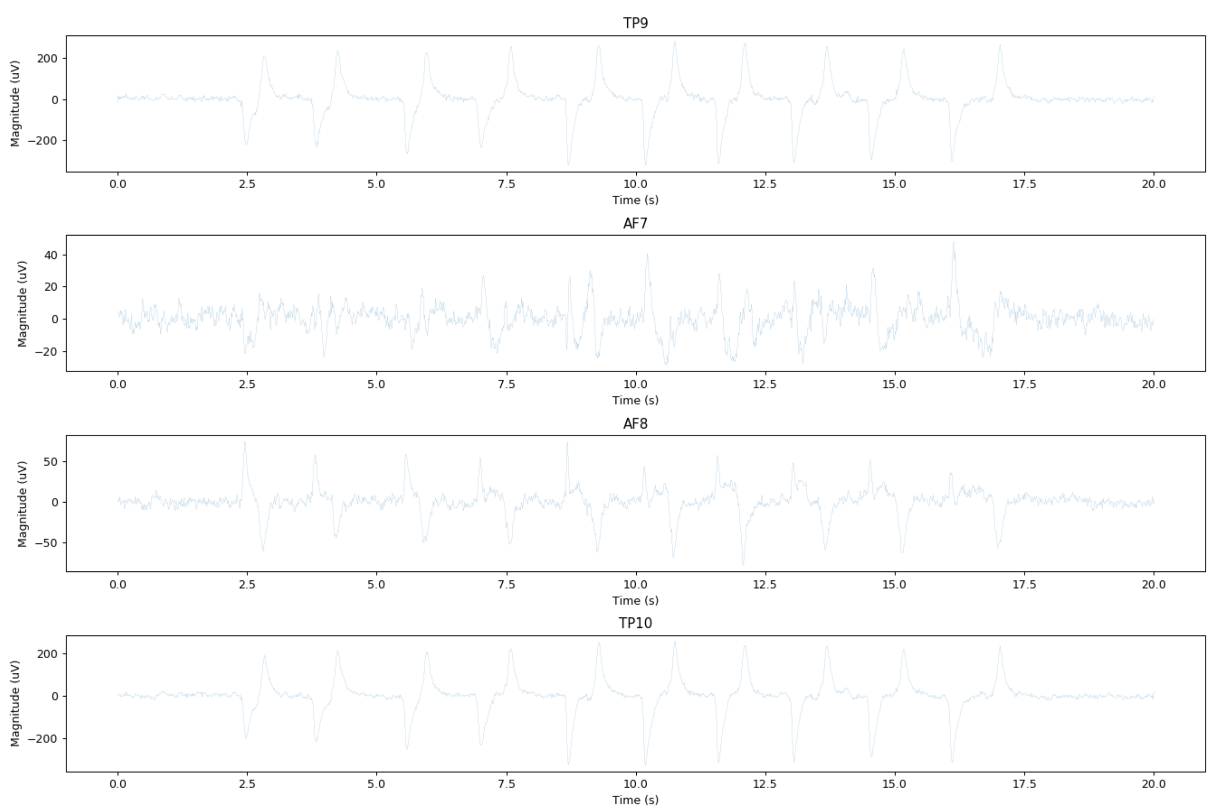


Figure 9. Blinks Test EEG Data.

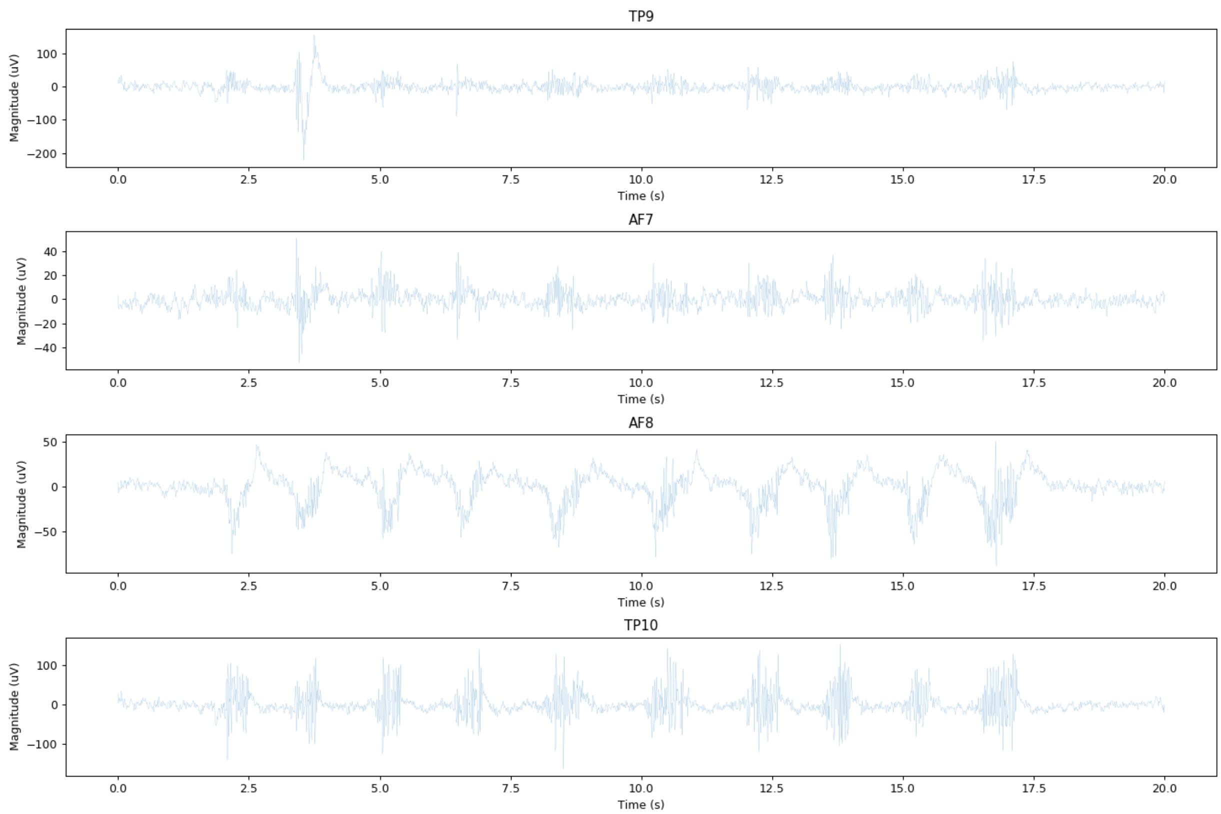


Figure 10. Jaw Clench Test EEG Data.

Figure 9 and Figure 10 above both display the EEG signals for the blinking and jaw clenching data acquisition tests. The criteria for successful measurement was for there to be EEG activity on all four electrodes, and ten distinct artifacts during the 20 second timeframe. As the criteria was satisfied, the test was successful and the headband is verified to accurately transmit data.

