Controlling Home Appliances through Thought Commands

K.A.T.A. Karunarathne
Index No: 14000628

Supervisor: Dr H.E.M.H.B.Ekanayake

January 2019

Submitted in partial fulfillment of the requirements of the B.Sc in Computer Science Final Year Project (SCS4124)

Controlling Home Appliances through Thought Commands

K.A.T.A. Karunarathne

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: Tharinda Karunarathne

……………………………………………….

Signature of Candidate   Date:

This is to certify that this dissertation is based on the work of

Mr. Tharinda Karunarathne

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle Supervisor's Name: Dr. Hiran Ekanayake

……………………………………………….

Signature of Supervisor   Date:

Abstract

Brain-Computer Interfaces (BCI) is a field which has shown rapid advancement over the past few decades. With the availability of low-cost electroencephalography (EEG) signal acquisition devices, it is becoming more feasible to develop hands-free environmental control systems with significant accuracies, which can be used for everyday use. These systems may be used as assistive technologies for disabled individuals or as alternative forms of control for healthy individuals. Hands-free environmental control systems often lack user-friendliness and intuitiveness due to the difficulty of mapping the users' intentions to control commands that can be used to control the appliances in a three dimensional environment. Therefore, this research has focused on developing a hands-free environmental control system based on P300 responses of EEG signals of a user. The developed solution is comparatively less disturbing for the user and does not use a screen for visual feedback; therefore, it improves the intuitiveness and user friendliness than existing solutions. Additionally, the system has achieved better than state-of-the-art selection times, high accuracies with 100% accuracy for multiple subjects and highly stable results for multiple subjects.

Preface

The new method of issuing control commands with an LED stimulator introduced by this research is a concept refined by me in conjunction with my supervisor. The data logging application, device synchronization, signal processing automation, automated classifier, prototype headband and model appliance controlling techniques are entirely my own work. The results of Results chapter rely upon experiments conducted by me with help of the UCSC Electronics Lab staff.

Acknowledgement

Special thanks go to my supervisor, Dr. Hiran Ekanayake, senior lecturer of University of Colombo School of Computing for guiding me on the topic and for the immense support and invaluable advices given to me throughout the research.

I would like to thank my advisor, Mr. Dushan Dinushka, instructor of University of Colombo School of Computing for supporting me throughout my research, the University of Colombo School of Computing Electronics Lab for supplying me with equipment required for the work and the staff at Electronics Lab for helping me conduct the required experiments. And many thanks to Mr. Turrance Nandasara, Mrs. Harshi Abeyratne and the staff at University of Colombo School of Computing Digital Media Studio for lending us the studio to conduct the experiments.

I would also like to thank Prof. Thashi Chang, Senior Lecturer of Faculty of Medicine, University of Colombo, and my sister, Dr. Sasmitha Karunarathne, Resident House Officer at The National Hospital of Sri Lanka, Colombo, for advising me on the medical, neurological and psychological aspects of this work.

I appreciate the feedback and motivation provided by my friends to achieve my research goals and I also would like to appreciate the support of all the volunteers who helped me for the user evaluation of the research.

Many thanks to my beloved mother and my dear father for always being my strength, showing me the correct direction, and helping me succeed many challenges.

Table of Contents

List of Figures

List of Tables

List of Acronyms

BCI	Brain Computer Interface
EEG	Electroencephalography
ADC	Analog to Digital Converter
ERP	Event Related Potential
IR	Infrared
ISI	Inter−stimulus Interval
IO	Input/output
RX	Receiver
USB	Universal Serial Bus
ITR	Information Transfer Rate
ALS	Amyotrophic Lateral Sclerosis

Chapter 1 - Introduction

A Brain−Computer Interface (BCI) is a direct communication pathway between an enhanced or wired brain and an external device. A BCI can record electrical activity signals from the brain and classify them into different states which can then be interpreted for communication. BCI research includes 3 main categories:

- Invasive BCI – Requires the devices to be implanted into the brain. These produce the highest quality signals of BCI devices.
- Partially Invasive BCI – the devices are implanted inside the skull, but outside the brain.
- Non−invasive BCI – the devices are connected outside of the skull and onto the scalp. These devices produce lower quality signals than Invasive BCI, but are less complex to apply. Majority of the BCI research and this work focus on this category.

Since BCI does not require muscle movements for communication, it can ideally provide an effective means of communication even for entirely paralyzed people [10]. BCI research began in the 70s and has shown rapid advancement over the last few decades.

1.1 Background to the Research

BCI research often aims for augmenting, or repairing human cognitive or sensory−motor functions. Currently, Non−invasive BCI devices can be used for applications such as moving a cursor on a computer screen [11], controlling home appliances [2], wheelchair control [9], recreational use [17, 18], and for spelling purposes [3]. When considering the existing applications of BCIs for environmental control, there is a noticeable lack of applications where the user's orientation in the 3D environment is taken into consideration. Several similar works [2, 8] have developed environmental control systems that use a display screen for visual feedback. In this work, the need of a screen was eliminated and the user's orientation in the 3D environment was used to create a more engaging and intuitive experience for the user.

1.2 Research Problem and Research Questions

The main research question that is being answered by this research is:

- How to control appliances in an enclosed physical environment using brain potentials with low cost, significant accuracy and stable results in a less disturbing way for the user?

Using brain potentials for control purposes is somewhat challenging, since EEG signals can only be interpreted into a limited number of discrete patterns. In order to overcome this challenge, the P300 method is being used. For the appliance selection task, a system that takes advantage of the positioning of various appliances in a 3D space has to be developed. This will be further explained in following sections. There already exist many BCIs developed with research and medical–grade EEG devices. While these BCIs can achieve better accuracies, they are not very effective to be used in an environmental control system that is intended for everyday use due to the high costs of the device, cost of using due to consumable electrodes and electrolytes and the difficulty of use and discomfort due to complexity of the gear that is worn on the head. Using low–cost devices make it easier for the users to adopt a BCI system for everyday use.

1.3 Research Aim and Objectives

The definitive aim of this project is to find a low–cost solution for communication that does not significantly depend on the users' muscle movements, and enable hands–free control of the appliances in user's environment. The following objectives will be achieved by this project.

1. Evaluate existing low–cost biophysical signal acquisition devices for their potential for capturing p300 brain responses.

   The reason for selecting low–cost devices for this research was because using low–cost devices for environmental control is more feasible since majority of the targeted users will not be able to afford a research grade or medical grade BCI device for the purpose of hands–free environmental control if this solution were to be released to the consumer market.

   The low–cost devices that have been tested are

   - BITalino Biomedical Sensor Kit – $200
   - Emotiv EPOC+ 14 Channel EEG Headset – $800

   These devices are significantly cheaper than the research or medical grade signal acquisition devices such as

   - g.USBamp Research Grade USB Bio–signal Amplifier – $13,200 [36]
   - g.USBamp Medical Grade USB Bio–signal Amplifier – $23,200

   The P300 response was chosen to benchmark these devices since P300 can only be observed if the device is capable enough to pick up true EEG signals and the procedures of conducting experiments (including synchronizing up to millisecond accuracy) and analyzing data are all accurate.

2. Develop a hardware interface to integrate biophysical data acquisition and appliance control.

   A hardware interface is required to control the various appliances using the biophysical signals that are acquired. The Arduino platform was chosen to accomplish the tasks of appliance control and providing visual stimuli and feedback for the user. The data acquisition and hardware control are integrated and synchronized to millisecond accuracies using a program developed in C++.

3. Classify biophysical signals captured from the brain to identify different control events.

   In order to interpret the acquired biophysical signals as control commands, the acquired signals need to be classified into classes of various commands. This involves signal processing, artefact removal and signal averaging.

4. Automate the processes to enable real–time control of appliances.

   The artefact removal and classification can be easily done manually by observing the graphical representations of the signal. But in order to run the system in real–time without human intervention, the entire process needs to be automated.

5. Evaluate the accuracy of solution under different conditions of users and the environment and other parameters.

   The accuracy of the system varies with the different conditions of the environment, such as external noises, distractions, lighting, and states of the user such as sleep deprivation and fatigue. The accuracy can also depend on parameters related to the experiments such as number of trials per selection, inter–stimulus interval, indicator flash duration, flash color and the classifier used. Therefore the system has to be tested by changing these parameters to improve the accuracy of control.

1.4 Justification for the research

Current environmental control systems that are already in existence use a control mask: a grid of flashing symbols in order to control the appliances. The approach followed in this research eliminates the need for such control masks and makes it possible for the user to control the appliances in the environment directly by looking at them. Experiments on the effect the color of the flashing stimulators have on the users' brain responses, the effect of various distances between stimulators, and also the distance between user and the stimulators have also been conducted. The usage of LED stimulators instead of the control mask is not a trivial change of the experiment setup, since the LED flashing involves hardware development, embedded programming and finally synchronizing the flashing controller with a computer program with milliseconds of precision.

1.5 Methodology

It has already been observed by research that the P300 response; the main BCI paradigm that has been used to build this system, is observable with low cost BCI devices [19]. And the P300 response is elicited when user encounters an unexpected stimulus in the middle of frequent stimuli. When compared with other types of P300 responses such as auditory P300, the visual P300 is rather strong and more prominent. The P300 response can also be observed for single stimulus experiments (presenting only the target stimulus in the absence of other stimuli) [33] which imply that the non-target indicators do not always require being in the user's field of view in our proposed method. Therefore it was assumed that a P300 response that would be observable with low-cost devices could be generated with the indicator panel technique that was followed in this research. By evaluation with multiple test subjects, we have proven that this assumption is correct (see Results chapter).

This research is mainly an experimental research. We have conducted experiments throughout the course of this research and collected quantitative data in order to evaluate the accuracy, classification speed, and information transfer rate of the system.

Once the system was finalized, the system was tested on 11 subjects in total. The test consisted of control tasks that the user should perform where the user was required to change the speed of a four speed fan. Additional data such as movement of the device, eye movements and eye blinks, were collected during the experiments, which helped in artefact removal in EEG signals. These additional data were used to give important insights to our experiments.

Subjects' feedbacks were recorded by survey forms which collected qualitative data, where the subjects were asked about the ease of use, any discomfort experienced and overall satisfaction of the system.

There are many reasons to take special care when conducting BCI experiments. Since EEG signals consist of very low voltages (in microvolts range), the signals recorded in the experiments are easily prone to noise. Therefore the experiments should be performed in an environment with low electrical noise, and the subject and the signal acquisition device should be properly grounded. As a safety measure, the devices that were used were battery powered signal acquisition devices which were powered by small 3.7V Lithium–Ion batteries. This prevents electric shocks since it's critical to eliminate the risk of an electric shock, since the electrodes are directly attached to the users' head through low impedance contact regions.

The subject's attention plays an important role in the P300 experiments since the P300 response is entirely based on the user paying attention to observing the target stimuli. Therefore, all unwanted distractions were avoided during the experiments. A silent studio environment was chosen to achieve this. The subjects were given breaks to avoid fatigue since it also could affect the results.

1.6 Outline of the Dissertation

Chapter 1 contains the introduction and the background for the research. A literature review and information about theories related to BCIs are included in Chapter 2. Chapter 3 contains the design of the research while the Chapter 4 contains its implementation. The results of the experiments and evaluation are provided in Chapter 5. Chapter 6 contains the conclusion for the dissertation.

1.7 Definitions

Disabled User

A person that is suffering from a neurological disease or condition such as (Amyotrophic lateral sclerosis (ALS), brainstem stroke, brain or spinal cord injury, cerebral palsy, muscular dystrophies, multiple sclerosis etc.), who is using an environmental control system to control the appliances in their environment. These individuals can be completely "locked–in" to their bodies which can be a distressing experience for them.

Healthy User

A person who uses an environmental control system that does not suffer from neurological diseases or conditions that might hinder their normal movement of hands and feet. These individuals may use assistive technologies to avoid the fatigue that is caused by other manual control systems or because they are engaged in activities that hinder the normal movement of the hands.

Thought Commands

Any type of communication made from the brain to an external device that is based on a psychological state of the user's mind, which can be classified into or interpreted as control commands. These commands may be used to control things in the environment as well as to communicate other thoughts or intentions to the external world if the user's normal modes of communication are not functioning properly.

Home Appliances

Appliances that are used in home environments such as televisions, air conditioners, fans, lights, sound systems, phones and intercoms which normally require conventional means of control such as button based interfaces or remote controllers.

1.8 Delimitations of Scope

The performance of the system might depend on several variables since it strongly depends on users' attention. For this to be possible, the appliance requires being in line of sight of the user, and should be close enough for the user to concentrate on the indicator light. Outdoor environments might be unsuitable for this since flashing indicators might not be clearly noticeable in daylight. It might also be difficult for the user to concentrate on the indicator if the user is in motion. Therefore a relaxed home or office environment might be best suited for this system.

- The system has not been tested with actual disabled or locked–in individuals due to ethical reasons. But according to expert opinions, the system should be useable by disabled individuals suffering from several known neurological diseases (see Results and Evaluation chapter).
- The system has not been developed to be used with multiple rooms or while the user is moving.
- The system should only be used indoors.
- The appliance that needs to be controlled should be in line of sight of the user.

This project focuses more on efficiently communicating user's intention to the system than controlling the actual appliances. Controlling of the appliance is a lot less complicated task, which can be achieved by simply connecting the system with an IR remote, or relays when compared to signal acquisition, processing and classification of signals into control commands.

1.9 Conclusion

This chapter laid the foundations for the dissertation. It introduced the research problem and research questions and hypotheses. Then the research was justified, definitions were presented, the methodology was briefly described and justified, the dissertation was outlined, and the limitations were given. On these foundations, the dissertation can proceed with a detailed description of the research.

Chapter 2 - Literature Review

2.1. Assistive Technologies and Environmental Control Systems

The following section is a literature review of environmental control systems that attempt to solve the problems mentioned above. Here I have categorized the related works according to type of control. Each type has its advantages and drawbacks with respect to healthy and disabled individuals.

2.1.1. Manual Remote Controls

Remote controls are an effective means of environmental control for healthy users which eliminate the need to physically walk to or reach the appliance. Majority of the users are already familiar with remote controls and they are highly accurate.
However this solution may not be effective for disabled individuals, since the controller often needs to be pointed at the appliance and some users have difficulty when operating the buttons on the controller. [13]

2.1.2. Mechanical Switches

This type of solution is a switch that opens or closes which is controlled by explicit physical movement. The switch may respond to a specific mechanical stimulus, including changes in displacement, tilt, air pressure (e.g., sip and puff), or force [10]. This type of interface can be operated by people with control of only one part of body. The switch should be used in a way that utilizes the existing neural pathways for disabled individuals. A similar interface was used by the famous cosmologist Stephen Hawking to control his computer which used only the movement of his cheek [12]. This solution is cheap and has a high accuracy since it uses explicit physical movement.

A drawback of this solution is fatigue caused by repeated movement of the same part of the body [10]. These switches have limited control channels which make it ineffective for the use of healthy users since far better alternatives exist.

2.1.3. Hand Gesture based Control

Hand gestures provide an intuitive way of interfacing with the world. And can form complex methods of communication such as sign language used by people with hearing and speech disabilities. This technique can be used to control home appliances such as TV by using several commands represented by signs and movements of hands. These movements can be picked up by an Infrared camera and processed and interpreted as control commands [14].

This method requires the users to be able to move their hands, which makes this an ineffective solution for disabled individuals who cannot move their hands. When this method was used by disabled individuals who still possessed limited movement capabilities, they could only use it for short durations since they experienced fatigue [15].

2.1.4. Voice Control

Voice controls are a popular method of environmental control that is already being used for many applications. There are several commercial home automation systems offered by major tech companies like Amazon Echo, Apple HomePod and Google Home. Voice controls require no physical movement and it eliminates the need to reach or walk to the appliances to control them.

Even though voice controls have these benefits it they are not widely used for assistive devices for several reasons. The user requirements for people with disabilities often have high levels of variation which creates a high cost for individual adaptation and development. Users who might benefit form voice commands often also have speech difficulties such that speech recognizers are unusable for them [16]. And some research results show that the accuracies can be significantly affected by noise in the environment [20].

2.1.5. Gaze Based and Eye–Tracking Control

Gaze–based communication can map eye movement to a cursor position on the screen. The dominant technologies in commercially available eye trackers are video–oculography (VOG) and electro–oculography (EOG). VOG–based approaches typically use an IR light source and a camera mounted on a computer display. Gaze direction is calculated by the offset between the corneal reflection and pupil centre [10]. EOG–based systems place electrodes around eyes to measure shifts in potential difference between cornea and retina that occur when user changes gaze direction [8].

Although eye tracking controls have speeds comparable to a hand–mouse, productivity in computer tasks is lower in practice. Since this input method uses the same channel for control and observation, there is no intuitive means of differentiating between an input command and a user activity. Gaze controlled devices can also have drawbacks like calibration drift, user fatigue and insufficient range of motion of the eye [10].

2.1.6. Electroencephalography (EEG) and BCI Solutions

EEG based devices can provide a non–muscular form of control from the electrical activity measured on scalp. Since this does not require significant muscle movement, it is usable by tetraplegic disabled individuals as well [9]. Present day BCIs can be divided into two main categories based on the type of signals extracted: consciously modulated spontaneous rhythms or evoked potentials.