4.2.3 - Test Three: EEG Filtering and Feature Calculation Results

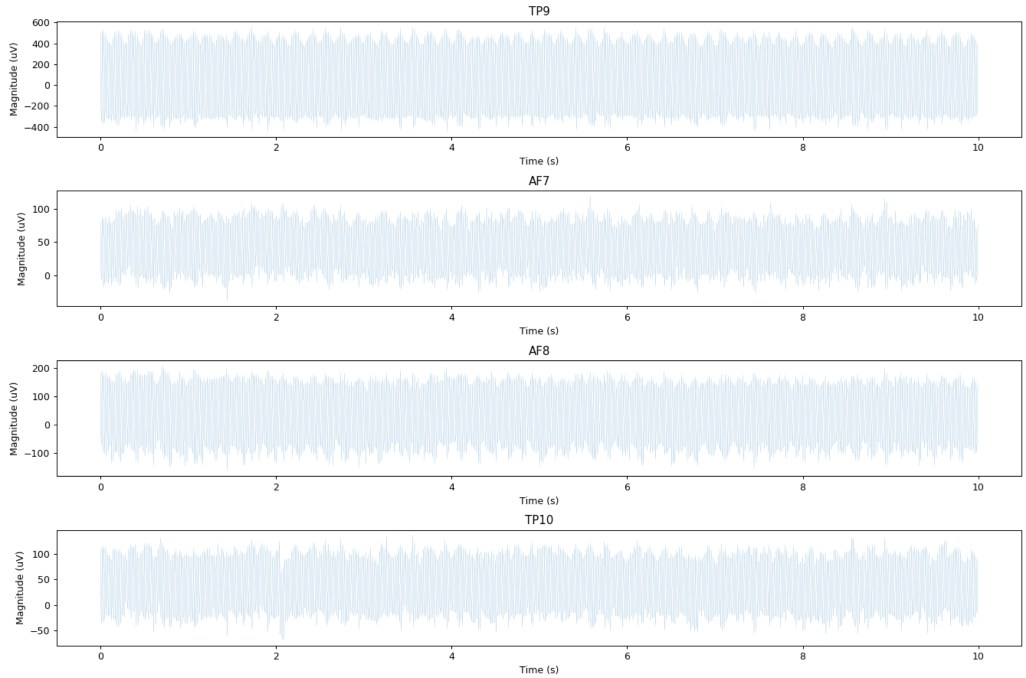


Figure 11. Unfiltered EEG Signals.

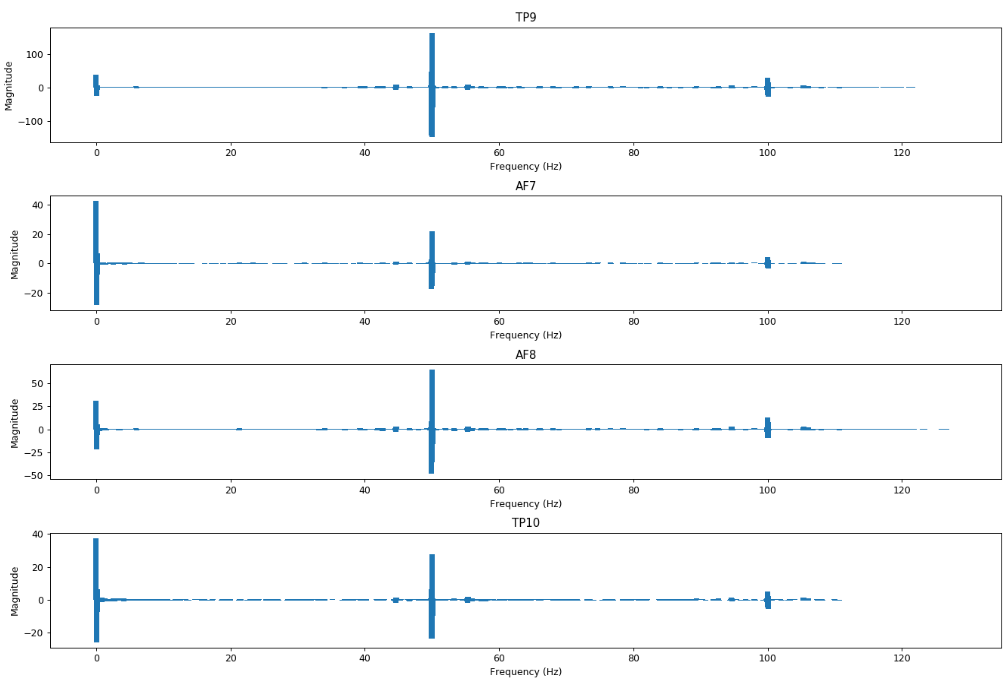


Figure 12. Unfiltered EEG Spectrogram.

Figure 11 above displays the raw EEG data collected. When observing the spectrogram of the EEG data in Figure 12, it is evident the 50 Hz mains power contributed a substantial amount of interference to the signal. There was also noise at 0 Hz, which can be attributed to the DC mean component, and at 100 Hz, which is the hum frequency of the 50 Hz mains. The 50 Hz and 100 Hz were removed by the bandpass and bandstop filter, and the DC mean component removed with the *signal.detrend* function.

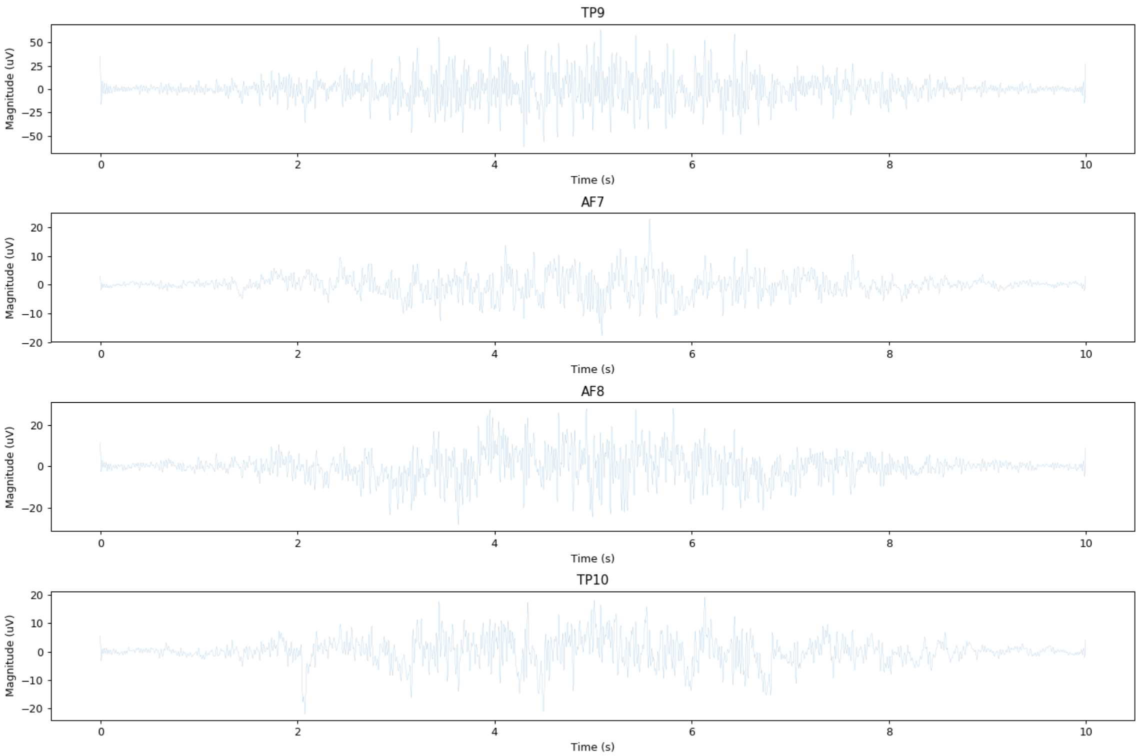


Figure 13. Filtered EEG With Hamming.

Figure 13 displays the filtered EEG signals, which have noticeably less noise. The effect of the hamming window is also visible, with the signal attenuated at the start and end.