The first category of BCI uses potentials that can be intentionally modulated by the user with some training. A research by Wolpaw et al [11] that used sensorimotor rhythms (SMRs) has shown that individuals with severe motor disabilities are able to control a cursor on a screen with two dimensional control signal using noninvasive BCI with accuracies up to 92% and relatively short response times [11]. In their research, vertical movement was controlled by a 24–Hz beta rhythm and horizontal movement by a 12–Hz mu rhythm. But however this type of BCI requires some training to improve the reliability.

The second category relies on responses to external stimuli such as visual stimulus like flashing of an indicator. Steady–state visual evoked potential (SSVEP) is an example of this.

An SSVEP based BCI research conducted by Chen et al [18] have developed a recreational device that controls a toy fish using brain signals. The signals with high energy levels at different frequencies are produced when the user looks at control commands displayed on a screen that are flashing at the corresponding frequency. An average classification accuracy of 89.51% was obtained for their research.

2.1.7. P300 Based BCI

Another type of signal that belongs to the evoked potential category is the P300 response. P300 potentials are positive deflections in EEG signal which were elicited approximately 300ms after encountering an intended stimulus among a group of irrelevant stimuli. This response can be used to create a BCI since it can identify which stimulus the user was concentrating on by analysing the EEG signals.

Since the P300 potentials have very low voltages (2–5μV) and are hidden within the EEG noise, they are not directly visible in an EEG recording [19]. In order to view these responses, the raw EEG signal should be band–passed (at around 1–20Hz), intervals of signal in multiple trials (called epochs, around −1000 to 2000ms relative to the stimulus) that are time–locked to the stimulus should be extracted, and then the epochs should be averaged (Fig. 1 and 2). This process averages out the noise voltages while accumulating the psychological properties of the stimulus, improving the Signal to Noise Ratio (SNR) of the response [25].

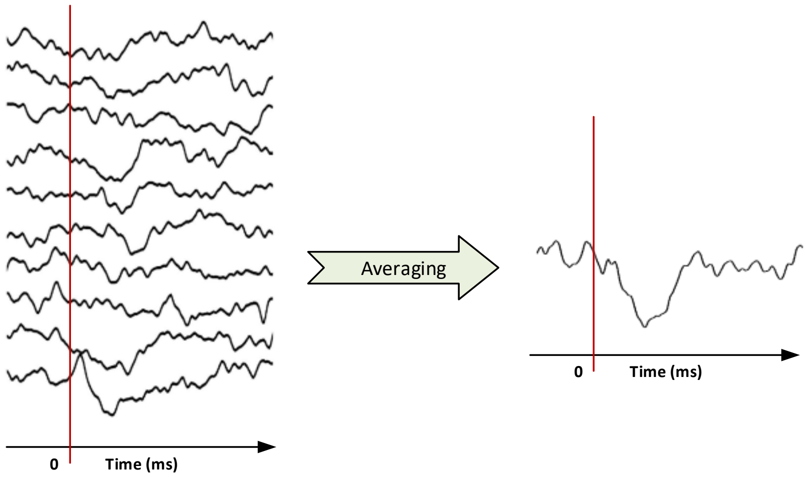


Figure 2.1 The EEG signal intervals from multiple trials (left) time−locked to the event (red line) and after averaging them (right) which shows a deflection common to all the trials while cancelling out the noise.

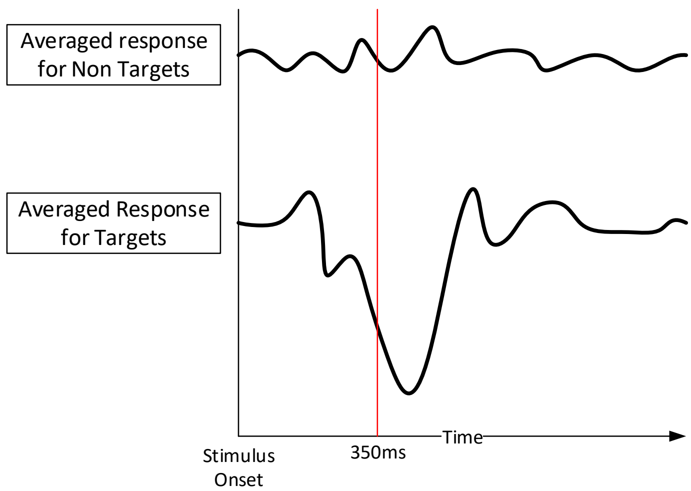


Figure 2.2 Averaged EEG signal of multiple trials for non−target responses and target responses

P300 response can be obtained by flashing columns and rows for spelling purposes [3]. A P300 based solution for smart home control developed by Holzner et al [2] has shown the suitability of the P300 response for selection of control commands. Their research used control masks for the selection of areas inside the home and to control various appliances such as a phone and a TV. Accuracies up to 100% have been achieved in their research for some control masks.

Another research by Cristian et al [8] which used a hybrid BCI approach that combined EEG and EOG signals has studied how the reliability of an EEG system can be improved by combining it with another control channel. In their research the user had the ability to cancel the selection made by the EEG BCI by changing the direction of his gaze.

2.1.8. The Types of Devices Used For EEG Signal Acquisition

The main focus of this research is finding a low cost solution for the problem of controlling home appliances through brain computer interfacing. Therefore the cost of the devices becomes an important concern when developing the ideal solution.

The devices can be separated into two main categories; consumer grade devices and research grade devices. Two signal acquisition devices have been evaluated in this research with respect to the quality of the signals obtained. Namely, BITalino biomedical signal acquisition device and Emotiv EPOC.

2.1.9. Types of Stimulus Generators Used

There are different types of stimulus generators used in BCIs. In works like [1] the visual stimuli are generated by a set of flashing symbols on a screen. This method of stimulus generation is different from the LED indicator stimulator introduced in this work since the flashing symbols on screen also incorporate the characteristic shape, which aids the visual separation of the symbols from each other. In order to enhance the visual separation of the LED indicators, some techniques like using different colored LEDs or increased distance between indicators have to be used.

2.1.10. Classifiers for P300 Detection

P300 response can be classified mainly by neural networks or by model−based classifiers. Due to the large number of training data points required to train a neural network based classifier, the users might experience fatigue when completing the training process of the system, since the parameters of the classifier has to be customized for each user. Another drawback of applying a neural network for this system is the opaque or "black−box" nature of neural networks. The classifier might train on some unexpected features of the input signals and may not train for the most prominent feature, which is the characteristic P300 peak of the averaged signal. Due to these reasons, it was decided to use a model based classifier for P300 classification of the system.

In the initial work of P300 speller by Farwell and Donchin [39] they have used four model−based techniques for classifcation purposes.

- Area – Calculates the area under the curve within the predefinced P300 window.
- Peak picking – The difference between the highest positive peak within and the lowest negative point prior to the P300 window.
- Stepwise Discriminant Analysis (SWDA) – Computes the distance of an epoch to the mean of a group containing P300 epochs as calculated from the training set. This score is obtained by applying a discriminant function to the data from each epoch.
- Covariance – Assesses the covariance of an epoch with a template. The template is calculated as the average of epochs belonging to attended symbols in the training set.

Selection times to reach certain accuracies by those classifiers in Farwell and Donchin work [39] are as follows.

Method	Subject	80% Accuracy		95% Accuracy	
		125ms ISI	500ms ISI	125ms ISI	500ms ISI
Area	1	29.1	39.9	76.7	59.3
	2	49.0	56.6	–	–
	3	29.3	12.6	55.8	17.9
	4	45.5	44.9	82.2	52.9
Peak Picking	1	–	28.2	–	42.5
	2	–	23.3	–	35.5
	3	39.8	17.3	–	26.0
	4	38.8	17.7	70.4	29.3
SWDA	1	15.7	114.8	21.6	202.8
	2	33.4	56.9	57.5	–
	3	22.3	11.1	46.4	17.6
	4	54.4	26.7	–	49.5
Covariance	1	–	42.9	–	62.4
	2	–	–	–	–
	3	41.8	15.5	82.2	22.6
	4	36.7	28.6	64.0	52.0

Table 2.1 Required time in seconds to reach either 80% accuracy or 95% accuracy for different ISIs, classification techniques and subjects in Farwell and Donchin's work

Several recently conducted researches have explored the suitability of Asynchronous BCI (A−BCI) for environmental control [21, 22]. Synchronous BCIs assume that the user is continuously controlling the system. They cannot detect whether the user is paying attention to the system or not and therefore they make incorrect selections when the user stops paying attention to the system. A synchronous P300 BCI makes a selection after a fixed number of trials regardless of whether the user was paying attention to the system or not.

In contrast to this, asynchronous BCIs make selections only when the response in the signals passes a threshold level [21]. Therefore they can determine whether the user is focusing on the system or not and switch between control and no−control states accordingly. This type of a system can be more practical than synchronous systems since the user can focus on other tasks without the need of notifying the system. However, A−BCIs require a specialized algorithm or a support vector machine (SVM) to analyze and discriminate between the target P300 epochs and non−target epochs with respect to the threshold level [22].

2.1.11. Hardware Interfaces and Protocols for Home Appliance Control

Infrared remotes can be simulated with IR blasters. IR blasters are infrared emitters that emit infrared in multiple directions. These can be used to control appliances that have IR remote control interfaces. Another such protocol for appliance control is the X10 power line communication (PLC) standard [38]. This is a protocol which transmits radio frequency waves through power lines which was developed in 1975 for communication among devices for home automation.

2.2. Theories Related to BCI

2.2.1. Capabilities of Invasive vs Non-invasive BCIs

BCIs can be divided into three main categories:

- Invasive BCI
- Partially invasive BCI
- Non-invasive BCI

Invasive BCIs are a type of BCI where the electrodes are placed directly into grey matter in the brain by neurosurgery. This type of BCI gives the best quality signals since they make direct contact with the brain. But they are also affected by scar tissue build-up, as the brain considers the electrodes as a foreign object and tries to reject it. This can cause the signals obtained by invasive BCIs to degrade over time [28]. When compared with other types of BCI, invasive BCIs are not used frequently, since they involve complicated procedures such as neurosurgery. Invasive BCIs have higher capabilities than other types of BCIs as shown in a famous research by William Dobelle [29] where an implant was used to give visual signals to the brain from a camera in order to restore vision of a blind person. However this system later caused infections and seizures for the user, which clearly shows the high risk associated with invasive BCIs despite.

Non-invasive BCIs only require the access to the scalp of the user, and does not require any type of surgery. In order for the electrodes to make a proper contact with the scalp, an electrode gel should be applied. The electrode gel typically consists of an electrolyte. This ensures the electrical connection between the skin and electrode, especially when the user has hair. Signals captured by non-invasive BCIs are noisier than the signals acquired by invasive BCI and hence cannot be used for complex tasks such as restoring vision or communicating precise movement controls. In non-invasive BCIs, the electrical potentials generated by firing of neurons have to travel through white matter, skull tissue and skin before reaching the electrode. This greatly reduces the quality of signal lowering the capabilities of non-invasive BCIs.

2.2.2. Different Types of EEG Signals and Their Characteristics

Brain waves are categorized into different types according to their frequency ranges.

- Delta wave – (0.1 – 3 Hz)
- Theta wave – (4 – 7 Hz)
- Alpha wave – (7.5 – 12.5 Hz)
- Mu wave (7.5 – 12.5 Hz)
- SMR wave – (12.5 – 15.5 Hz)
- Beta wave – (16 – 31 Hz)
- Gamma wave – (32 – 100 Hz)

Following is a description of several important types of brain waves observed in EEG.

Alpha Waves

Alpha waves are brain waves in the frequency range of 7.5 – 12.5 Hz and they were one of the first EEG waves observed by the German Psychiatrist Hans Berger along with Beta waves. Alpha waves become more prominent when a subject is resting with eyes closed, and when the subject opens his eyes, alpha waves decrease and Beta waves increase. Alpha waves are commonly used in sleep studies and researchers have used Alpha waves to predict mistakes done by humans [34]. In this research the presence of Alpha waves in EEG signal is used as a verification that the electrodes are connected correctly and the signal acquisition is functioning correctly, prior to conducting experiments.

Beta Waves

Beta waves fall into the frequency range of 16 – 31 Hz. Unlike Alpha waves which are more prominent during relaxation, Beta waves are often associated with active, busy or anxious thinking and active concentration [35]. Beta activity is also associated with motor functions and Beta activity is increased when movement has to be resisted or voluntarily suppressed.

Sensorimotor Rhythm

Sensorimotor rhythms or SMR have a frequency of 12.5 – 15.5 Hz. Users can gain control over SMR activity through Neurofeedback training.

Deliberate modification of SMR amplitude can be used to control a BCI by motor imagery as in [11].

Mu Waves
Mu waves fall into the same frequency range as Alpha waves. But unlike the Alpha signals, the Mu waves are observed over the motor cortex of the brain, roughly in a band over the head from ear to ear. This type of wave can also be used in motor imagery BCIs [11].

2.3. Summary

Assistive technologies can be implemented in many forms. Manual remote controls are commonly used in our everyday life. They make it easier for healthy users to control appliances in the environment since they eliminate the need to reach an appliance to control it, although disabled users might experience difficulties when using them. Mechanical switches are effective for disabled individuals who still have control of movement in one part of the body, but can only provide controls that are too limited to be effectively used by a healthy user. Hand gesture based controls have the advantage of not requiring touching anything to control, but have the same drawbacks of the manual remotes for disabled individuals. When compared to the solutions above, BCIs offer a truly hands−free control experience for the users.

Invasive BCIs give higher quality signals when compared to non−invasive BCIs but are more complex to use since they require the electrodes to be surgically implanted into the brain. Non−invasive BCIs do not require any type of surgery, which is one reason they are used more frequently than invasive BCIs.

EEG signals are categorized into different types according to their frequencies and observed characteristics. Some of them can be voluntarily controlled by the users which make them useful for creating BCIs.

3. Design

When developing an environmental control solution using BCI, selecting the correct appliance by sensing user's intention is somewhat challenging. The type of EEG signal to be analysed should be chosen such that the system is able to select the correct appliance from a number of options with sufficient accuracy.

An evoked potential based signal such as P300 is better suited for this task than spontaneous rhythms such as sensorimotor rhythms (SMR) since spontaneous rhythm signals provide no way to stop and remain on a discrete option for accurate selection [11]. The SMR selection can continue to change even after passing over the required option. Choosing evoked potential based signals also have the added advantage of not requiring extensive user training.

The existing P300 based environment control systems use control masks (Fig. 3) to choose the appliance and to issue control commands [2, 8]. The user has to focus on the symbol he wishes to select from the control matrix and the system will detect which symbol he concentrated on by the presence of P300 response in the EEG signals.

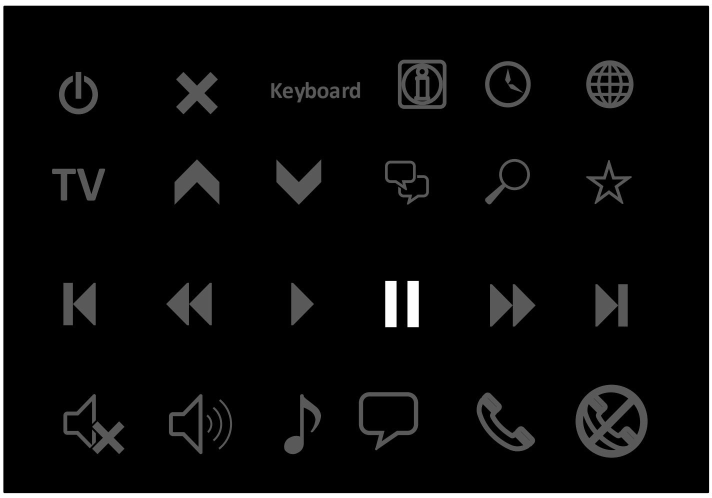


Figure 3.1 Typical example of a control mask used in a P300 BCI showing various control commands

This approach seems less intuitive especially for the healthy users, since it depends on selecting all the possible commands using a control mask, similar to using an entirely menu−based UI. This approach might also cause fatigue for the users since they need to constantly look at a display to control the system.

Therefore, this research intends to propose a different approach than the existing ones; that is, instead of using a control mask to select the users' choice, the appliance itself is highlighted by flashing the symbols in an indicator panel near each appliance (Fig. 4). This provides the user with a visual stimulus which will give the P300 response in EEG signals.

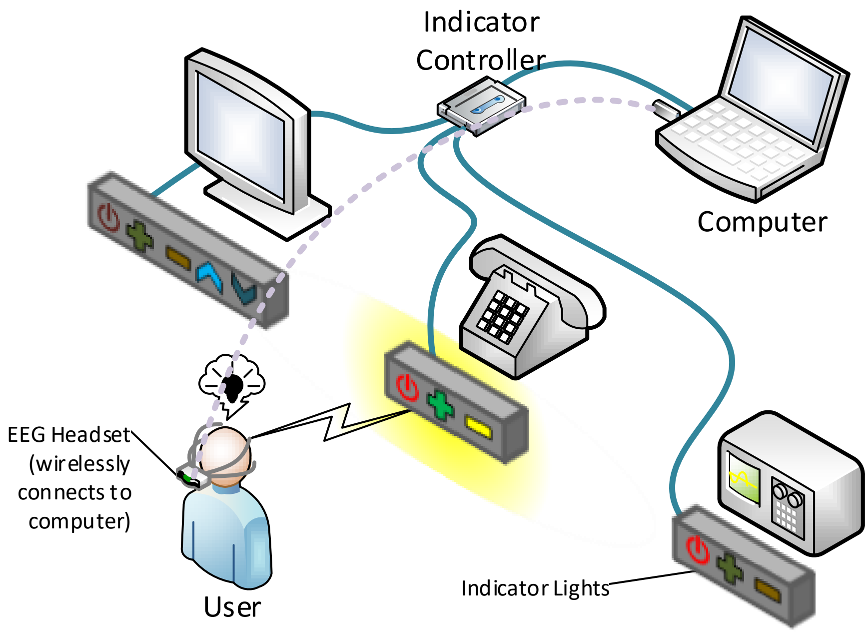


Figure 3.2 System overview: The user concentrates on the indicator of a certain appliance.

In order to record these responses effectively, signals will be captured from multiple positions in the scalp (Fig. 5). Since the visual P300 response is present more prominently around the visual cortex, multiple electrode positions (O1, O2, P7) around occipital and parietal regions will be used for capturing the response. One position in the frontal region (AF4) also used to detect the electrooculography (EOG) artefacts of eye movement and eye blinks. These signal channels will be monitored and will help in automated artefact removal where the EEG signals will automatically be discarded and will not be used for processing if artefacts are present in the EOG data at the same time period.

In the next step these signals are conditioned. The EEG signals are band–passed at 1–20Hz filtering the unwanted frequencies. Then the artefacts in the EEG signals are removed, which can be automated as described before. Finally the signals are processed and interpreted as control commands.

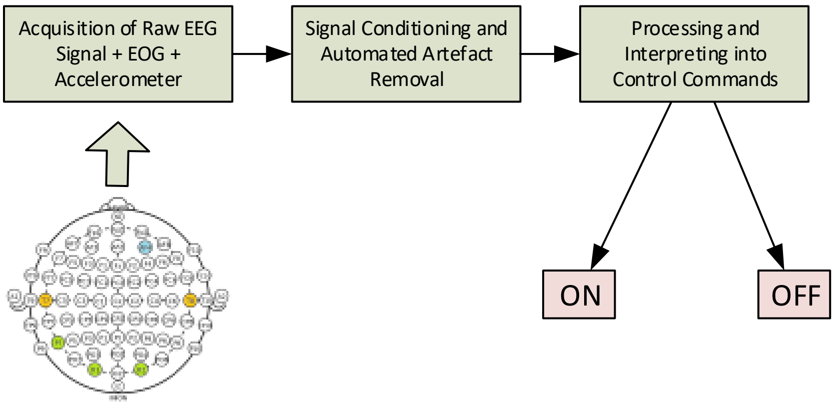


Figure 3.3 Process flow: EEG signal acquisition from multiple electrodes placed in the scalp, signal conditioning, processing and interpreting signals into control commands.

Selection of the command will be done by flashing an indicator panel that is placed next to the relevant appliance. These commands can be used either to select an appliance or to control an already selected appliance. The EEG signals will be then acquired and processed. The system decides that the user concentrated on a specific command by the presence of P300 response in the detected EEG signals acquired in a time–locked interval after the flash relevant to the command. Next the indicator panel will flash a set of different symbols associated with the set of control commands for the selected appliance. Here the user concentrates on the control command he wants to issue to the appliance. The correct command will be detected by the system in the same manner the appliance was selected.

3.1. Prototype Control Commands

A prototype system will be developed which simulates the controlling of four appliances. First the user has to select one of these four appliances. Once the selection has been made and confirmed the user can issue commands to control the selected appliance.

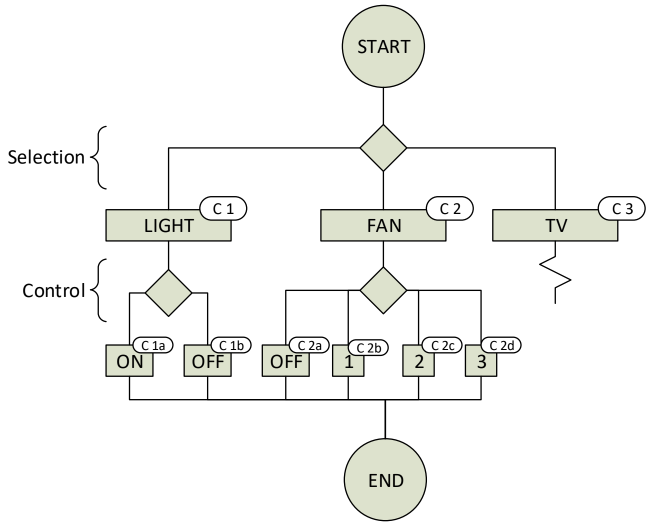


Figure 3.4 Typical flow of control of the system showing selection and control phases

The flash controller flashes the symbols associated with the commands in a random manner. The flash controller also sends a real-time stream of markers to the signal processing application (eg: C3, C1b, C2d, C3a...) which corresponds to the command that is being flashed at the moment.

The signal processing application receives four types of data streams: EOG stream from AF4 position, EEG signal channels (from positions O1, O2, P7), signal from a head-mounted accelerometer and marker stream. The EOG signal and accelerometer input are used for automatic artefact detection and removal. EOG signal will be used to detect eye movement and eye blinks while the accelerometer will detect the movements of neck and other head muscles. Both these movements introduce unwanted noise to the EEG signal, and removing these can improve the accuracy of the system [25].

After artefact removal the EEG signal is band-passed (typically at 1-20Hz) to remove the unwanted frequency components and retain only the components required for classification. Then the multiple channels of EEG signals are averaged, and time−locked intervals (epochs) that correspond to the command markers in the marker stream are extracted from this averaged EEG signal stream (Fig. 8). Note that some of these extracted epoch intervals might be overlapping, since the inter−flash interval is not necessarily greater than the extracted epoch interval.

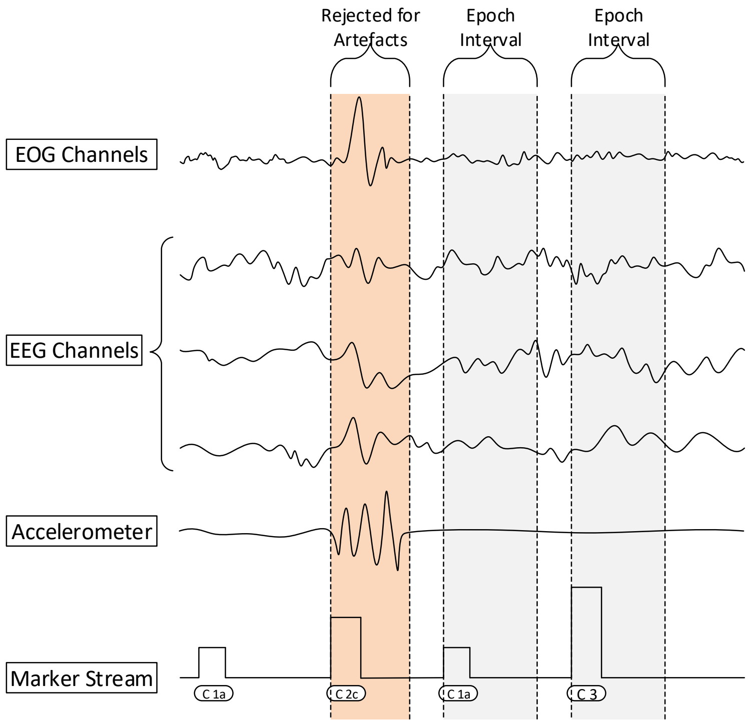


Figure 3.5 Four types of data streams received by the application

After extracting epochs for a repeated number of trials, the epochs are averaged for each command marker that was flashed. This highlights the P300 response from the otherwise−noisy EEG signals. After averaging epochs, the characteristic P300 response will become prominent only for the target command channel while the non−target command channels will have an irregular and low response (Fig. 9).

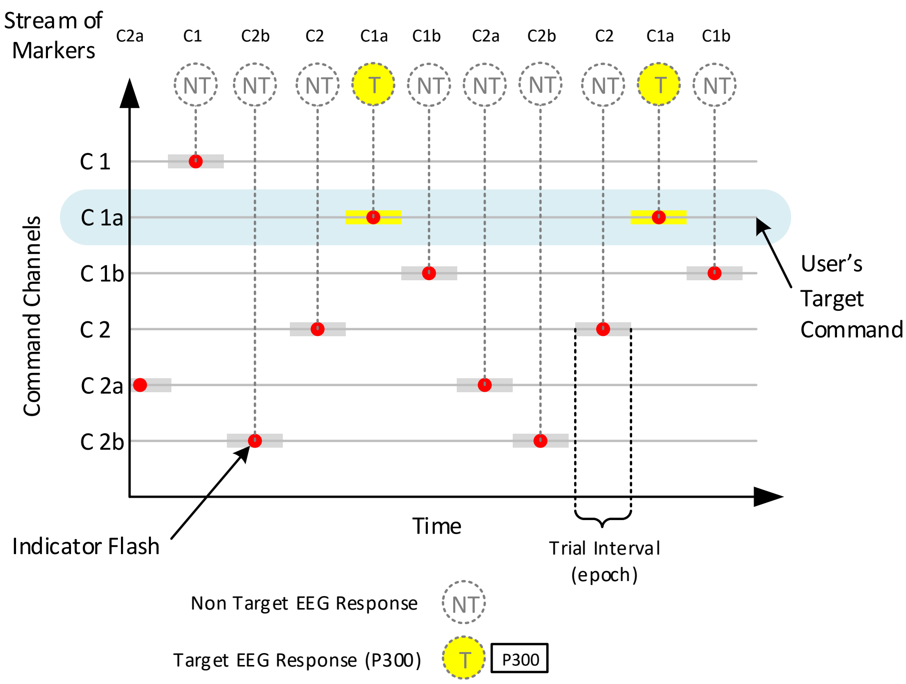


Figure 3.6 Time−locked intervals for flash markers in multiple control channels.

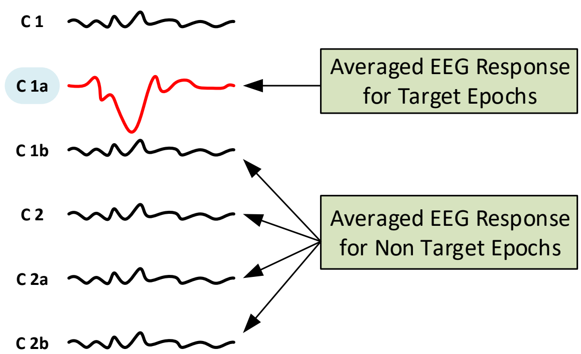


Figure 3.7 The target P300 response is only observed for the time−locked intervals to the marker of target command.

The averaged response can be classified into targets and non−targets by several methods such as Support Vector Machines (SVMs) or statistical analysis [25]. The classification method used as of now is comparing the areas under the curves of the different control commands at the time interval where P300 response can be observed for each user. After classification, the target command is selected as the command that the user intended to communicate to the system.

3.2. Possible Risks Involved and Health and Safety Precautions Taken when Conducting EEG experiments.

EEG recording procedures have been in use for over 30 years and are used routinely in hospitals to test brain function and diagnose neurological illnesses. There are no known major risks associated with EEG devices other than a mild discomfort for some people who have sensitive skin, when wearing the sensors. This is not permanent and does not cause serious consequences [37].

However, special care should be taken to prevent any harm from the electrodes that are attached to the scalp, since the scalp is a sensitive region and any electric shock to that region can be very harmful.

When using the gel electrodes of the EEG sensors for the experiments, low impedance contact areas are made with the scalp. If the devices were powered with a power source directly connected to the mains current (eg: 5V USB power), in an unlikely event of a power surge; for example caused by a thunderstorm, there is a possibility that hazardous currents may reach the scalp.

Therefore the signal acquisition devices being used connect to the computer wirelessly and are strictly battery powered. Before each experiment it is made sure that the devices are disconnected from external power and properly isolated from EMI sources which can interfere with the sensor readings.

The electrode pads used in the headband are single use pads and they are replaced for each new user. The inner surface of the band is covered with a lining of masking tape which is also replaced for each user. These procedures minimize the risk of disease transmission by contact between multiple users.

3.2.1. Factors Affecting Users' Comfort

The custom–made elastic headband that is used for the experiments is comfortable to wear and the feeling of the electrodes on the scalp becomes quite unnoticeable after some time when the tension of the band is correctly adjusted. The experiments require the users to minimize their blinking, jaw movements and muscle movements during the trials to minimize artefacts in the EEG signals. This can cause fatigue for the users when using the system for prolonged periods of time. In order to minimize fatigue we have designed the system to give periodic breaks to the user after a certain time of conducting the trials (once every 30 seconds) where the user is free to blink and move. Even if the user blinks or moves during the trials, the artefacts caused by those actions are removed during automatic artefact removal. The users should be acknowledged of this fact so that they can blink or move if they feel an urge to do so, avoiding discomfort. No other adverse effects have been observed so far with the current configuration of the system.

4. Implementation

As a preliminary experiment and a proof of concept, it was decided to conduct the typical P300 experiment first. Another goal of conducting a preliminary experiment like this is to test the capability of various data acquisition devices that are intended for use. The first device that was tested for this experiment was the BITalino biomedical sensor toolkit [23].

4.1. BITalino Biomedical Sensor Kit (cost ~ $200)

Figure 4.1 Componenets in the BITalino sensor kit

BITalino is a low cost biomedical sensor toolkit that includes multiple sensors such as Electromyography (EMG), Electrocardiography (ECG), Electrodermal Activity (EDA), Electroencephalography (EEG), Accelerometer and Light sensor. Although this is a low cost device, some researches show that BITalino is comparable to the research grade device [24] offered by the same manufacturer [24].
The microcontroller unit of this device consists of an Atmel ATMega328p [26] chipset which is commonly found in Arduino UNO, Arduino NANO and Arduino MINI. BITalino is capable of simultaneously capturing data

from 6 analog and two digital channels with up to 1000Hz frequency. The device converts analog sensor data to digital values using the Analog to Digital Converter(ADC) in the ATMega328p chip with a resolution of 10bits or 1024 levels (in 4 of the 6 input channels). The EEG sensor of the device we used had a gain of 40000 with a range of ±41.25μV.

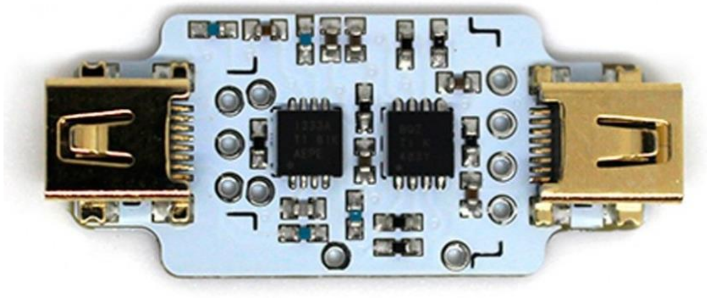


Figure 4.2 BITalino EEG module

Conducting the P300 Experiment with BITalino

In the first phase of the experiment an EEG cap made by Electro−Cap International (ECI) [27] was used as the electrode array. Usage of this cap gave the advantage of easily finding the required electrode positions since we were mainly interested in the readings from P3 and P4 electrode positions. However using this EEG cap required a significant preparation time (of about 15 min) and was uncomfortable to wear. We were not able to acquire a proper EEG signal during this experiment due to issues of connecting the ECI EEG cap to BITalino which were incompatible with each other by default.

In the second phase of the experiment we directly connected the sensor cable provided with the BITalino to the scalp with help of electrode pads and conductive electrode gel. We secured the electrodes on the head by using an elastic headband.

Figure 4.3 Connecting BITalino 3 electrode cable for EEG signal acquisition

The EEG data that was captured with this setup was observed for the presence of Alpha waves present in the EEG signals during relaxation with closed eyes and for the typical artifacts that are present in EEG signals. A section of EEG data that was observed is provided in the Results Chapter.

In order to conduct the P300 experiment, it was modeled in OpenViBE BCI software. The experiment modelled in OpenViBE displays the P300 speller visualization and generates a stream of markers for target flashes and non−target flashes. This stream of markers should be synchronized and combined with the signal stream obtained from the BITalino. In order to achieve this, a tool was developed in C++ using the BITalino C++ API and VRPN button server. The two streams were synchronized with a tolerance of 10−40ms with the help of a light sensor.

After conducting the P300 experiment a several times we analysed the collected data using EEGlab EEG signal processing toolkit for MATLAB. The data was bandpass filtered at 1−20Hz, visible artefacts were removed, events and epochs were extracted and response for each event type was averaged separately. The results obtained for these P300 experiments are shown in the results chapter.

The ADC in BITalino is only capable of converting the analog signal with a resolution of 10 bits, while other low-cost devices like EMOTIV EPOC and OpenBCI Ganglion are capable of conversion with up to 16 bits and 24bits of resolution respectively. The BITalino EEG module is capable of acquiring only one channel of EEG data, which limits its possibility to be used as an effective BCI device. And the P300 experiments conducted with EMOTIV EPOC appears to give a better response than BITalino [19]. Due to these reasons it was decided to repeat the experiment using the EMOTIV EPOC signal acquisition device.