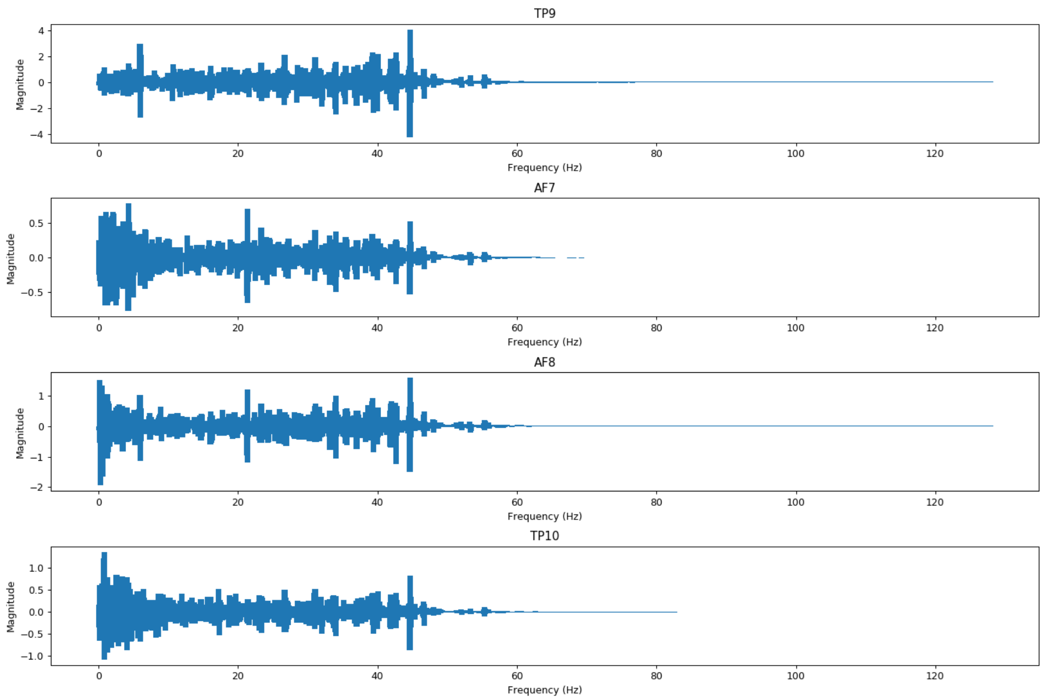


Figure 14. Filtered EEG Spectrogram.

The spectrogram of the filtered signals in Figure 14 confirm that the peaks at 0 Hz, 50 Hz and 100 Hz have been filtered out, with much more relevant spectral data visible. The criteria for successful test were all met, verifying successful EEG filtering and feature calculation capabilities.

4.2.4 - Test Four: Classifier Mental Tasks Results

Figure 15 below displays the plotted band powers for the Task 1, 'no' classification test. The absolute band powers plot displays the powers of the delta, theta, alpha and beta frequency bands for each of the four electrode channels. These powers are displayed on a logarithmic scale. The relative band powers plot displays the band powers as a percentage of each frequency band with relation to the sum of all the other powers. The delta and theta bands had quite high band powers, with the alpha band comprising of approximately 10% of the total band power, with the beta band power so low relative to the other bands that it wasn't visible on the relative band powers plot.

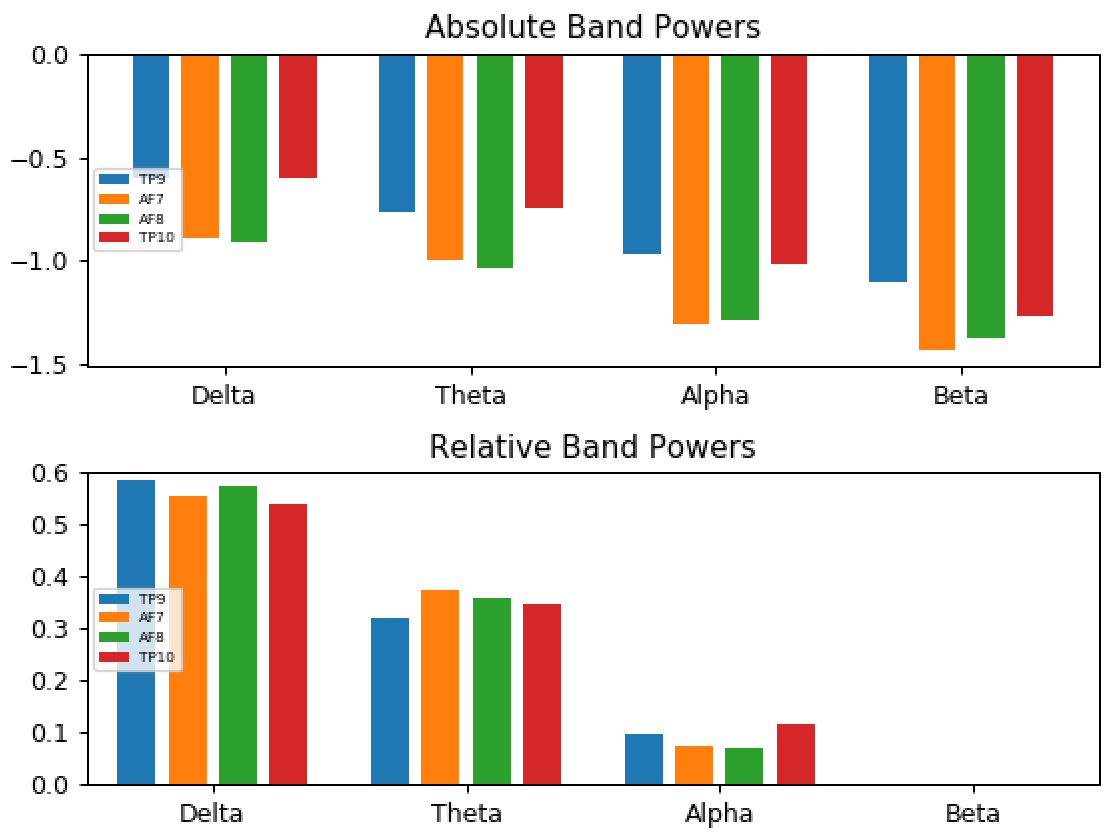


Figure 15. Task Set 1, 'No' Classification Band Powers.

Table 13. Test Four - Mean Absolute Band Power Results.

Task Set	Task	Delta	Theta	Alpha	Beta
1	0	-0.750	-0.886	-1.144	-1.295
1	1	-0.734	-0.916	-1.154	-1.286
Band Power Difference		0.016	-0.030	-0.010	0.009
2	0	-0.735	-0.935	-1.146	-1.226
2	1	-0.707	-0.878	-1.164	-1.272
Band Power Difference		0.028	0.057	-0.018	-0.046

Table 13 above displays the absolue band power values, taken as the mean across all four electrodes. For the first task set, task 1 was expected to have a higher alpha power, and lower beta power than task 0 due to the lack of imagined movement in the subject. The results show that there was an increase in the delta and beta powers, whilst the theta and alpha powers decreased. For the second task set, task 1 was expected to have a higher beta power with a decrease in theta power compared to task 0. There was an increase in delta and theta powers, whilst a decrease in alpha and beta powers. This was the opposite of the expected result. The importance of this task was to determine which two mental tasks resulted in the greatest difference in band powers, in order to achieve a greater accuracy in classification. Although the task set 2 resulted in the opposite of the expected results, the difference in band powers was much greater than between the tasks in the task set 1. Thus, the mental tasks to control the classifier were chosen to be task set 2.

Table 14. Test Four - Mean Relative Band Power Results.

Task Set	Task	Delta	Theta	Alpha	Beta
1	0	0.562	0.349	0.089	0
1	1	0.600	0.317	0.083	0
2	0	0.646	0.292	0.062	0
2	1	0.603	0.333	0.064	0

Table 14 above displays the relative band power results for test four, taken as the mean across the four electrodes. As can be seen, the beta values were so low in relation to the other band powers. Thus, the relative band powers were not of use to this project and not included in further processing and calculations.

When inputting both of the task sets into the classifier, the classifier returned a score of 90% for task set 1, and 91% for task set 2. Although a minute difference, this score reinforces the decision to use task set 2 for the mental tasks. The classifier score relates to the percentage of data points that fit to the classifier model, and accuracy of further predictions.

4.2.5 - Test Five: Classifier Accuracy Results

Table 15. Classifier Accuracy Test Results.

Order	**Attempted Command to Produce**	**Classifier Output**
1	Yes	Yes
2	No	No
3	Yes	Yes
4	No	No
5	Yes	Yes
6	Yes	Yes
7	No	No
8	No	No
9	Yes	Yes
10	No	No
Classifier Training Time	15 Seconds	
Decision Time	7 Seconds	
Classifier Score	98%	
Tested Accuracy	10/10=100%	

Table 15 above displays the results for the classifier accuracy test. The classifier was trained with 15 seconds of training data, and returned a score of 98%. The time period for which the classifier predicts decisions was 7 seconds, and the final decision was calculated by taking the mode of the decision predictions in the time period. The tested accuracy of the classifier was found to be 100%, verifying successful capability to classify mental tasks and produce communication outputs. It is important to note however that this result was taken after several practice attempts at producing the desired classifier output. Testing was also performed in a quiet room with no disturbances, with the subject easily able to focus on the necessary mental tasks.