4.2. EMOTIV EPOC+ 14 Channel Mobile EEG (cost ~ $800)

Figure 4.4 Emotiv EPOC+ EEG Headset

The Emotiv EPOC+ is a relatively low cost BCI device which is capable of acquiring EEG signals from 14 channels and has an ADC with a resolution of 14bits. It is capable of capturing data with a rate of 128 samples per second. Although it is less than the sampling rate of BITalino (1000 samples per second), it is sufficient for the detection of P300 potential. Since this device has multiple EEG channels, it can acquire EOG data as well, which can detect eye movements and can help in artefact removal.

4.2.1. Conducting the P300 Experiment with Emotiv EPOC and LED Indicator

The Emotiv EPOC device that is used for this work is a customized version of Emotiv EPOC where only the internal circuitry from the original device is used and the electrode contacts are broken out. Using headers and 3.5mm connector sockets it is possible to attach up to 3 electrode sensor cables which are typically used for acquiring ECG signals. This configuration has considerably decreased the electrical impedance of the contacts made with the scalp, which has also increased the prominence of the EEG responses captured with this customized device than the original configuration. This modification also gives the freedom to use a desired number of electrodes and place them anywhere on the scalp as needed.

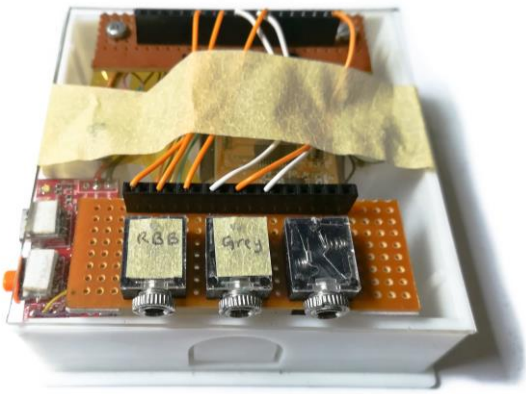


Figure 4.5 The customized Emotiv EPOC device

A custom made headband was attached with 5 electrodes for positions T7, T8, P7, O1, and O2 which can be worn around the head. The electrode used to detect eye blinks is directly pasted onto the skin above the right eye.

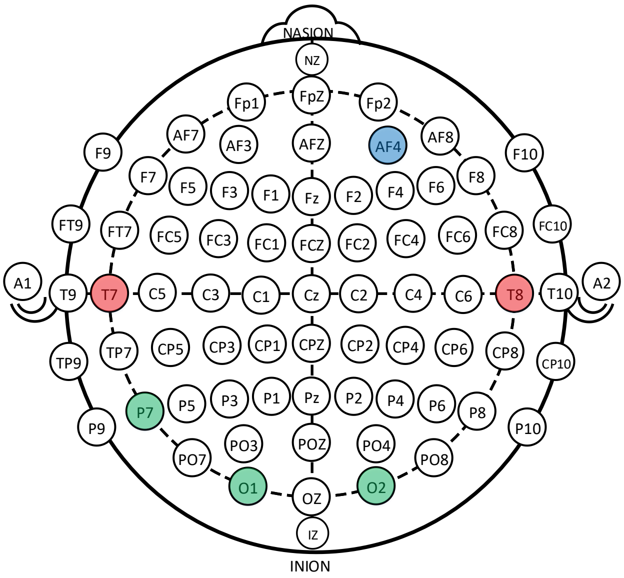


Figure 4.6 Electrode positions that are being used

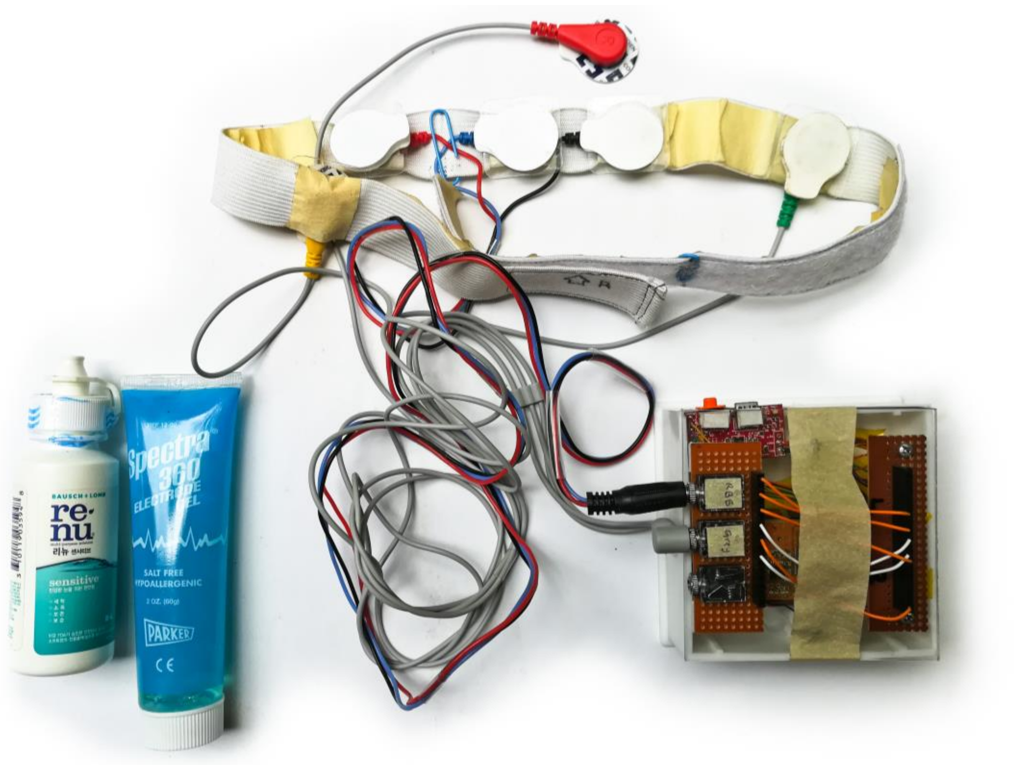


Figure 4.7 The custom-made EEG band (Version 1.0) connected to the device and the electrode gel and solution (left)

The headband requires less preparation time as compared to the ECI EEG cap we used previously. A small amount of conductive electrode gel should be applied to each of the single-use electrode pads before the band is worn in order to ensure a good electrical contact between the scalp and the electrode. The quality of electrical contact can be checked by using the Emotiv TestBench application. In order to verify that the EEG signals are being correctly captured, we check for the presence of Alpha waves in the signal feed using Emotiv TestBench before starting the experiments.

Since the P300 response was clearly observed in the first few datasets we acquired with the Emotiv, we proceeded to conduct the experiments by flashing LEDs instead of the OpenViBE P300 Speller visualization stimulator we used in the previous experiments.

A simple LED flashing circuit was created using an Arduino Pro Micro and an RGB (Red-Green-Blue) LED which can be flashed in any desired color using Pulse Width Modulation (PWM)

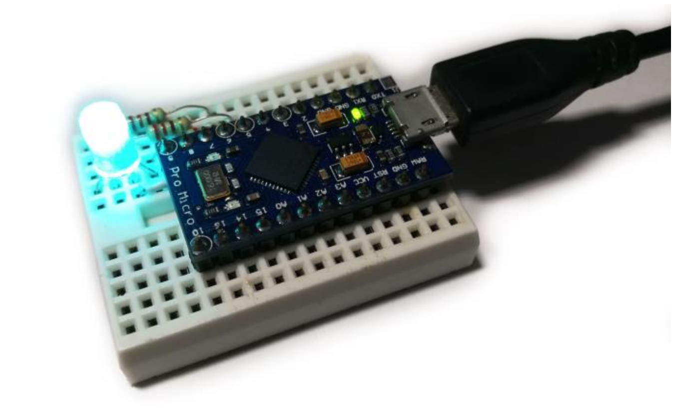


Figure 4.8 LED indicator which is connected to the computer via USB

During the experiments, the LED is flashed in two different colors: Green for targets Red for non-targets.

An application was developed in C++ to handle data collection and stimulus generation. This application consists of two threads which are run in parallel in real-time. One thread generates a stream of random markers and sends them to the Arduino through USB serial which in turn flashes the LED in different colors depending on the marker value. The

Arduino sends the received marker back to the program, which helps in synchronizing the communication.

The other thread collects the EEG data stream at 128 samples per second and combines this with the marker stream generated by the previously mentioned thread and writes them into a file. This program needs to run virtually in real−time with synchronization between two threads as well as the Arduino communication within millisecond tolerances since even a small out of sync of the EEG stream and marker stream might result in the P300 response not being visible in the averaged EEG signal. The source code for this application is provided in Appendix C.

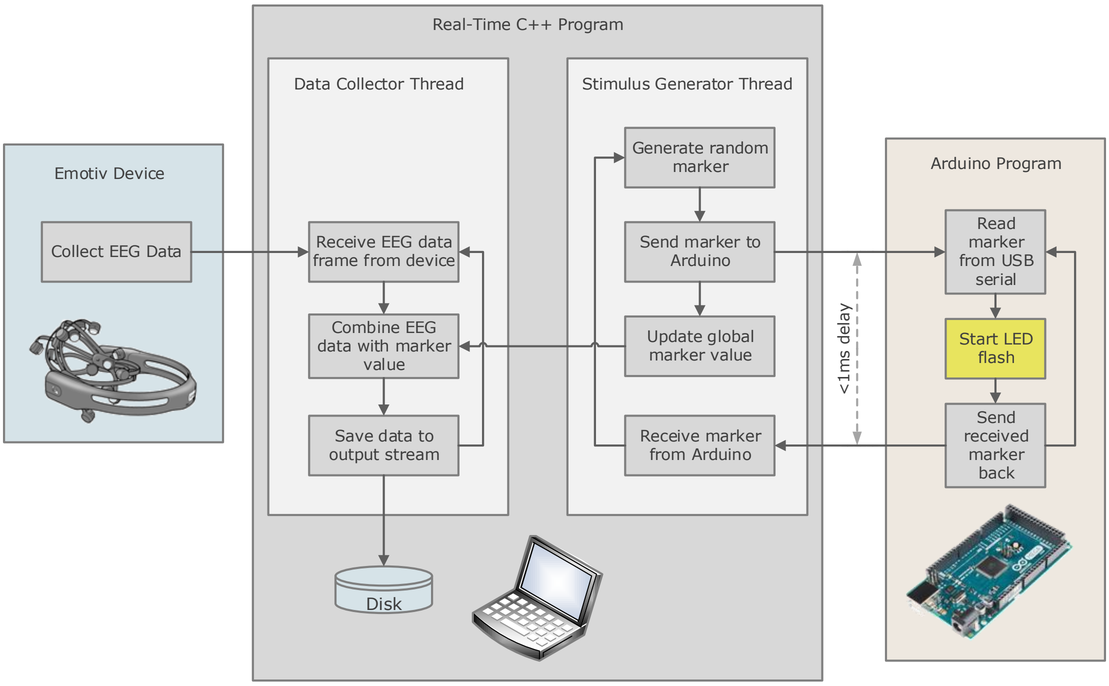


Figure 4.9 Overview of the C++ application – version 1

After confirming that the results were stable, the artefact removal and classification process was automated. The automatic artefact removal was achieved with a threshold function where any trials containing EEG and EOG signals out of the −40 to +40 microvolt range was rejected (Figure 20). The automatic classification was performed by comparing the sum of EEG potential in a time window where the P300 potential is expected to be present. The averaged epochs where the most negative sum was chosen as the target input.

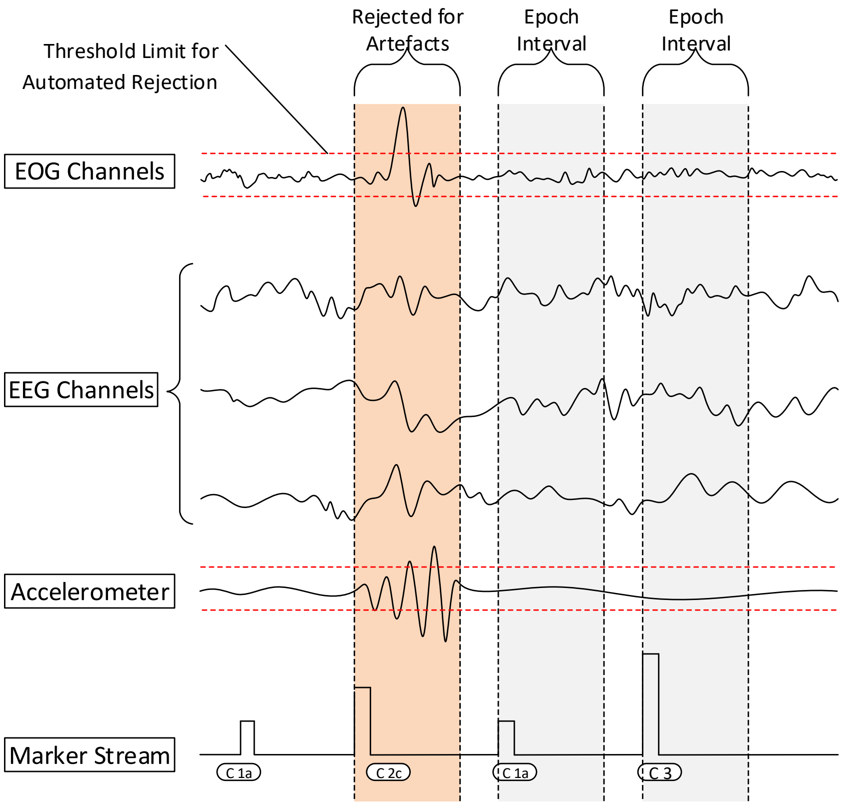


Figure 4.10 Four types of data streams received by the application

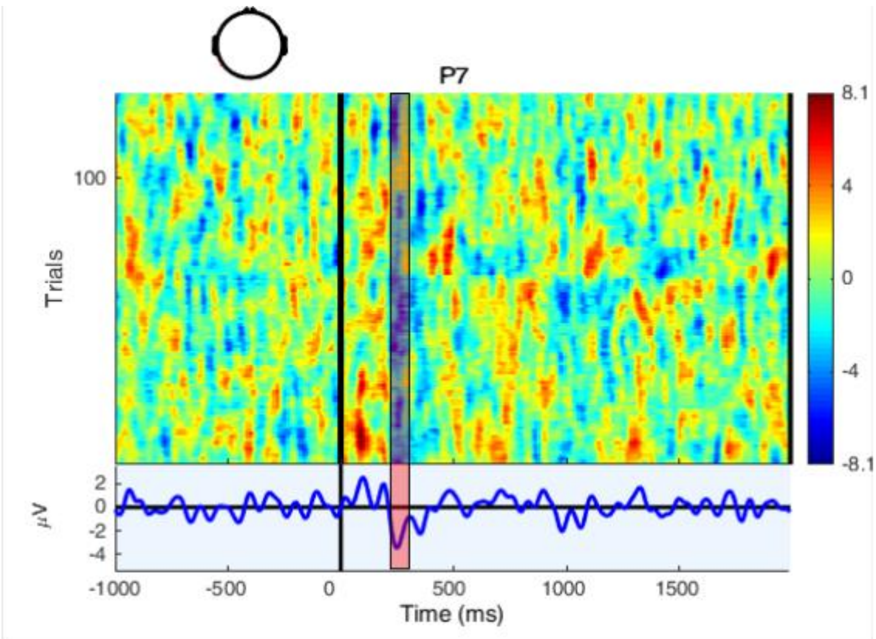


Figure 4.11 The interval classifier

4.2.2. Hardware Development of the Final Version of Complete System

After testing performance of the system which uses a single LED indicator with the automated classifier, a new indicator device which uses an Arduino MEGA was developed with 4 indicator LEDs, which were attached to flexible wire ribbons. This indicator utilizes 12 Pulse Width Modulation (PWM) pins of the Arduino MEGA which makes it possible to flash the four indicator LEDs in multiple colors.

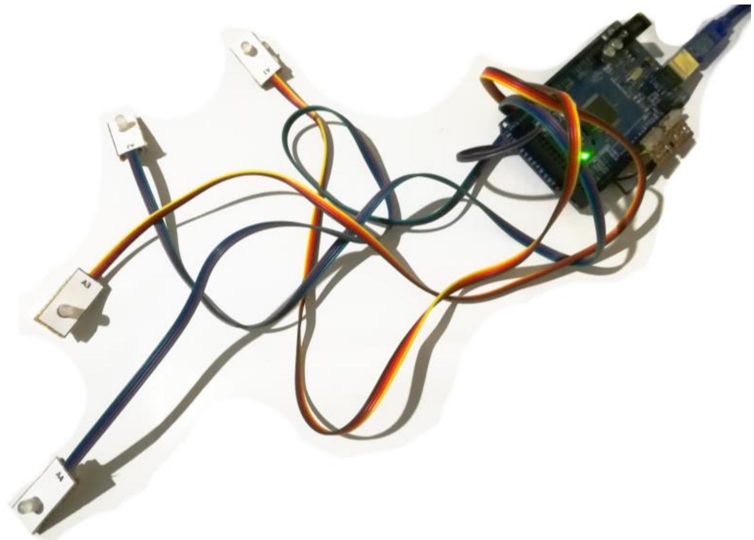


Figure 4.12 Indicator with 4 LEDs

In order to improve the visual appeal and user friendliness of the system, a new headband and a new enclosure was made for the system. The headband was designed in a way that minimizes disturbances to user's appearance. The new headband is black colored, in order to match the hair color of majority of intended users. Electrode adapters are fixed onto adjustable buckles fixed along the headband, which makes the electrodes adjustable to the correct locations on different users with various head sizes. The band is elastic and the size can be adjusted between 44cm and 59cm. The enclosure of the modified Emotiv EPOC was also upgraded to make it more appealing and user friendly.