4.2.6 - Test Six: Visual and Auditory Communication Outputs

Figure 16 below displays a screenshot of the visual output the software produces. As can be seen, the main large text at the top displays when the software is deciding the communication output. The icons at the bottom display the subject's mental and emotional state at any given moment in time. When the countdown reaches zero, the display outputs a yes or no command, and also produces an auditory output from the text to speech module. This test produces the expected results as per the criteria, thus the software can effectively produce auditory and visual communication outputs.

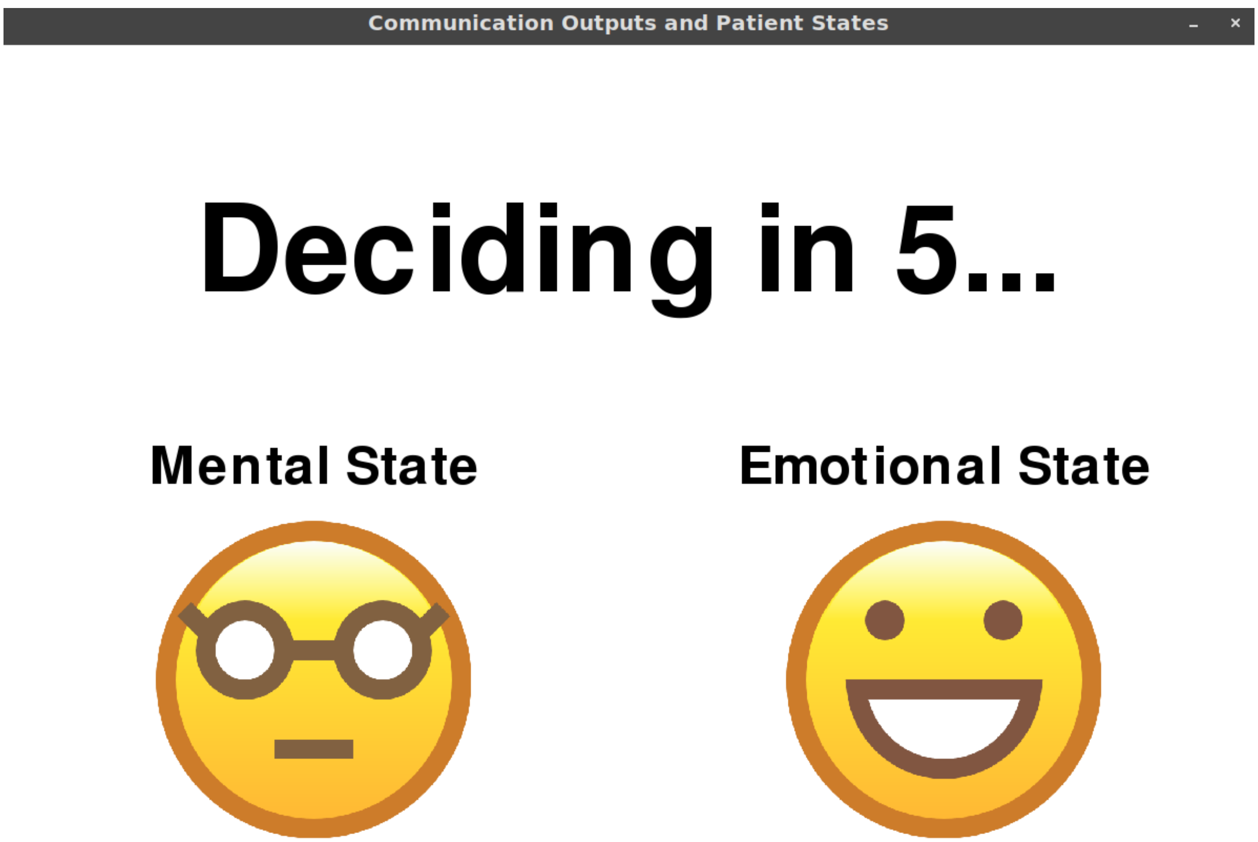


Figure 16. Software Visual Output.

4.2.7 - Test Seven: Effectiveness of State Classification

Table 16. EEG Feature Differences Between Baseline and Elicited States.

State	Delta (%)	Theta (%)	Alpha (%)	Beta (%)
Relaxed	4.18	-5.58	-13.92	-3.79
Focused	-57.67	-44.20	-20.14	-4.47
	Prefrontal Left (AF7) Alpha (%)		**Prefrontal Right (AF8) Alpha (%)**	
Positive Valence	20.78		-1.70	
Negative Valence	-15.42		-28.74	

Table 16 above displays the results for test seven. The relaxed state test was expected to have an increase in alpha and theta powers, with a decrease in beta powers. The test resulted in an increase only in delta, with a decrease in theta, alpha, and beta powers. The focused state test resulted in a large decrease in delta and theta powers, a moderate decrease in alpha power and a minimal decrease in beta power. The decrease in theta and alpha was to be expected due to the increase in focus required for this task. The inconsistency of the results when compared to the expected results from literature were isolated to two potential causes. First, could be the inaccurate measurement of baseline data. The subject could have not been in a neutral state from which the other states could have been effectively determined. The recorded data may have also not been measured for long enough to establish an effective baseline. Second, inaccurate elicitation in the subject of the states that were measured would cause inaccuracies in the expected EEG features.

The positive valence test resulted in a 20.78% increase in the left prefrontal alpha power. This was to be expected when referencing literature results. There was a minimal decrease in prefrontal right alpha power, also to be expected. The negative valence test showed a 15% decrease in prefrontal left alpha power, which was to be expected however resulted in a large 28.74% decrease in the prefrontal right alpha power, where an increase was to be expected. Further testing must be performed to determine whether this is due to artifacting or error, or in the classification methodology. The considerable increase in the prefrontal left alpha power during the positive valence test, and decrease during the negative valence test suggests that the prefrontal left alpha power was a reliable indicator of emotional valence, and thus implemented into the software prototype.

5 - Discussions

5.1 - Low Cost vs High Cost System Analysis

In order to evaluate the performance and capabilities of the low-cost system developed in this project, it was compared with two higher cost systems with longer project time-frames. The system developed for this project is referred to as System One. System Two was developed by Martin Spuler in 2017 for the purpose of evaluating BCI performance using dry electrodes [21]. System Three was developed by Luu et al. in 2017 to develop the effect of a closed loop BCI on cortical involvement during human gait [22]. The system components to be compared and analysed were the EEG measurement equipment, the signal filtering methods, the signal processing and finally the performance capabilities and accuracy.

5.1.1 - Low Cost vs High Cost System Equipment Analysis

Table 17. Comparison of Equipment Used in Different BCI Systems.

Systems	System One	System Two	System Three
EEG Equipment	Muse Headband	g.tec g.USBamp 16x g.Sahara electrodes	Brain Products Acticap 64 channel V-Amp amplifier
Noise Floor	2 uV	<0.4 uV	<2 uV
Electrodes	Dry 2 x Silver 2 x Silicone-Rubber	Dry 16 x Gold Alloy	Wet 64 x Silver Chloride
Sampling Rate	256 Hz	600 Hz	128 Hz
Cost (Estimated)	$249 USD	$25,000 USD	$30,000 USD

Table 17 above compares the equipment used in the three BCI systems. The Muse Headband used in this project has the highest noise floor, and contains only four electrodes which read the user's EEG via dry conduction (no conductive gel between electrodes and the skin). The sampling rate is at 256 Hz, which is in the mid-range of the other systems and costs less than 1% of the other two systems at only $249 USD. System Two uses the the g.USBamp and g.Sahara electrodes manufactured by g.tec. They have the lowest noise floor out of all the systems at <0.4 uV, and allow dry measurement with the 16 Gold Alloy electrodes. The sampling rate of 600 Hz is the highest, and the price is estimated to be around the $25,000 USD price point. System Three uses the Brain Products 64 channel acticap with the v-amp amplifier. This utilises a wet conduction method, where the electrodes have gel applied to the surface to increase the conduction against the surface of the scalp.