Figure 4.13 Version 2.0 of the device with fully adjustable headband and upgraded device enclosure

A model environment was built with 2 appliances, a lamp with two control states (on, off) and a fan with 4 control states (off, speeds – 1, 2, 3). The fan works by a motor driver which is controlled by a PWM output of an Arduino UNO. The lamp is directly powered by a digital IO pin of the Arduino UNO. Each model appliance can be controlled with a single character sent to the RX pin of the Arduino UNO through USB serial.

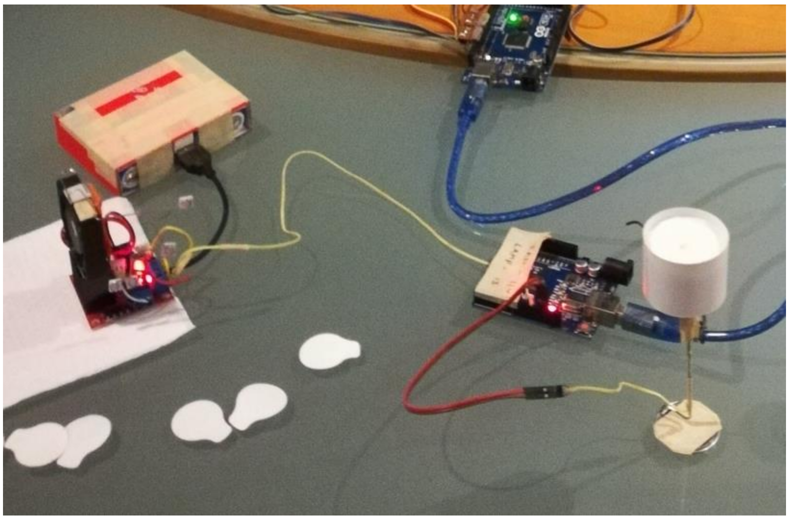


Figure 4.14 Model environment with fan (left) and lamp (right)

4.2.3. Development of a Model-based, Adaptive Classifier with Gaussian Temporal Filtering and Weighted Channel Averaging

The accuracy of the prototype automated classifier was found to be inadequate to correctly classify the control commands at selection times lower than 60 seconds. Therefore the need of a better classifier became prominent. A thorough analysis of the most prominent features were performed in order to extract the features that can help the most in discriminating the target P300 response from the unattended non-targets.

In the target P300 response EEG patterns, it was observed that the electrode locations at the front of the head detected a negative potential while the locations near the visual cortex of the brain detected positive potentials. Therefore taking the potential difference between positions at the front of the head enhanced the target P300 response by a significant amount. The signal response was further enhanced by a Gaussian filter positioned around a time delay where the P300 response was expected. This time delay differs from person to person and has to be calibrated for each user.

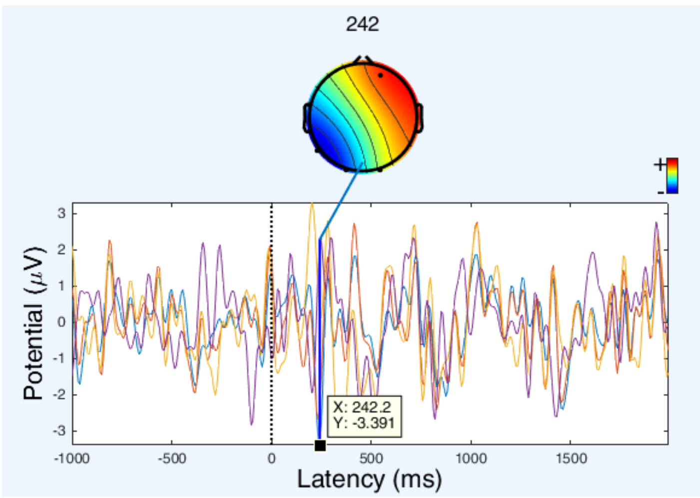


Figure 4.15 The potential difference between channel locations at the front of the head and behind head (top)

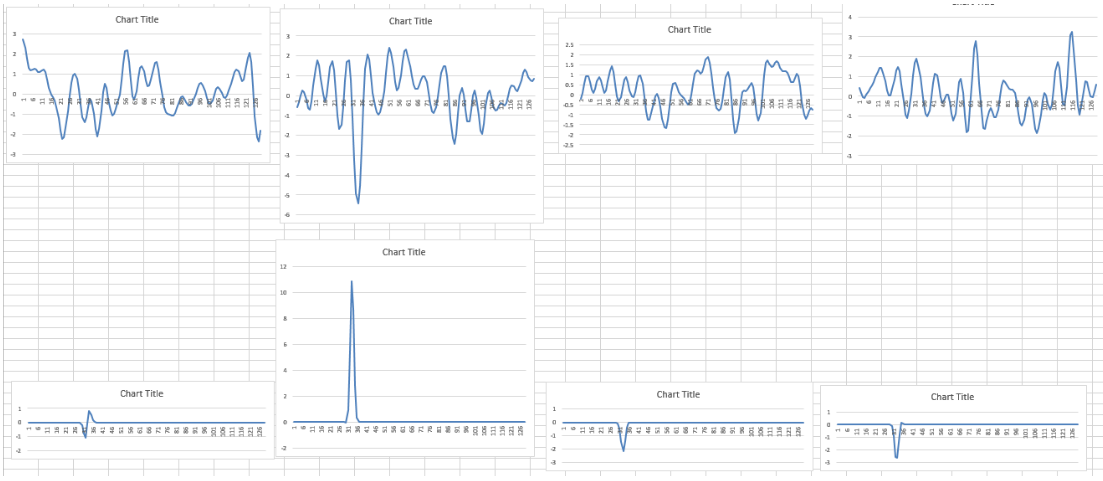


Figure 4.16 Enhancement of the target signal by the Gaussian filter and potential difference. Signal pattern of ordinary responses (top) and after applying new technique (bottom) – #2 is the target response

The implementation of the classifier in C++ is shown below

```
double gaussian(double x, double b){
       double a = 1.0, c = 10.0;

       return pow((a * 2.71828),
              (-(pow((x - b), 2)) / (2 * pow(c, 2))));
}

double weighting(double AF4, double P7, double O1, double O2){
       double w1 = 0.2, w2 = 1.0, w3 = 0.6, w4 = 0.6;

       AF4 = (AF4 > 0) ? AF4 : 0;
       P7 = (P7 < 0) ? P7 : 0;
       O1 = (O1 < 0) ? O1 : 0;
       O2 = (O2 < 0) ? O2 : 0;

       return (AF4*w1 - P7*w2 - O1*w3 - O2*w4);
}
```

Integration of the Entire System in main C++ Program

The system consists of the Emotiv data collection API, stimulus generator, Arduino code for the two Arduinos used. All these components were combined through the central C++ program.

5. Results and Evaluation

5.1. Evaluation Plan

This project was quantitatively evaluated under three different aspects: Classification accuracy, Selection speed and Information transfer rate. Users' feedback was gathered using feedback forms. These data were considered as qualitative data.

5.1.1. Classification Accuracy

Classification accuracy is the accuracy of correctly classifying a command the user intends to input to the system during a set of trials (an experiment). Achieving a high accuracy can be challenging, considering the limitations of low-cost equipment. The accuracy is calculated as follows:

$$Accuracy = \frac{Correctly\ Classified\ Commands}{Total\ Number\ of\ Commands} \times 100\%$$

The accuracy depends on several factors such as number of repeated trials in an experiment, number of commands or options we select from, inter–stimulus interval for the P300 experiment and the classification method. Accuracy will also be affected by factors such as user's level of concentration, user's physiological factors and external electrical noise.

5.1.2. Selection Speed

Selection speed is inversely proportional to selection time: which is the time taken to select one command from a set of trials. It is important to achieve a selection time as low as possible while maintaining a high accuracy to make the system more usable. This depends on several variables such as the number of control commands (command channels), number of repetitions per command channel, total number of trials and the inter–stimulus interval of the P300 experiment.

Since the P300 response cannot be identified with a single trial, the experiment should be repeated for a number of trials for a command channel. The total number of trials depends on number of command channels and the number of repetitions per channel.

$$Total\ Number\ of\ Trials = No.of\ Command\ Channels\ \times Repetitions\ per\ channel$$
$$Selection\ Time = Total\ No.of\ Trials\ \times Inter\text{-}Stimulus\ Interval$$

Selection time is the product of total number of trials and the inter−stimulus interval (ISI). ISI is the interval between two stimuli (flashes of the indicator). Lowering ISI reduces the selection time, but it also reduces the accuracy since it reduces the prominence of P300 response. This has been observed with ISIs below 600ms [28]. Therefore, in order to achieve a maximum speed for this system, an optimal trade−off between classification accuracy and selection speed has to be found.

5.1.3. Information Transfer Rate (ITR)

ITR is the amount of information transferred through the BCI in a unit time, which is usually measured in bits per minute. When classifying to $\boldsymbol{N}$ commands or discrete values, and classification time $\boldsymbol{t}$ the information transfer rate $\boldsymbol{B}$ can be obtained as follows.

$$B = \frac{1}{t}\log_2 N$$

But this is only valid when considering a perfect classification (classification with 100% accuracy). When considering an imperfect classification accuracy of $\boldsymbol{p}$ and classification time $\boldsymbol{t}$ the ITR can be written as follows [30].

$$B = \frac{1}{t}\left[\log_2 N + p\log_2 p + (1-p)\log_2\left(\frac{1-p}{N-1}\right)\right]$$

While speed and accuracy gives an idea about the usability of the system, ITR can represent the capability of the system to communicate thoughts.

5.1.4. User Feedback

The feedbacks of users were collected through feedback forms after each user completed the experiment session. Parameters that can affect the experiment outcomes such as fatigue, stress and the users' overall opinion about the system were evaluated through this. The questionnaire form is included in the Appendix.

5.2. Results of preliminary experiments conducted with BITalino

5.2.1. First successful acquisition of EEG signals

With the initial setup using the electrodes provided with the BITalino sensor kit, it was possible to successfully capture EEG data which could be noted by the presence of artefacts such as eye blinks, eye movement and jaw clenching which were expected to be present in EEG data.

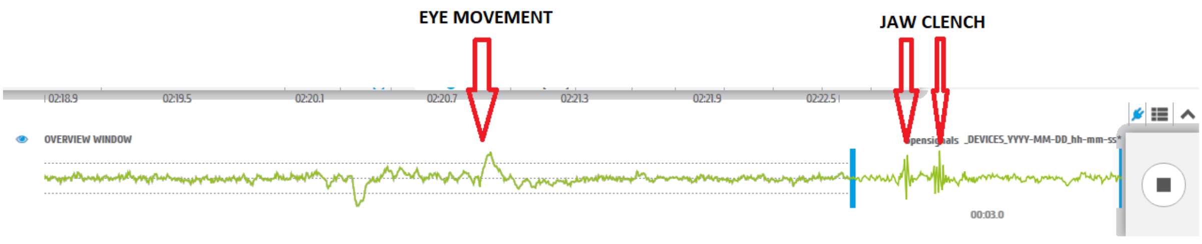


Figure 5.1 EEG data aquired with BITalino showing typical artefacts

5.2.2. P300 experiment with BITalino

After conducting the P300 experiment a several times we analysed the collected data using EEGlab EEG signal processing toolkit for MATLAB. The data was bandpass filtered at 1–20Hz, visible artefacts were removed and events and epochs were extracted. After averaging the epochs, a faint deflection of the EEG signal could be seen at ~450ms in the target data set of one of the experiments. And the deflection seemed to be consistent throughout the target epochs in that dataset. With this observation we can have a certain confidence that BITalino is capable of capturing the P300 response. However the results were unstable and the response was not very prominent.

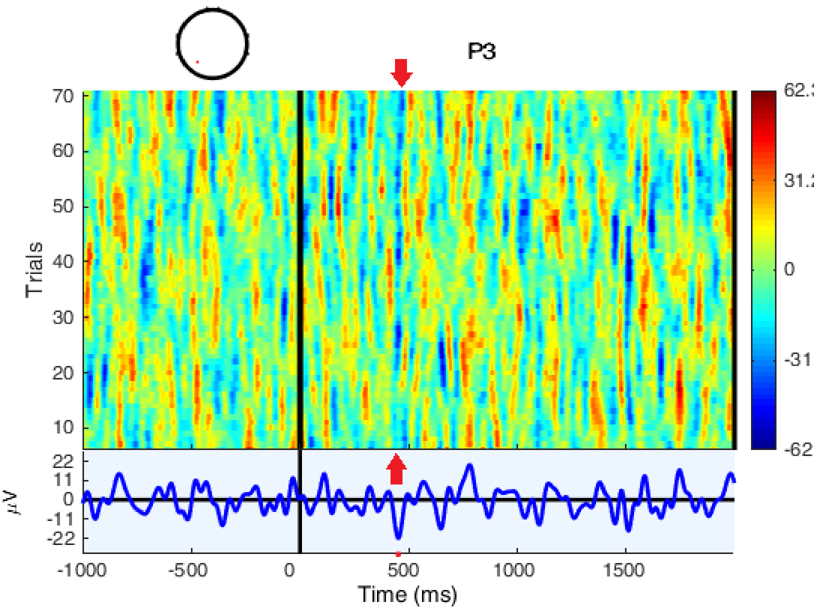


Figure 5.2 Averaged response for target epochs

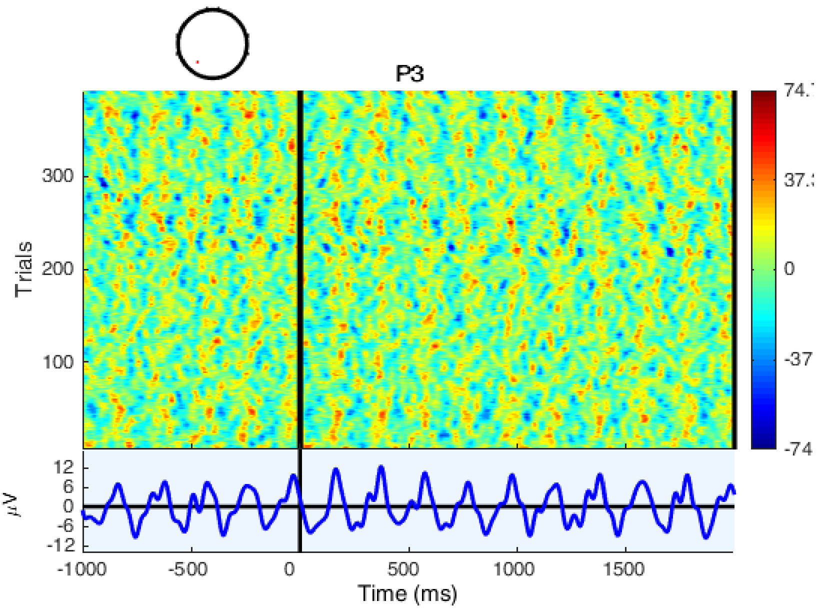


Figure 5.3 Averaged response for non−target epochs

5.3. P300 experiments conducted with Emotiv EPOC

The results obtained with Emotiv and LED stimulator were stable where the response could be easily observed in all of the conducted experiments. The P300 response can be observed in the averaged signals at around 250ms latency.

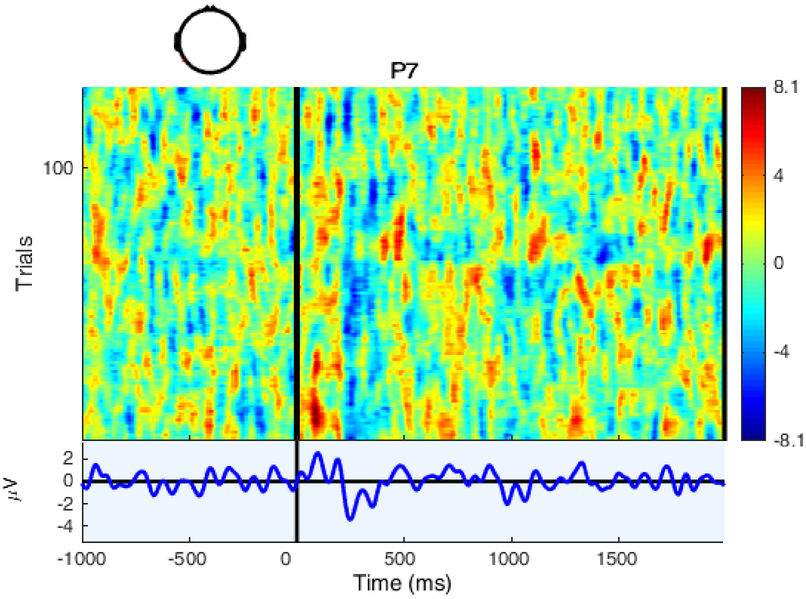


Figure 5.4 Averaged EEG signal of target epochs. Notice the vertical blue region and the negative peak at ~250ms.

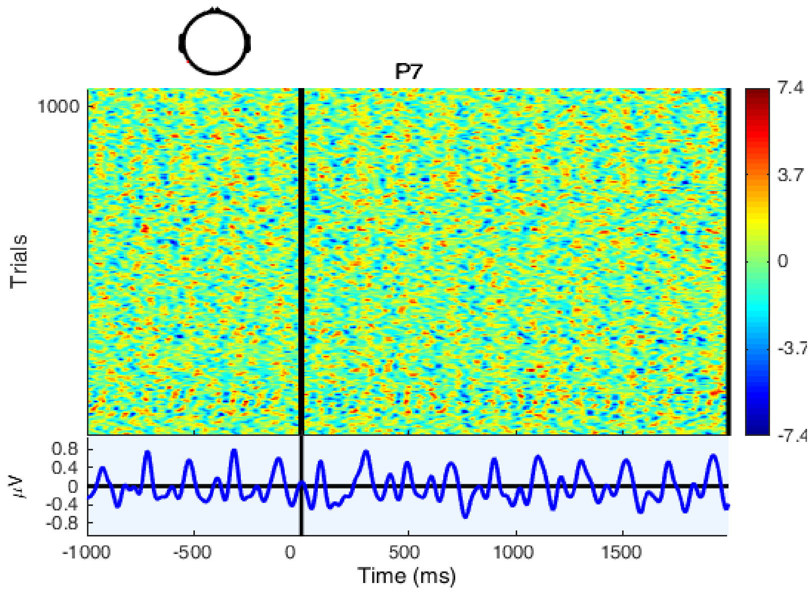


Figure 5.5 Averaged EEG signal of non−target epochs

It was also notable that the C++ application synchronized with the Arduino board with a precision of 230 microseconds.

5.3.1. Performance of prototype classifier

This method of classification is fairly accurate as of now (100% accuracy). However, better classifiers will be required in order to reduce the number of trials required to make a selection. Currently a single command experiment requires about 400 trials, which takes around 2 minutes to perform a single selection. The next step of the research will be reducing the number of trials while maintaining a high accuracy.

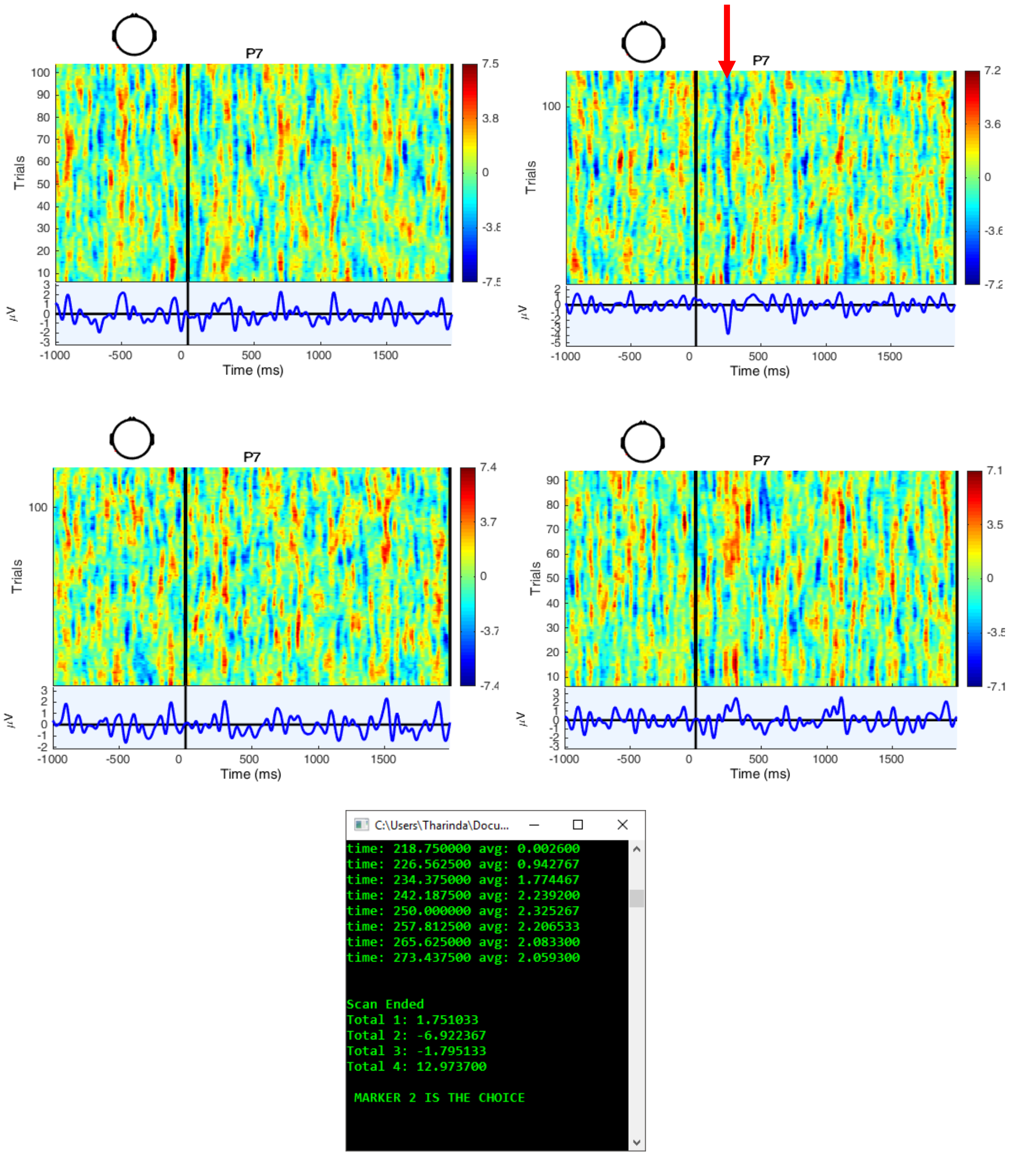


Figure 5.6 Responses for targets #1–NT (top left), #2–T (top right), #3–NT (bottom left), #4–NT (bottom right) and the output of the classifier (bottom)

5.4. System Testing with Test Users

After the system development was completed, it was tested with 10 users. A soundproofed TV studio was chosen as the environment for the experiments since the environmental conditions such as the lighting, background noise can be controlled precisely. The control tasks consisted of controlling the four speed fan mentioned in the Implementation chapter.

Note: All participants have provided written consent to participate in the study and the photographs appearing in this document are published with explicit approval of the individual participants.

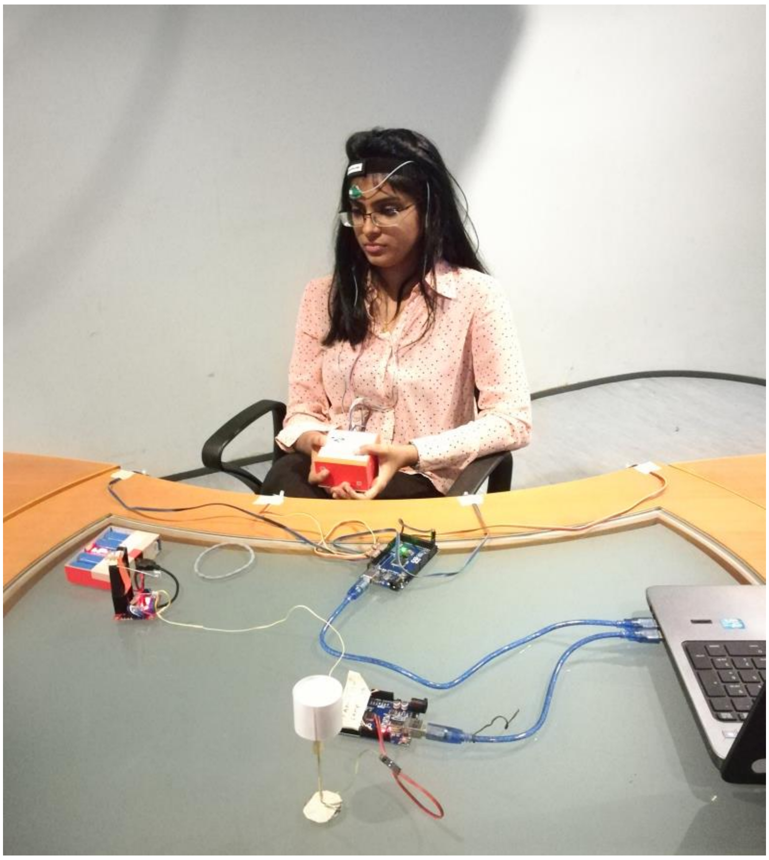


Figure 5.7 The complete experiment setup with a female user wearing the device. Distance from user’s eyes to indicators here is 50cm.

The experiments were conducted under different conditions such as varying distance from the indicators to the users' eyes and different lighting conditions.

Figure 5.8 A male user testing the system at a distance of 2.5m from the indicators

For the female users, the band has the advantage of being able to put under a layer of hair, which is impossible to do with EEG caps that cover the entire head of the user. Otherwise the results of EEG recordings could be affected by the thickness of hair of female users.

Figure 5.9 Detail of the way the band is put on for female users

5.4.1. Procedure for the User Testing Sessions

- Volunteered test user is given the consent form and given a thorough explanation of the nature of the experiments and any health and safety information regarding EEG procedures. He/she may or may not confirm the consent form at their will.
- The headband is fitted with fresh electrode pads, gel is applied and put on the users head.
- User is asked to concentrate on a predefined indicator in order to calibrate the parameters of the classifier.
- Once calibration is complete, user is given a sequence of targets to concentrate on, for the control tasks (off, 2, 3 ,2, 1, off, 3, off, …).
- The band is removed, head is cleaned up and user is given the feedback form to fill and return.