As to be expected, the high-cost EEG measurement equipment have higher accuracy in the measurements, with a vastly greater number of electrodes capable of obtaining EEG from many different areas of the brain. However, in situations where a large number of EEG channels are not necessary, the Muse Headband is a very viable option. This is due to the comparable noise floor and sampling rate, and a cost of less than 1% of the other measurement systems.

5.1.2 - Low Cost vs High Cost System Filtering Analysis

The filtering of the EEG differs greatly between the three systems. In System One, a bandstop was used to attenuate the 50 Hz mains power component, and a bandpass to filter out frequencies not being measured. The mean DC component was also removed to filter out the 0Hz component.

System Two applied a notch filter at 50 Hz and two bandpass filters, one over the range of 0.5-60 Hz and another over the smaller range of 1-30 Hz. Spatial filtering was then performed, from canonical correctional analysis to reduce multi-channel EEG data into a single channel and improve the signal-noise ratio. This has shown to increase the efficiency of visual evoked potential (VEP) classification [23].

System Three incorporates the most advanced filtering methods. First, a 0.1 Hz high pass was performed on the data, and EEG channels with a standard deviance greater than 1 mV were removed from processing. Unlike the other two systems, the electrooculography (EOG) was also measured using six of the electrodes and used to feed artifact filtering algorithms. EOG is the measurement of the potential between the front and back of the eye, and as EOG and EEG signals contaminate each other, are a common cause of artifacting in EEG. Generally, filtering out EOG results in the loss of important EEG data, however new EOG filtering algorithms have been shown to preserve EEG data [24]. Along with the EOG filtering, artifact subspace reconstruction was used to remove artifacts from eye blinks and muscle bursts, and replace the EEG with predicted data. Finally, any EEG channels with remaining artifacts were visually observed and removed.

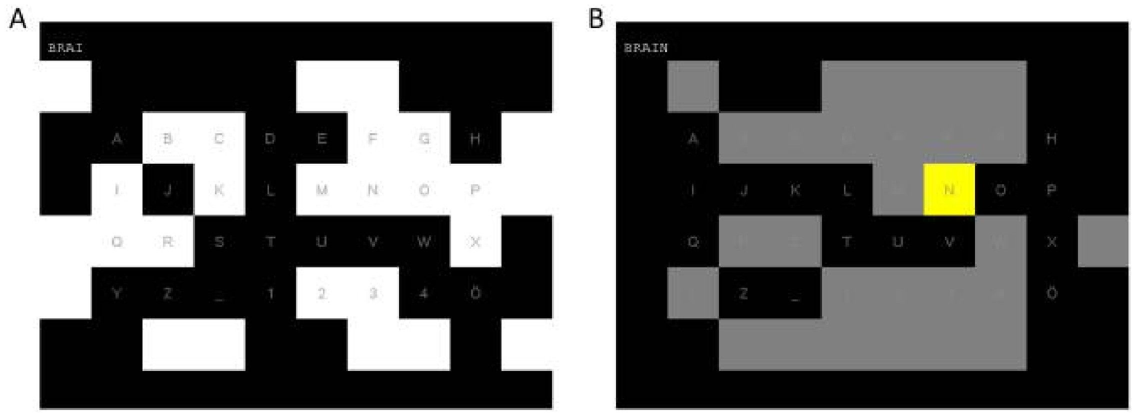
System One performs the very basic filtering necessary on the raw EEG data before processing. System Two utilises similar filtering, with the addition of spatial filtering to further decrease noise and increase classification accuracy. System Three however uses the most complex filtering processes out of the three systems. These are representative of the necessary filtering to produce highly accurate EEG for complex BCIs reliant on highly accurate data measurement.

5.1.3 - Low Cost vs High Cost System Signal Processing Analysis

The system in this project uses a simple SVM binary classifier to predict communication outputs from EEG features. The state classification is performed simply by determining the relevant features with the highest percentage shift from an initial, baseline recording.

System Two uses code-modulated visual evoked potentials (c-VEP) - which are changes in EEG due to a visual stimulus, in order to select or classify data markers which shift over time. The visual stimulus is modulated for each data marker, resulting in different EEG dependant on the marker that the subject focuses on. Classification of each data marker is obtained by averaging multiple test trials. Detection of the target focused on by the subject was determined by the highest calculated pearson correlation coefficient between the template and measured EEG data. Calibration was performed by having the subject initially write 64 characters, by selecting the corresponding data markers, with visual feedback on successful selection. Figure 17 below displays screenshots of the BCI. Left is a screenshot during a trial, Right displays the selection of the letter ‘N’.

Figure 17. Screenshots of the System Two BCI.



System Three utilises an unscented kalman filter (UKF) in order to decode the neural data and determine the joint angles of the subject’s lower limbs during operation. This data was used to control a walking avatar on display, for the subject to visually observe to create a closed feedback loop of their walking cycle. During the BCI operation, the parameters were constantly updated using a closed-loop decoder adaptation (CLDA) in order to improve performance. In order to evaluate the effect of the closed feedback loop on the walking cycle, the difference in spectral band power between the two walking conditions was calculated by segmenting the data into time periods for each gait cycle. A 3D electrode localisation system, the BrainVision Captrak by Brain Products, was used in order to record the position of the electrodes and ensure the electrode cap was fitted consistently to the subjects each time.

System One uses basic processing in order to obtain functional information from the measured EEG, however both Systems Two and Three are demonstrate the advanced processing used in highly sophisticated BCI applications.

5.1.4 - Low Cost vs High Cost System Capabilities Analysis

To determine the capabilities of System One, the results from Test Five were considered. The recorded trial resulted in an accuracy of 100%, taking seven seconds per classification. Thus, as each classification was an output of a 0 or 1, the output bitrate of this system was determined by the number of classifications per minute. This was calculated to be 8.5 bit/min. The system was found to have a high accuracy in the initial testing, yet a low output bitrate. As only one trial was performed with only one subject, the calculated system accuracy is highly likely to be inaccurate.

System Two utilised testing with twelve subjects, each performing four experimental blocks of three runs where 64 characters were attempted to be selected. The final accuracy of the system was determined to be 75.9%, with an accuracy of 46.2 bit/min. Due to the amount of subjects and trials, the determined system accuracy is likely to be quite accurate.

System Three's performance was dependent on the accuracy of the neural decoder in determining lower limb joint angles, rather than its ability to produce communication outputs. Therefore, it could not be quantifiably compared to Systems One and Two. The pearson R value was used in measuring the accuracy of the joint angles, with the median values for the hip, knee, and ankle being 0.85, 0.79 and 0.63 respectively. The system was also able to determine the location for the cortical locations responsible for the motor processes using independent component analysis and k-mean clustering.

To compare the capabilities of the low cost and high cost systems, it is best to compare System One and Two as they both aim to improve communication ability using a BCI. Whilst System One can only produce an output of 'Yes' or 'No', and very crudely predict the subject's emotional and mental states, System Two allows the subject to self initiate communication and produce sentences, with the ability to erase accidental classifications. Thus, System Two is far superior in its desired function compared to System One.

A direct limitation of the low cost equipment used in System One was not ascertained as the signal processing and BCI methodology were different. Whilst not directly comparable, the capabilities of System Three were also far more sophisticated than System One. The high cost of equipment and advanced measurement, filtering and processing functions allowed for accurate determination of limb position and movement from the EEG produced by a specific area in the brain. There are limitations of low cost EEG devices, such as a lower number of channels, higher signal to noise ratio and lower sampling frequencies in comparison to higher cost devices. However, the measured data can certainly still be useful in a variety of BCI and research applications.

5.2 - Recommendations

During this project, there were numerous challenges encountered. Some of these resulted in a large amount of time spent on work that was eventually not relevant to the final product, hindering the design process.

Initially, a lot of measurement and testing was performed with the Muse Headband being incorrectly worn which resulted in a high amount of artifacting, and inaccurate testing results which lead to poor design considerations. In future, it is advised to follow the manufacturer recommendations and perform testing on the accuracy of the data measured before completing any other design steps.

Inconsistencies with the physical location of the subject during data measurement and the positioning of the headband on the subject's head lead to incorrect classification. Thus, there was a lot of misdirected attention on modifying software code and debugging the classifier. This lead to the recalibration of the baseline and classifier data on every test and trial, which resolved the issues, however resulted in every test having to be performed again during the same session. For future works, calibration of EEG measurements should be performed every time the data is measured, and testing to be done in a single session if possible to reduce the effects of inconsistencies between trials.

During the state classification, there were unexpected results with the EEG spectral features when compared to findings from other works performed. The baseline data was suspected to be a main cause, where the subject may not have been in a neutral state. For further testing, it is recommended that the baseline data be measured over a period of minutes, rather than the thirty seconds measured in this project. Also, other works should be reviewed to identify the instructions given to the subject to ensure they remain in a neutral state effective for baseline calibration. Inaccurate elicitation of the subject's state during testing could have also caused the unexpected results. It was identified too late in the project development that other studies utilise a standardised way to elicit subject emotional states. This is known as the International Affective Picture System (IAPS), which is a set of pictures designed to form a standardised set in which to study emotional responses in subjects. The IAPS is recommended in any future works involving emotional responses and classification.

With regards to the accuracy of the testing, only one trial was performed with one subject for each test. Thus, the results from each test are likely to be highly inaccurate, and it is recommended for future to perform multiple trials on a group of subjects. This will result in a more accurate determination of the system's performance capabilities.

Finally, the majority of the time spent working on the project was involved with developing the software code to perform the low-level filtering and processing functions. It was later discovered that there are comprehensive packages available such as the MATLAB toolbox EEGLAB. EEGLAB contains a large amount of capability in the plotting, filtering and processing of EEG data. If such a package was used initially, the focus of the project could have been more directed at improving the functionalities of the BCI, or to develop more ambitious and sophisticated BCI applications. As a result, the functionality of the software prototype was kept very basic in order to complete the project within the time restraints.

5 - Conclusion

In conclusion, the objectives set out in the beginning of this project were achieved. A working prototype system was developed allowing a patient to communicate a 'Yes' or 'No' to others visually and auditorily using just EEG data. A basic method of determining the patient's emotional and mental state was also implemented into the software prototype, based on testing identifying relevant features of the EEG data. Finally, an analysis was performed determining the effectiveness of a low-cost system for data measurement and BCI application in comparison to two high cost systems. By doing this, the development process for a basic BCI is available for others to review, with recommendations for further works performed. Also, the working prototype is available to trial and use in any situation where a person does not retain the traditional means to communicate, however has standard EEG. Finally, it is concluded that a low-cost, consumer grade EEG device can be used for accurate measurement of brain activity. This data can then be used for a multitude of BCI applications or research purposes, and is not limited to those who can afford the tens of thousands of dollars for medical grade measurement equipment.

6 - References

[1] J. R. Duffy, *Motor Speech Disorders: Substrates, Differential Diagnosis, and Management*. Elsevier Health Sciences, 2012.

[2] G. Bauer, F. Gerstenbrand, and E. Rumpl, "Varieties of the locked-in syndrome," *J. Neurol.*, vol. 221, no. 2, pp. 77–91, Aug. 1979.

[3] B. A. Wilson, "What is Locked-In Syndrome? What causes it? Who is most at risk? Do people recover from this condition?," in *Locked-In Syndrome after Brain Damage*, pp. 1–4.

[4] M. O. Krucoff, S. Rahimpour, M. W. Slutzky, V. Reggie Edgerton, and D. A. Turner, "Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation," *Front. Neurosci.*, vol. 10, 2016.

[5] S. Herculano-Houzel, "The human brain in numbers: a linearly scaled-up primate brain," *Front. Hum. Neurosci.*, vol. 3, 2009.

[6] E. Niedermeyer and F. H. L. da Silva, *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.

[7] W. O. Tatum, IV, and DO, *Handbook of EEG Interpretation, Second Edition*. Demos Medical Publishing, 2014.

[8] G. B. Young and J. Mantia, "Continuous EEG monitoring in the intensive care unit," in *Handbook of Clinical Neurology*, 2017, pp. 107–116.

[9] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain," *Rev. Mod. Phys.*, vol. 65, no. 2, pp. 413–497, 1993.

[10] T. Mulholland, "Objective EEG Methods for Studying Covert Shifts of Visual Attention," in *The Psychophysiology of Thinking*, 1973, pp. 109–151.

[11] Interaxon, “The Muse Headband.” 2018.

[12] R. Carter, *The Human Brain Book*. DK Publishing (Dorling Kindersley), 2019.

[13] Nucleus Medical Art, “Anatomy and Functional Areas of the Brain.” 2018.

[14] J. W. Britton *et al.*, *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. Chicago: American Epilepsy Society, 2016.

[15] G. G. Knyazev, “Motivation, emotion, and their inhibitory control mirrored in brain oscillations,” *Neurosci. Biobehav. Rev.*, vol. 31, no. 3, pp. 377–395, 2007.

[16] M.-K. Kim, M. Kim, E. Oh, and S.-P. Kim, “A review on the computational methods for emotional state estimation from the human EEG,” *Comput. Math. Methods Med.*, vol. 2013, p. 573734, Mar. 2013.

[17] A. L. Anderson and M. E. Thomason, “Functional plasticity before the cradle: a review of neural functional imaging in the human fetus,” *Neurosci. Biobehav. Rev.*, vol. 37, no. 9 Pt B, pp. 2220–2232, Nov. 2013.

[18] F. Lotte *et al.*, “A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update,” *J. Neural Eng.*, vol. 15, no. 3, p. 031005, Jun. 2018.

[19] J. Kaiser *et al.*, “Self-initiation of EEG-based communication in paralyzed patients,” *Clin. Neurophysiol.*, vol. 112, no. 3, pp. 551–554, Mar. 2001.

[20] X. Hu *et al.*, “EEG Correlates of Ten Positive Emotions,” *Front. Hum. Neurosci.*, vol. 11, p. 26, Jan. 2017.

[21] M. Spüler, “A high-speed brain-computer interface (BCI) using dry EEG electrodes,” *PLoS One*, vol. 12, no. 2, p. e0172400, Feb. 2017.

[22] T. P. Luu, S. Nakagome, Y. He, and J. L. Contreras-Vidal, “Real-time EEG-based brain-computer interface to a virtual avatar enhances cortical involvement in human treadmill walking,” *Sci. Rep.*, vol. 7, no. 1, p. 8895, Aug. 2017.

[23] M. Spüler, A. Walter, W. Rosenstiel, and M. Bogdan, “Spatial filtering based on canonical correlation analysis for classification of evoked or event-related potentials in EEG data,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 6, pp. 1097–1103, Nov. 2014.

[24] J. Wu, J. Zhang, and L. Yao, “An automated detection and correction method of EOG artifacts in EEG-based BCI,” in *2009 ICME International Conference on Complex Medical Engineering*, 2009.

7 - Appendices

7.1 - Prototype User Manual

1. Turn on the Muse Headband and connect it to the computer with the following code entered into a terminal with the path set to the project folder:

   sudo python muse-lsl.py -n 'Muse-A145'

2. In another terminal with the path set to the project folder, run the following in order to view the live EEG data:

   python lsl-viewer.py

3. Place the Muse Headband on the subject's head and tighten the band. Monitor the live EEG data to ensure each channel has good connectivity of the electrodes. An indicator of good connectivity is if each electrode channel has a similar magnitude, and displays what looks like continuous, oscillating signals.

4. If it is desired to change the system parameters, the project code file *project.py* can be opened with a text editor and the parameters edited at the top of the file.