5.5. Experiment Results

The results of the experiments could be observable from the output of experiment history of the main C++ application. Output of results for one such user who tested the system at a distance of 2.5m from the indicators is shown below.

```
        Correct:-------10
        Wrong:---------0
        Accuracy:------100%
~~~~~~-----------------------~~HISTORY~~------------------------~~~~~~
#       C/W/A   Target  Result  Conf     Peak     Flash   ISI     Time
#1       1       4       4       72.14%   242.19   100     400     40
#2       1       1       1       69.07%   242.19   100     300     30
#3       1       4       4       31.97%   242.19   100     300     30
#4       1       1       1       50.24%   242.19   100     300     30
#5       1       3       3       48.07%   242.19   100     300     30
#6       1       4       4       54.24%   250      100     300     30
#7       1       1       1       64.52%   242.19   100     300     30
#8       1       4       4       67.62%   250      80      250     20
#9       1       1       1       83.16%   242.19   80      250     20
#10     1       2       2       59.75%   242.19   80      250     20
```

In each of the test sessions, the selection time was initially set to a higher value and then was decreased gradually until the accuracy was dropped.

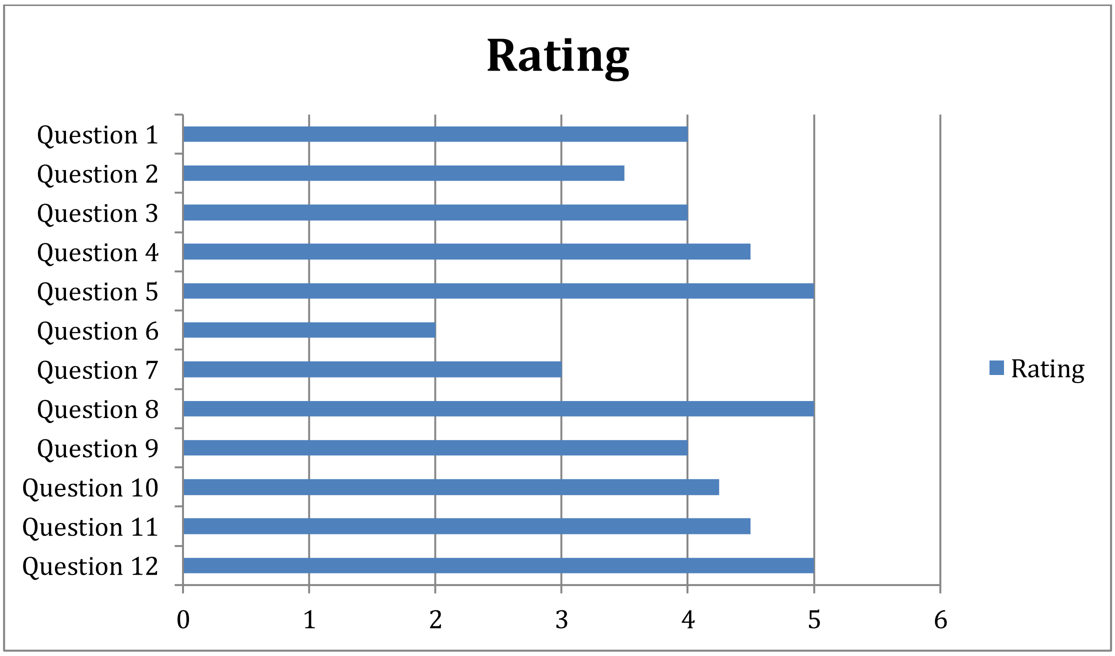
Overall accuracies for the 10 subjects tested are as follows

Subject #	Correct	Wrong	Total	Accuracy	Gender	Remarks
0	14	2	16	87.5%	M	
1	5	0	5	100%	M	
2	5	0	5	100%	M	
3	–	–	–	–	M	Experiment was stopped early
4	14	2	16	87.5%	F	
5	20	6	26	76.9%	F	
6	7	6	13	53.8%	F	User was sleep deprived
7	21	4	25	84%	M	Minimum time of 8s reached
8	10	0	10	100%	M	Tested at a distance of 2.5m
9	4	6	10	36.4%	F	Synchronization error in system

Table 5.1 Accuracies for test users

5.6. User Feedback Results (Averaged)

Refer to Appendix D for the questions



5.7. Evaluation of Results

The overall cumulative accuracy for each user falls when the selection time becomes lower than a certain threshold. It can be considered that 12 seconds is a stable minimum of selection time for the system. The ITR at that selection time is 10 bits per minute considering the system selects out of 4 possible values

Overall accuracy of the correctly functioning system:

$$Accuracy = \frac{96}{116} \times 100\% \; = 82.75\%$$

Minimum stable selection time can be considered as 12 seconds. This value is lower than the standard value generally accepted as the norm in related research publications.

6. Conclusions

6.1. Introduction

This chapter includes a review of the research aims and objectives, research problem, limitations of the current work and implications for further research.

6.2. Conclusions about research questions (aims/objectives)

With this research we have discovered a new method of controlling appliances in a physical environment using brain potentials. Other than discovering a new and improved method of BCI, we have also conducted experiments on the effect of using different color stimulators have on the subjects' brain responses. We have achieved very high accuracies for some test users and reached selection times lower than what is considered as the state-of-the-art selection times in BCI domain. The system is significantly cheaper than the alternatives that are capable of achieving similar performance and is less disturbing for the users. The device can be used with both male and female users alike.

6.3. Conclusions about research problem

The main research problem we are focused on in this research is finding a more intuitive means of environmental control which can be used by healthy and disabled people alike. The experiments conducted by us suggest that this system can be successfully used by healthy individuals, and also this system is quite user friendly, considering the short training time required.

According to expert opinion given to us after analyzing this solution by Professor Thashi Chang, senior Lecturer of Faculty of Medicine, University of Colombo, This system is also useable by disabled individuals having the following disease conditions

- Quadriplegia due to spinal code lesions
- Locked in syndrome
- Myopathies and muscular dystrophies
- Brainstem stroke

However, according to him, the system should be tested and proven in order to determine whether it is useable by users with

- Motor-Neuron Disease (ALS)

This study has contributed to the domain of Brain-Computer Interfaces by introducing a new method of providing the visual feedback for the user by eliminating the requirement of display screens for visual feedback. The previous method of environmental control lacked intuitiveness and also did not take advantage of the positioning of the appliances in the 3 dimensional space to issue control commands.

6.4. Limitations

The new method has the limitation of only being able to control the appliances that are in the line of sight of the user. The distance between the user and the appliance is also a limitation, since the visual separation of the indicators reduces when the user moves further away from the appliance.

6.5. Implications for further research

Although this proposed method was intended for the use of both disabled and healthy individuals alike, it was not evaluated on disabled users as part of this work. Therefore a further extension of this research will be to evaluate this system with disabled users.

This system may be extended further by creating a hardware framework that can interact with real home appliances.

Another future work will be to further reduce the time taken for a selection and to improve the system and develop this system to a product level. That is, to develop this entire system as an off-the-shelf system that can be easily customized and configured for various user requirements and implemented in different households.

References

[1] Guger, C., Schlogl, A., Neuper, C., Walterspacher, D., Strein, T., & Pfurtscheller, G. (2001). Rapid prototyping of an EEG-based brain-computer interface (BCI). *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *9*(1), 49–58.

[2] Guger, C., Holzner, C., & Groenegress, C. (2008). Control of a smart home with a brain-computer interface. *Brain–Computer Interface*, (i), 2–6.

[3] Dean J Krusienski, Eric W Sellers, François Cabestaing, Sabri Bayoudh, Dennis J Mcfarland, et al.. A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering, IOP Publishing*, 2006, 3 (4), pp.299-305.

[4] McFarland, D. J., & Wolpaw, J. R. (2011). Brain-computer interfaces for communication and control. *Communications of the ACM*, *54*(5), 60.

[5] Craig, A., Tran, Y., McIsaac, P., & Boord, P. (2005). The efficacy and benefits of environmental control systems for the severely disabled. *Medical Science Monitor : International Medical Journal of Experimental and Clinical Research*, *11*(1), RA32-A39.

[6] SU, M.-C., SU, S.-Y., & CHEN, G.-D. (2005). a Low-Cost Vision-Based Human-Computer Interface for People With Severe Disabilities. *Biomedical Engineering: Applications, Basis and Communications*, *17*(6), 284–292.

[7] Kilgore, K. L., Hoyen, H. A., Bryden, A. M., Hart, R. L., Keith, W., & Peckham, P. H. (2009). An Implanted Upper-Extremity Neuroprosthesis Using Myoelectric Control. *J Hand Surg Am*, *33*(4), 539–550.

[8] Cristian-Cezar P., Alexandra C., Alina P., Doru T. Evaluation of a P300-Based Interface for Smart Home Control. Luis M. Camarinha-Matos; Ehsan Shahamatnia; Gonçalo Nunes. *3rd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS)*, Feb 2012*, Springer, IFIP Advances in Information and Communication Technology*, AICT-372, pp.179-186, 2012,

[9] Leeb, R., Friedman, D., Müller-Putz, G. R., Scherer, R., Slater, M., & Pfurtscheller, G. (2007). Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic. *Computational Intelligence and Neuroscience*, *2007*.

[10] Tai, K., Blain, S., & Chau, T. (2008). A Review of Emerging Access Technologies for Individuals With Severe Motor Impairments. *Assistive Technology*, *20*(4), 204–221.

[11] Wolpaw, J. R., & McFarland, D. J. (2004). Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences*, *101*(51), 17849–17854.

[12](2018) Stephen Hawking, Official Website / The Computer [Online] Available: http://www.hawking.org.uk/the-computer.html
[13]Verstraete, M., Derboven, J., Gemmeke, J., Karsmakers, P., Broeck, B. Van Den, & Van Hamme, H. (2012). The design of voice controlled assistive technology for people with physical disabilities. *Proceedings of the First International Workshop on Language Technology in Pervasive Computing (LTPC)*, 2–5.
[14]Solanki, U. V., & Desai, N. H. (2011). Hand gesture based remote control for home appliances: Handmote. *Information and Communication Technologies (WICT), 2011 World Congress on*, 419–423.
[15]Reilly, R. B., & O'Malley, M. J. (1999). Adaptive noncontact gesture-based system for augmentative communication. *IEEE Transactions on Rehabilitation Engineering*, *7*(2), 174–182.
[16]Verstraete, M., Derboven, J., Gemmeke, J., Karsmakers, P., Broeck, B. Van Den, & Van Hamme, H. (2012). The design of voice controlled assistive technology for people with physical disabilities. *Proceedings of the First International Workshop on Language Technology in Pervasive Computing (LTPC)*, 2–5.
[17]Soderlund Staffan "Method For Playing Games Using Brain Waves" International Patent Application WO2004SE01778 20041129 21st July, 2005
[18]Chen, S. C., See, A. R., Chen, Y. J., Yeng, C. H., & Liang, C. K. (2013). The use of a brain computer interface remote control to navigate a recreational device. *Mathematical Problems in Engineering*, *2013*.
[19]Ekanayake, H. (2010). "P300 and Emotiv EPOC: Does Emotiv EPOC capture real EEG?" Retrieved from: http://neurofeedback.visaduma.info/emotivresearch.htm
[20]Aqeel-ur-Rehman, Arif, R., & Khursheed, H. (2014). Voice Controlled Home Automation System for the Elderly or Disabled People. *Journal Applied Environmental and Biological Sciences*, *4*(8), 55–64.
[21]Aloise, F., Schettini, F., Aricò, P., Leotta, F., Salinari, S., Mattia, D., Cincotti, F. (2011). P300-based brain-computer interface for environmental control: An asynchronous approach. *Journal of Neural Engineering*, *8*(2). https://doi.org/10.1088/1741-2560/8/2/025025
[22]Zhang, H., Guan, C., & Wang, C. (2008). Asynchronous P300-based brain - Computer interfaces: A computational approach with statistical models. *IEEE Transactions on Biomedical Engineering*, *55*(6), 1754–1763.
[23]BITalino Product Homepage http://bitalino.com/en/ [accessed: 20/06/2018]
[24]biosignalsplux Product Homepage http://biosignalsplux.com/en/biomedical-research
[25]Batista, D., Silva, H., & Fred, A. (2017). Experimental characterization and analysis of the BITalino platforms against a reference device. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2418–2421. https://doi.org/10.1109/EMBC.2017.8037344

[26] Atmel ATMega328p Datasheet http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Summary.pdf
[27]Electro-Cap International (ECI) Product Page http://electro-cap.com/index.cfm/caps/
[28]Polikov, V. S., Tresco, P. A., & Reichert, W. M. (2005). Response of brain tissue to chronically implanted neural electrodes. *Journal of Neuroscience Methods*, *148*(1), 1–18. https://doi.org/10.1016/j.jneumeth.2005.08.015
[29]Dobelle, W. H., Quest, D. O., Antunes, J. L., Roberts, T. S., & Girvin, J. P. (1979). Artificial vision for the blind by electrical stimulation of the visual cortex. *Neurosurgery*, *5*(4), 521–527. https://doi.org/10.1227/00006123-197910000-00022
[30]Kaper, M, P300 Based Brain Computer Interfacing 2006.
[31]Homepage of OpenBCI [Accessible online] http://openbci.com/
[32]Product page of Emotiv EPOC [Online] Available: https://www.emotiv.com/epoc/
[33]Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. Clinical Neurophysiology, 118(10), 2128–2148.
[34](2009) Brain Wave Patterns Can Predict Blunders. Webpage [Online] Available: https://www.ucdavis.edu/news/brain-wave-patterns-can-predict-blunders-new-study-finds
[35]Baumeister, J., Barthel, T., Geiss, K. R., & Weiss, M. (2008). Influence of phosphatidylserine on cognitive performance and cortical activity after induced stress. Nutritional Neuroscience, 11(3), 103–110.
[36]g.tec g.USBamp Product Page [Accessible online] http://www.gtec.at/Products/Hardware-and-Accessories/g.USBamp-RESEARCH-Specs-Features
[37]Emotiv: EEG Basic Participant Information and Safety [Accessible online] https://emotiv.zendesk.com/hc/en-us/articles/204701495-EEG-Basic-Participant-Information-and-Safety
[38]Jilani, A., Rabbani, S., Sheikh, I., Ahmed, O., Haider, A., & Asghar, F. (2015). Domestic Implementation using X-10 Protocol Controlled by PLC, *6*(2), 111–118.
[39]Farwell, L. A., & Donchin, E. (2014). Talking off the top of your head : A mental prosthesis utilizing event- related brain potentials Talking off the top of your head : toward a mental prosthesis utilizing event-related brain potentials, (May).

Appendix A: Publications

Tharinda Karunarathne, Hiran Ekanayake, Controlling Home Appliances through Thought Commands, Proceedings of International Conference on Advances in ICT for Emerging Regions, Colombo, LK
– September 2018

Controlling Home Appliances through Thought Commands

Karunarathne, K.A.T.A. #1, Ekanayake, H.E.M.H.B. #2

#University of Colombo School of Computing,
UCSC Building Complex,
35, Reid Avenue, Colombo 7
Sri Lanka

[1]katharinda@gmail.com

[2]hbe@ucsc.cmb.ac.lk

Abstract— **Brain−Computer Interfaces (BCI) is a field which has shown rapid advancement over the past few decades. With the availability of low−cost electroencephalography (EEG) signal acquisition devices, it is becoming more feasible to develop hands−free environmental control systems which can be used for everyday use. These systems may be used as an assistive technology for disabled individuals or as an alternative form of control for healthy individuals. Hands−free environmental control systems often lack user−friendliness and intuitiveness due to the difficulty of mapping the users' intentions, to control commands that can be used to control the appliances in a 3D environment. Therefore, this paper reports on an ongoing research to develop a hands−free environmental control system based on P300 responses of EEG signals of a user. The proposed solution does not use a screen for visual feedback, therefore, it improves the intuitiveness and user friendliness than existing solutions.**

Keywords− **Brain−Computer Interface, Assistive Technology, Electroencephalography, Environmental Control System**

I. Introduction

Brain−Computer Interfaces (BCI) can facilitate communication between the brain and an external device. A BCI can record electrical activity signals from the brain and classify them into different states which can then be interpreted for communication. Since BCI does not require muscle movements for communication, it can ideally provide an effective means of communication even for entirely paralysed people [10]. BCI research began in the 1970s and has shown a rapid advancement over the last few decades.

BCI research often aims for augmenting or repairing human cognitive or sensory−motor functions. BCI devices have been used for applications such as moving a cursor on a computer screen [11], controlling home appliances [2], wheelchair control [9], recreational use [17, 18], and for spelling purposes [3]. When considering the existing applications of BCIs for environmental control, there is a noticeable lack of applications where the user's orientation in the 3D environment is taken into consideration. Several similar works [2, 8] have developed environmental control systems that use a display screen for visual feedback. In this research we attempt to eliminate the need of a screen and make use of the user's orientation in the 3D environment to create a more engaging and intuitive experience for the user. With this research we intend to answer two main questions:

1. How to select an appliance that needs to be controlled in a 3D space?
2. How to control various types of appliances with minimal physical movement?

II. Motivation

People living with severe conditions of neurological disorders like Amyotrophic lateral sclerosis (ALS), brainstem stroke, brain or spinal cord injury, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases may lose all voluntary muscle control and may be completely 'locked−in' to their bodies, unable to communicate in any way. These conditions may significantly affect the person's quality of life due to loss of their ability to control the devices in their immediate environment [4, 5].

There are three options to restore function for these disabled individuals. First is to increase the capabilities of remaining functioning nerve pathways [6]. Second is to restore function by detouring around breaks in the neural pathways that control muscles [7]. The final option to restore function is provide brain with an alternate non−muscular communication channel in the form of a BCI to convey messages to the external world. This research attempts to help these individuals by minimizing the physical activity required to control the appliances in the environment.

Another aspect of this solution is providing healthy users with an alternate channel of communication that requires minimal muscle movement. Even healthy individuals might face situations where they cannot use their hands to control appliances (eg: while cooking, working with contaminated materials or chemicals, surgeons performing surgery). Currently, the majority of appliances we use every day are controlled manually or using a remote control. Recently, several large technology corporations have developed mass−produced consumer−grade home automation and environmental control systems that allow hands−free control of the home appliances.

Since recently, low−cost, yet sophisticated BCI equipment like OpenBCI [23] and Emotiv EPOC [24] have been developed. These devices are capable of capturing signals with sufficient quality that make them useful for BCI applications while also being significantly cheaper than medical or research grade BCI devices. These recent trends in the tech industry and capability of low cost devices have motivated us to develop an environmental control system which can be used for everyday use by healthy individuals as well.

As mentioned before, similar systems that currently exist have the limitation of having to depend on a display screen for visual feedback. And none of the existing systems utilize the positioning of the appliance in the 3D environment to select and issue complex control commands to various types of appliances, which makes a clear research gap.

III. Background and Related Work

Devices used in BCI research can be classified into three types: invasive, partially invasive, and non−invasive.

- Invasive BCI – Requires the devices to be implanted into the brain. These produce the highest quality signals of BCI devices.
- Partially Invasive BCI – the devices are implanted inside the skull, but outside the brain.
- Non−invasive BCI – the devices are connected outside of the skull and onto the scalp. These devices produce lower quality signals than Invasive BCI, but are less complex to apply. Majority of the BCI research focus on this category.

The following section is a literature review of environmental control systems that attempt to minimize the effort needed to control. Here the related works are categorized according to type of control. Each type has its advantages and drawbacks with respect to healthy and disabled individuals.