5. Open another terminal window and run the project with the following:

   python project.py

6. A prompt will ask if new calibration data is desired. This is recommended every time the headband is taken on and off the subject's head, or if the location of testing is changed. The program will output to the terminal, notifying of what data is being recorded. Two contrasting mental tasks will need to be recorded, used to control the classifier decision in outputting a 'Yes' or 'No' command when the main program starts. The subject's baseline state will also be recorded, as a reference for future state classification.

7. An application window will appear on screen, which serves as the visual display for the communication and patient's states. When ready, press enter in the terminal window to start classifying the patient's EEG data to determine the 'Yes' or 'No' state.

8. When the decision process has completed, the 'Yes' or 'No' determination will visually appear along with a text to speech audible output. The patient's mental and emotional state will also appear in the form of emoticons, if the patient's state changes from the baseline recorded previously.

7.2 - Program Code

7.2.1 - Supporting Functions

```
import numpy as np
import scipy.fftpack as fft
import scipy.signal as signal
from sklearn.preprocessing import minmax_scale
from sklearn import svm

def nextpowerof2(i): #calculate next power of 2 for optimal fft window size
    n=1
    while n<i:
        n*=2
    return n

def myfft(data,fs=256): #fft for single channel data
    ts=1/fs #sampling time interval
    n_samples,=data.shape #acquire number of samples
    ham=np.hamming(n_samples) #create hamming window
    data=(data.T*ham).T #apply hamming to data
    nfft=nextpowerof2(n_samples) #number of samples for fft in power of 2
    x=fft.rfftfreq(nfft,ts) #create frequency array
    y=fft.rfft(data,n=nfft,axis=0) #fft the eeg data
    y=y*2/n_samples #normalise by dividing by number of non zero samples and multiply by 2 for positive freqs only
    return x,y

def myffteeg(data,fs=256): #fft for 4 channel eeg data
    n_samples,n_channels=data.shape # number of samples and channels
    ham=np.hamming(n_samples) #create hamming window
    data=(data.T*ham).T #apply hamming to data
    ts=1/fs #sampling time
    nfft=nextpowerof2(n_samples) #next power of 2 for n samples for FFT optimisation
    x=fft.rfftfreq(nfft,ts) #create frequency array
    y=fft.rfft(data,n=nfft,axis=0) #fft the eeg data
    y=y*2/n_samples #normalise by dividing by number of non zero samples and multiply by 2 for positive freqs only
    return x,y

def bandstop(data,fl,fh,fs=256): #bandstop filter
    fl=fl/(fs/2) #normalise filter frequencies
    fh=fh/(fs/2)
    b,a=signal.butter(5,[fl,fh],'bandstop') #create filter coefficients
    out=signal.filtfilt(b,a,data) #filter signal with coefficients
    return out

def bandpass(data,fl,fh,fs=256): #bandstop filter
    fl=fl/(fs/2) #normalise filter frequencies
    fh=fh/(fs/2)
    b,a=signal.butter(5,[fl,fh],'bandpass') #create filter coefficients
    out=signal.filtfilt(b,a,data) #filter signal with coefficients
    return out

def lowpass(data,fc,fs=256):
    fc=fc/(fs/2) #normalise filter frequency
    b,a=signal.butter(5,fc) #create filter coefficients
    out=signal.filtfilt(b,a,data) #filter signal with coefficients
    return out

def filter_eeg(data, fs=256): #eeg filtering functions
    for i in range(len(data[0,:])):
        data[:,i]=signal.detrend(data[:,i]) #remove dc offset
        data[:,i]=bandstop(data[:,i],49,51) #Notch remove 50 Hz mains
        data[:,i]=bandpass(data[:,i],1,20) #only allow relevant frequencies through with bandpass
    return data
```

```
def bandpowers(data): #acquire absolute band powers from eeg data
    n_samples,n_channels=data.shape #get data array shape
    x,y=myffteeg(data) #fft data
    y=np.abs(y) #take abs to find power spectral density (PSD)
    ind_delta,=np.where((x>=1)&(x<=4)) #find indices in array for specified frequency bands
    ind_theta,=np.where((x>=4)&(x<=8))
    ind_alpha,=np.where((x>=8)&(x<=14))
    ind_beta,=np.where((x>=14)&(x<=20))
    mean_delta=np.mean(y[ind_delta,:],axis=0) #take mean of PSD at frequency bands
    mean_theta=np.mean(y[ind_theta,:],axis=0)
    mean_alpha=np.mean(y[ind_alpha,:],axis=0)
    mean_beta=np.mean(y[ind_beta,:],axis=0)
    bandpowers=np.concatenate((mean_delta,mean_theta,mean_alpha,mean_beta),axis=0) #create band powers vector
    absbandpowers=np.log10(bandpowers) #apply to log scale for abs band powers
    return absbandpowers

def relbandpowers(data): #acquire relative bandpowers from absolute band powers
    n_bands,n_channels=data.shape #get data shape
    output=np.zeros(data.shape) #initialise vector with zeros
    data=10**data #inverse log of abs band powers
    data=minmax_scale(data,axis=0)
    for i in range(n_channels): #calculate relative band powers
        sumpowers=0
        for z in range(n_bands):
            sumpowers+=(data[z,i])
        for y in range(n_bands):
            output[y,i]=data[y,i]/sumpowers
    return output

def epochs(data,samples_epoch,samples_overlap):
    n_samples,n_channels=data.shape #find number of samples and channels
    samples_shift=samples_epoch-samples_overlap #determine shift length of samples
    n_epochs=int(np.floor((n_samples-samples_epoch)/float(samples_shift))+1) #determine epoch number
    markers=np.asarray(range(0,n_epochs+1))*samples_shift #create markers for epoch locations
    markers=markers.astype(int) #make values integers
    epochs=np.zeros((samples_epoch,n_channels,n_epochs)) #initiate epoch matrix with zeros
    for i in range(0,n_epochs):
        epochs[:,:,i]=data[markers[i]:markers[i]+samples_epoch,:] #fill epoch matrix with data
    return epochs

def bandpowers_epoch(epochs):
    n_epochs = epochs.shape[2] #determine epoch number
    feat= bandpowers(epochs[:,:,0]) #initialise with matrix with number of epochs and features
    feat = feat.T
    bandpowers_epoch = np.zeros((n_epochs, feat.shape[0]))
    for i in range(n_epochs):
        bp=bandpowers(epochs[:, :, i]) #create bandpowers for epoch matrix
        bandpowers_epoch[i, :] = bp
    return bandpowers_epoch

def trainclassifier(feature_matrix_0, feature_matrix_1, algorithm='SVM'):
    class0 = np.zeros((feature_matrix_0.shape[0], 1)) #fill class vectors with 0 and 1
    class1 = np.ones((feature_matrix_1.shape[0], 1))
    y = np.concatenate((class0, class1), axis=0)  # concatenate features and labels
    y = y.ravel()
    features_all = np.concatenate((feature_matrix_0, feature_matrix_1),axis=0)
    mu= np.mean(features_all, axis=0)
    std= np.std(features_all, axis=0)
    X = (features_all - mu) / std #normalise with mean and std
    clf=svm.SVC()
    clf.fit(X,y) #train SVM
    score = clf.score(X,y) #acquire classifier score
    return clf,score,mu,std

def update_buffer(data_buffer, new_data): #update data buffer
```

```
    if new_data.ndim == 1:
        new_data = new_data.reshape(-1, data_buffer.shape[1])
    new_buffer = np.concatenate((data_buffer, new_data), axis=0)
    new_buffer = new_buffer[new_data.shape[0]:, :]
    return new_buffer

def get_last_data(data_buffer, epoch_window): #return epoch from buffer
    epoch = data_buffer[(data_buffer.shape[0] - epoch_window):, :]
    return epoch

def classify(clf, feature_vector, mu, std): #classify new data
    X = (feature_vector - mu) / std #normalise with normalisation mean and std
    decision = clf.predict(X)
    return decision
```

7.2.2 - Main Program

```
#import external functions
import numpy as np # Matrices Calculation Module
from pylsl import StreamInlet, resolve_byprop  # EEG Comms Module
import pyttsx3 #text to speech
import time #time module
import pygame #visual outputs
from threading import Thread #multi threading
import myfunc as f #my written functions
import save_eeg_data #to measure and save eeg data
import scipy #python science functions

"""Define Functions"""
def speak(text): #speak string
    engine.say(text)
    engine.runAndWait()

def text_objects(text, font): #create text object
    textSurface = font.render(text, True, black)
    return textSurface, textSurface.get_rect()

def message_display(text,emotion,mentality):
    gameDisplay.fill(white) #fill screen with white
    pygame.display.update() #reset display

    largeText = pygame.font.Font('freesansbold.ttf',115) #create main text
    TextSurf, TextRect = text_objects(text, largeText)
    TextRect.center = ((display_width/2),(display_height/4))

    smallText = pygame.font.Font('freesansbold.ttf',50) #create image titles
    TextSurfEmot, TextRectEmot = text_objects('Emotional State', smallText)
    TextRectEmot.center=((3*display_width/4),(display_height/2))
    TextSurfMent, TextRectMent = text_objects('Mental State', smallText)
    TextRectMent.center=((display_width/4),(display_height/2))

    if isinstance(mentality,str):
        mentality_image=pygame.image.load('png/'+mentality+'.png') #create mental image
        mentality_image = pygame.transform.scale(mentality_image, (300, 300))
        mentality_image_rect=mentality_image.get_rect()
        mentality_image_rect.center=((1*display_width/4),(3*display_height/4))
        gameDisplay.blit(mentality_image,mentality_image_rect)

    if isinstance(emotion,str):
        emotion_image = pygame.image.load('png/'+emotion+'.png') #create emotion image
        emotion_image = pygame.transform.scale(emotion_image, (300, 300))
        emotion_image_rect=emotion_image.get_rect()
        emotion_image_rect.center=((3*display_width/4),(3*display_height/4))
```

```
        gameDisplay.blit(emotion_image,emotion_image_rect)

    gameDisplay.blit(TextSurf, TextRect) #display all graphics
    gameDisplay.blit(TextSurfEmot,TextRectEmot)
    gameDisplay.blit(TextSurfMent,TextRectMent)

    pygame.display.update() #refresh display window

def init_decision():
    init_decision.quit=None
    init_decision.init=None
    while(1):
        #wait for keyboard input
        if input('Press any key to start processing patient decision. Type "quit" to exit.\n')=='quit':
            init_decision.quit=True
        else:
            init_decision.init=True

#set parameters
display_width = 1200
display_height = 800
buffer_length=15
epoch_length=1
overlap_length = 0.8
training_time=10
decision_time=7
shift_length=epoch_length-overlap_length
mental_threshold=0.1
emotion_threshold=0.1
fs=256
index_channel=[0,1,2,3]
n_channels=4
ch_names=['TP9','Fp1','Fp2','TP10']
bands = ['delta', 'theta', 'alpha', 'beta','gamma','mu']
feature_names = []

#Ask to acquire training data
while True:
    train_data=input('Do you want to train new classifier and baseline data? (yes/no)\n')
    if train_data=='yes':
            save_eeg_data.main('eeg_data0',training_time)
            save_eeg_data.main('eeg_data1',training_time)
            save_eeg_data.main('baseline',training_time)
            break
    elif train_data=='no':
        break
    print('Invalid selection, please try again...')

#train classifier
#Load Saved State Data
eeg_data0=np.loadtxt('eeg_data0')
eeg_data1=np.loadtxt('eeg_data1')
#filter data
eeg_data0=f.filter_eeg(eeg_data0)
eeg_data1=f.filter_eeg(eeg_data1)
#Divide data into epochs
eeg_epochs0 = f.epochs(eeg_data0, epoch_length * fs,overlap_length * fs)
eeg_epochs1 = f.epochs(eeg_data1, epoch_length * fs,overlap_length * fs)
#Compute Features and Train Classifier
feat_matrix0 = f.bandpowers_epoch(eeg_epochs0)
feat_matrix1 = f.bandpowers_epoch(eeg_epochs1)
classifier,score,mu,std = f.trainclassifier(feat_matrix0, feat_matrix1)

print('Classifier Trained with a score of %d%%.' %(score*100))

#Establish Baseline
```

```
baseline=np.loadtxt('baseline')
baseline=f.filter_eeg(baseline)
baseline_features=f.bandpowers(baseline)
baseline_features=baseline_features.reshape(-1,n_channels)

# Initialise Communication and search for EEG Stream
print('Initialise and wait for EEG Stream...')
streams = resolve_byprop('type', 'EEG', timeout=2)
if len(streams) == 0:
    raise RuntimeError('Can\'t find EEG stream.')

# Receive data stream, Perform time correction to match local clock
print("Start receiving data stream and perform time correction")
inlet = StreamInlet(streams[0], max_chunklen=12)
eeg_time_correction = inlet.time_correction()

#Initialize raw EEG Data buffer
eeg_buffer=np.zeros((int(fs*buffer_length),n_channels))

#Initialise display and audio output
engine=pyttsx3.init()
pygame.init()
black = (0,0,0)
white = (255,255,255)
gameDisplay = pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption('Communication Outputs and Emotional State')
gameDisplay.fill(white)
pygame.display.update()

mentals=['sleep','tired','relaxed','focus']
emotions=['sad','happy']

try:
    decision_array=[]
    deciding=None
    Thread(target=init_decision).start()
    while init_decision.quit!=True:
        #Obtain EEG Data from LSL Stream
        eeg_data,timestamp=inlet.pull_chunk(timeout=1,max_samples=int(shift_length*fs))
        eeg_data = np.array(eeg_data)[:, index_channel]
        eeg_data=f.filter_eeg(eeg_data) #filter eeg data
        eeg_buffer=f.update_buffer(eeg_buffer,eeg_data)
        #Get New samples
        data_epoch=f.get_last_data(eeg_buffer,epoch_length*fs)
        #Compute Features
        feat_vector=f.bandpowers(data_epoch)
        #Receive Classifier Decision
        decision=f.classify(classifier,feat_vector.reshape(1,-1),mu,std)

        emotion=None #reset emotional and mental state
        mental=None
        feat_vector_stacked=feat_vector.reshape(-1,n_channels)  #rearrange feature vector
        mental_dif=np.mean((feat_vector_stacked-baseline_features)/baseline_features,axis=1) #find mental and emotional feature differences
        emotion_dif=((feat_vector_stacked[2,1:2]-baseline_features[2,1:2])/baseline_features[2,1:2]).reshape(-1,1)
        mentalmax=np.argmax(mental_dif,axis=0) #find max mental state position
        mentalmaxval=mental_dif[mentalmax] #find max mental state value
        emotionmax=np.argmax(emotion_dif,axis=0) #find max mental state position
        emotionmax=emotionmax[0]
        emotionmaxval=emotion_dif[emotionmax]  #find max mental state value

        if emotionmaxval>emotion_threshold: #if over threshld set states
            emotion=emotions[emotionmax]
        if mentalmaxval>mental_threshold:
            mental=mentals[mentalmax]
        if init_decision.init==True: # if user initiates deciding sequence, start deciding
            deciding=True
```

```
            end_time=time.time()+decision_time
            prev_time=decision_time+1
            init_decision.init=False
        if deciding==True:
            decision_array.append(decision)
            #If not time to make a decision
            if (prev_time - (end_time-time.time()))>=1:
                message_display('Deciding in ' + str(prev_time-1) + '...',emotion,mental)
                prev_time-=1
            #if time to make a decision, take average decision, output it then reset timer and decision values
            if time.time()>=end_time:
                decision,_=scipy.stats.mode(decision_array)
                if decision == 1:
                    message_display('Yes.',emotion,mental)
                    speak('Yes.')
                elif decision == 0:
                    message_display('No.',emotion,mental)
                    speak('No.')
                deciding=False
                decision_array=[]
        else:
            message_display('',emotion,mental)

except KeyboardInterrupt:
    print('Closing')
    time.sleep(1)
    quit()
```