A. Manual Remote Control

Remote controls are an effective means of environmental control for healthy users which eliminate the need to physically walk to or reach the appliance. Majority of the users are already familiar with remote controls and they are highly accurate.

However, this solution may not be effective for disabled individuals since the controller often needs to be pointed at the appliance and some users have difficulty when operating the buttons on the controller [13].

B. Mechanical Switches

This type of solution is a switch that opens or closes which is controlled by explicit physical movement. The switch may respond to a specific mechanical stimulus, including changes in displacement, tilt, air pressure (e.g., sip and puff), or force [10]. This type of interface can be operated by people with control of only one part of body. The switch should be used in a way that utilizes the existing neural pathways for disabled individuals. A similar interface was used by the famous cosmologist Stephen Hawking to control his computer which used only the movement of his cheek [12]. This solution is cheap and has a high accuracy since it uses explicit physical movement.

A drawback of this solution is fatigue caused by repeated movement of the same part of the body [10]. These switches have limited control channels which make it ineffective for the use of healthy users since far better alternatives exist.

C. Hand Gesture based Control

Hand gestures provide an intuitive way of interfacing with the world. And can form complex methods of communication such as sign language used by people with hearing and speech disabilities. This technique can be used to control home appliances such as TV by using several commands represented by signs and movements of hands. These movements can be picked up by an Infrared camera and processed and interpreted as control commands [14].

Although hand−gestures can be an effective form of environmental control for healthy individuals, it requires the users to be able to move their hands, which makes it an ineffective solution for disabled individuals who cannot move their hands. When this method was used by disabled individuals who still possessed limited movement capabilities, they could only use it for short durations since they experienced fatigue [15].

D. Voice Control

Voice controls are a popular method of environmental control that is already being used for many applications. There are several commercial home automation systems offered by major tech companies like Amazon Echo, Apple HomePod and Google Home. Voice controls require no physical movement and it eliminates the need to reach or walk to the appliances to control them.

Even though voice controls have these benefits they are not widely used for assistive devices for several reasons. The user requirements for people with disabilities often have high levels of variation which creates a high cost for individual adaptation and development. Users who might benefit from voice commands often also have speech difficulties such that speech recognizers are unusable for them [16]. And some research results show that the accuracies can be significantly affected by noise in the environment [20].

E. Gaze Based and Eye−Tracking Control

Gaze−based communication can map eye movement to a cursor position on the screen. The dominant technologies in commercially available eye trackers are video−oculography (VOG) and electrooculography (EOG). VOG−based approaches typically use an IR light source and a camera mounted on a computer display. Gaze direction is calculated by the offset between the corneal reflection and pupil centre [10]. EOG−based systems place electrodes around eyes to measure shifts in potential difference between cornea and retina that occur when user changes gaze direction [8].

Although eye tracking controls have speeds comparable to a hand−mouse, productivity in computer tasks is lower in practice. Since this input method uses the same channel for control and observation, there is no intuitive means of differentiating between an input command and a user activity. Gaze controlled devices can also have drawbacks like calibration drift, user fatigue and insufficient range of motion of the eye [10].

F. Electroencephalography (EEG) and BCI Solutions

EEG based devices can provide a non−muscular form of control from the electrical activity measured on scalp. Since this does not require significant muscle movement, it is usable by tetraplegic disabled individuals as well [9]. Present day BCIs can be divided into two main categories based on the type of signals extracted: consciously modulated spontaneous rhythms or evoked potentials.

The first category of BCI uses potentials that can be intentionally modulated by the user with some training. A research by Wolpaw et al [11] that used sensorimotor rhythms (SMRs) has shown that individuals with severe motor disabilities are able to control a cursor on a screen with two−dimensional control signal using non−invasive BCI with accuracies up to 92% and relatively short response times [11]. However, this type of BCI requires some training to improve the reliability.

The second category relies on responses to external stimuli such as visual stimulus like flashing of an indicator. Steady−state visual evoked potential (SSVEP) is an example of this.

An SSVEP based BCI research conducted by Chen et al [18] have developed a recreational device that controls a toy fish using brain signals. The signals with high energy levels at different frequencies are produced when the user looks at control commands displayed on a screen that are flashing at the corresponding frequency. An average classification accuracy of 89.51% was obtained in their research.

Another type of signal that belongs to the evoked potential category is the P300 response. P300 potentials are positive deflections in EEG signal which were elicited approximately 300ms after encountering an intended stimulus among a group of irrelevant stimuli. This response can be used to create a BCI since it can identify which stimulus the user was concentrating on by analysing the EEG signals.

Since the P300 potentials have very low voltages (2−5μV) and are hidden within the EEG noise, they are not directly visible in an EEG recording [19]. In order to view these responses, the raw EEG signal should be band−passed (at around 1−20Hz), intervals of signal in multiple trials (called epochs, around −1000 to 2000ms relative to the stimulus) that are time−locked to the stimulus should be extracted, and then the epochs should be averaged (Fig. 1 and 2). This process averages out the noise voltages while accumulating the psychological properties of the stimulus, improving the Signal to Noise Ratio (SNR) of the response [25].

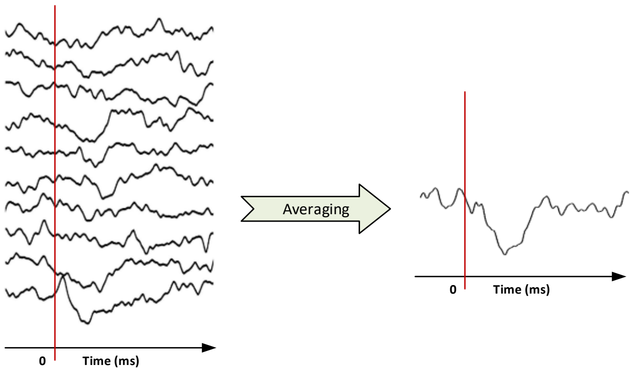


Fig. 1 The EEG signal intervals from multiple trials (left) time−locked to the event (red line) and after averaging them (right) which shows a deflection common to all the trials while cancelling out the noise.

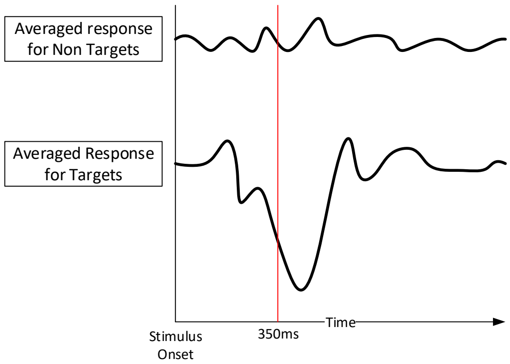


Fig. 2 Averaged EEG signal of multiple trials for non−target responses and target responses

P300 response can be obtained by flashing columns and rows of letters for spelling purposes [3]. A P300 based solution for smart home control developed by Holzner et al [2] has shown the suitability of the P300 response for selection of control commands. Their research uses control masks for the selection of areas inside the home and to control various appliances such as a phone and a TV. Accuracies up to 100% have been achieved in their research for some control masks.

Another research by Cristian et al [8] which used a hybrid BCI approach that combined EEG and EOG signals has studied how the reliability of an EEG system can be improved by combining it with another control channel. In their research the user had the ability to cancel the selection made by the EEG BCI by changing the direction of his gaze.

Several recently conducted researches have explored the suitability of Asynchronous BCI (A−BCI) for environmental control [21, 22]. Synchronous BCIs assume that the user is continuously controlling the system. They cannot detect whether the user is paying attention to the system or not and therefore they make incorrect selections when the user stops paying attention to the system. A synchronous P300 BCI makes a selection after a fixed number of trials regardless of whether the user was paying attention to the system or not.

In contrast to this, asynchronous BCIs make selections only when the response in the signals passes a threshold level [21]. Therefore, they can determine whether the user is focusing on the system or not and switch between control and no−control states accordingly. This type of a system can be more practical than synchronous systems since the user can focus on other tasks without the need of notifying the system. However, A−BCIs require a specialized algorithm or a support vector machine (SVM) to analyse and discriminate between the target P300 epochs and non−target epochs with respect to the threshold level [22].

IV. Research Methodology and Design

When developing an environmental control solution using BCI, selecting the correct appliance by sensing user's intention is somewhat challenging. The type of EEG signal should be chosen such that the system is able to select the correct appliance from a number of options with sufficient accuracy.

An evoked potential based signal such as P300 is better suited for this task than spontaneous rhythms such as sensorimotor rhythms (SMR) since spontaneous rhythm signals provide no way to stop and remain on a discrete option for accurate selection [11]. The SMR selection can continue to change even after passing over the required option. Choosing evoked potential based signals also have the added advantage of not requiring extensive user training.

The existing P300 based environment control systems use control masks (Fig. 3) to choose the appliance and to issue control commands [2, 8]. The user has to focus on the symbol he wishes to select from the control matrix and the system will detect which symbol he concentrated on by the P300 response present in the EEG signals.

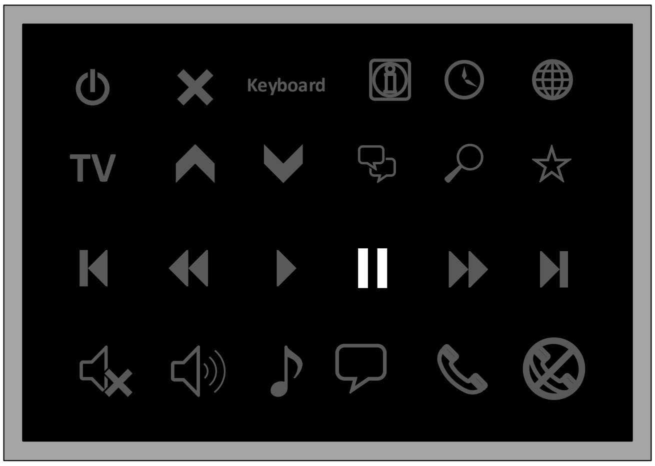


Fig. 3 Typical example of a control mask used in a P300 BCI showing various control commands

This approach seems less intuitive especially for the healthy users, since it depends on selecting all the possible commands using a control mask, similar to using an entirely menu−based UI. This approach might also cause fatigue for the users since they need to constantly look at a display to control the system.

Therefore, this research intends to propose a different approach than the existing ones; that is, instead of using a control mask to select the users' choice, the appliance itself is highlighted by flashing the symbols in an indicator panel near each appliance (Fig. 4). This provides the user with a visual stimulus which will give the P300 response in EEG signals.

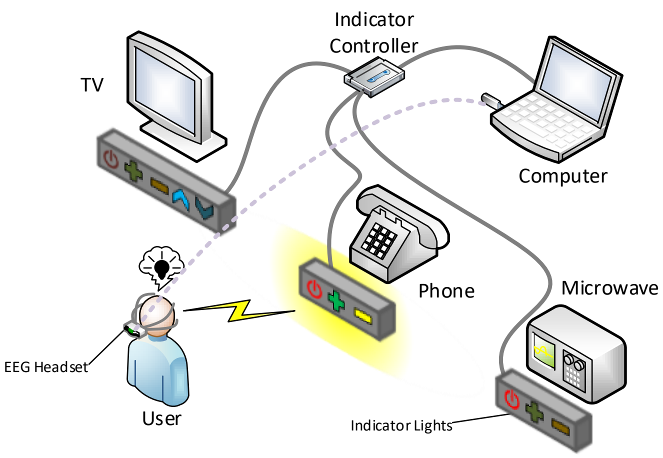


Fig. 4 System overview: The user concentrates on the indicator of a certain appliance.

In order to record these responses effectively, signals will be captured from multiple positions in the scalp (Fig. 5). Since the visual P300 response is present more prominently around the visual cortex, multiple electrode positions (Cz, P3, Pz, P4, O1, O2) around occipital and parietal regions will be used for capturing the response. Two positions in the frontal region (Fp1 and Fp2) are also used to detect the electrooculography (EOG) artefacts of eye movement and eye blinks. These signal channels will be monitored and will help in automated artefact removal where the EEG signals will automatically be discarded and will not be used for processing if artefacts are present in the EOG data at the same time period.

In the next step these signals are conditioned. The EEG signals are band−passed at 1−20Hz filtering the unwanted frequencies. Then the artefacts in the EEG signals are removed, which can be automated as described before. Finally the signals are processed and interpreted as control commands.

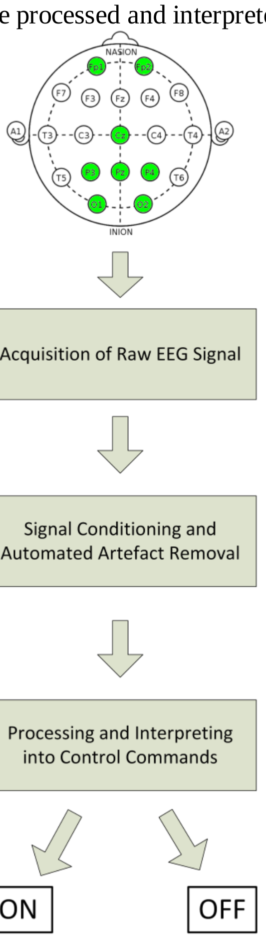


Fig. 5 Process flow: EEG signal acquisition from multiple electrodes placed in the scalp, signal conditioning, processing and interpreting signals into control commands.

Selection of the command will be done by flashing an indicator panel that is placed next to the relevant appliance. These commands can be used either to select an appliance or to control an already selected appliance. The EEG signals will be then acquired and processed. The system decides that the user concentrated on a specific command by the presence of P300 response in the detected EEG signals acquired in a time–locked interval after the flash relevant to the command. Next the indicator panel will flash a set of different symbols associated with the set of control commands for the selected appliance. Here the user concentrates on the control command he wants to issue to the appliance. The correct command will be detected by the system in the same manner the appliance was selected.

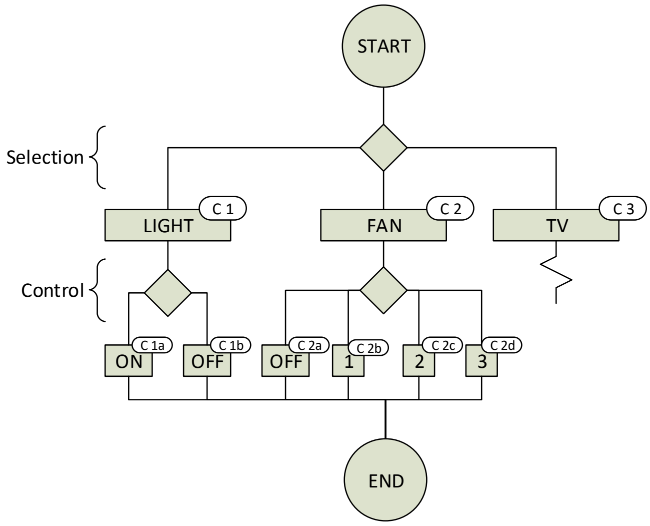


Fig. 6 Typical flow of control of the system showing selection and control phases

The flash controller flashes the symbols associated with the commands in a random order. The flash controller also sends a real–time stream of markers to the signal processing application (eg: C3, C1b, C2d, C3a…) which corresponds to the command that is being flashed at the moment.

The signal processing application receives four types of data streams: EOG streams (from electrodes Fp1 and Fp2), EEG signal channels (from positions (Cz, P3, Pz, P4, O1, O2), signal from a head–mounted accelerometer and marker stream.

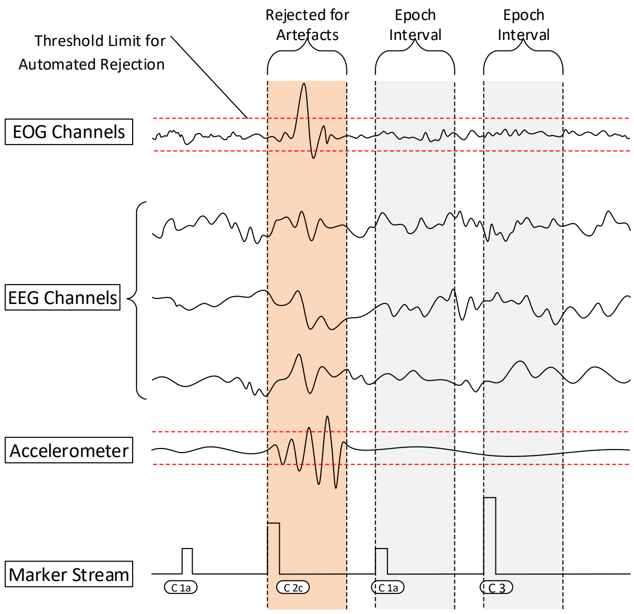


Fig. 7 Four types of data streams received by the application

The EOG signal and accelerometer input are used for automated artefact detection and removal. EOG signal will be used to detect eye movement and eye blinks while the accelerometer will detect the movements of neck and other head muscles. Both these movements introduce unwanted noise to the

EEG signal, and removing these can improve the accuracy of the system [25]. Automated artefact removal is performed by checking the signal stream for sections that contain signal values that exceed a certain pre−set threshold, and rejecting sections of the signal stream that contain such values.

After artefact removal the EEG signal is band−passed (typically at 1−20Hz) to remove the unwanted frequency components and retain only the components required for classification. Then the time−locked intervals (epochs) that correspond to the command markers in the marker stream are extracted from this averaged EEG signal stream (Fig. 8, top). Note that some of these extracted epoch intervals might be overlapping, since the inter−flash interval is not necessarily greater than the extracted epoch interval.

After extracting epochs for a repeated number of trials, the epochs are averaged for each command marker that was flashed. This highlights the P300 response from the otherwise−noisy EEG signals. After averaging epochs, the characteristic P300 response will become prominent only for the target command channel while the non−target command channels will have an irregular and low response (Fig. 8, bottom)

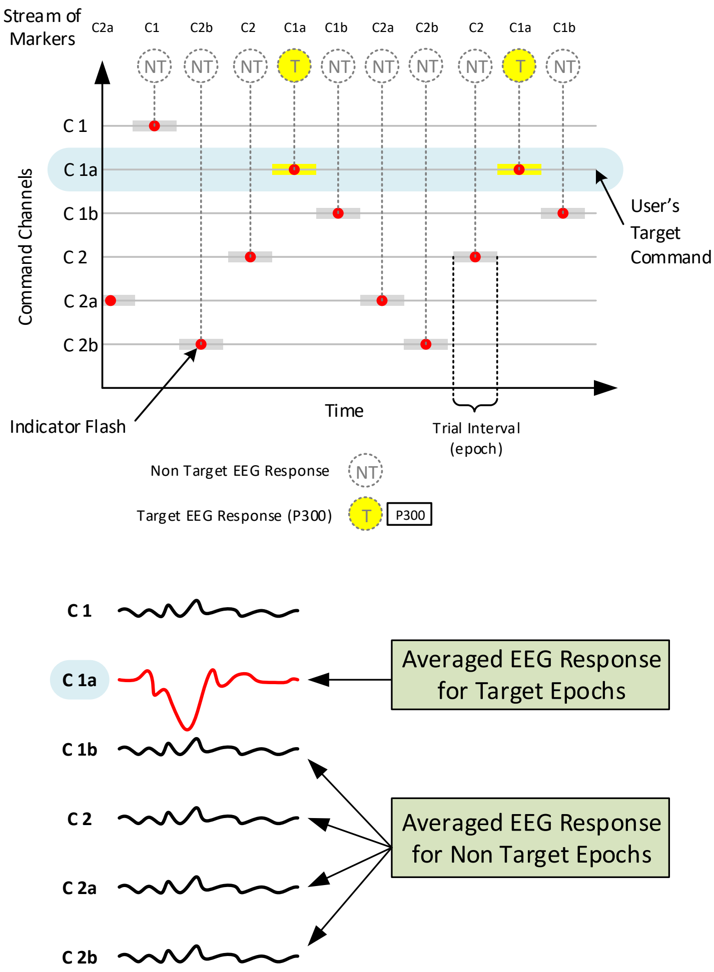


Fig. 8 Time−locked intervals for flash markers in multiple control channels (top). Here the target P300 response is only observed for the time−locked intervals to the marker of target command (bottom).

The averaged response can be classified into targets and non−targets by several methods such as Support Vector Machines (SVMs) or statistical analysis [25].

A less complicated method for classification is calculating the area under the curve within a pre−calibrated time interval where we expect the P300 peak to be present. The marker with a curve that has the most negative area in the interval is chosen as the classifier's decision.

After classification, the target command is selected as the command that the user intended to communicate to the system, and the appliances are controlled accordingly.

B. Tools and Equipment Intended to be Used

We intend to use an Emotiv EPOC EEG headset [24] for capturing of EEG, EOG, and accelerometer data. Emotiv EPOC is a relatively low−cost EEG device which is capable of signal acquisition from 14 electrode positions in the scalp, up to a sampling rate of 256Hz. The Analog−to−Digital converter (ADC) of this device is capable of conversion with a resolution of 14bits which gives $2^{14} = 16384$ signal levels per sample. Research has shown that this device is capable of capturing the P300 response [19].

For the signal processing and classification, we intend to use the EEGLAB toolkit for MATLAB [26]. This toolkit makes it easier to process multi−channel EEG data since it contains pre−defined functions for signal filtering, epoch extraction, averaging and visualization. To simulate preliminary P300 experiments, we have used OpenViBE BCI software [27].

V. Conclusions and Discussion

Compared to other assistive technologies and alternative forms of environmental control, BCI based systems offer a truly hands−free experience for the users. However, the most commonly used techniques for these systems are not very intuitive. The technique used in P300 based environmental control systems could be improved as proposed by this research which will be a significant improvement to the similar systems that have been developed during the past few years.

The proposed system has several differences than the ordinary control−mask based systems. When selecting the appliance, the system has to detect the P300 response for a lower number of commands than the control−mask based systems (limited to the number of appliances in the room the user is in). After the appliance is selected, the number of selections is reduced to the number of different control commands for the selected appliance. Having a lower number of selections at each stage can improve the speed of the system since the total number of trials needed for a selection will be reduced.

Also, the user will have to be in the same room as the appliance he intends to control, and the appliance should be in the line of the user and cannot be occluded, unlike in control−mask based systems where the user could control any appliance that is shown in the display screen.

VI. Future Work

This paper only presents the concept of a system that eliminates several drawbacks of the existing systems. A future work of this might be to develop this system and evaluate its performance. This proposed system only focuses on utilizing the P300 evoked potential to make selection Therefore this work can be extended by combining this technique with another type of BCI such as steady−state visually evoked potentials or another type of biofeedback such as electrooculography or electromyography to form a hybrid BCI.

Acknowledgment

The authors would like to thank the University of Colombo School of Computing# Electronics Lab for supplying us with equipment required for the work and the staff at Electronics Lab for helping us conduct the required experiments.

References

[40] Guger, C., Schlogl, A., Neuper, C., Walterspacher, D., Strein, T., & Pfurtscheller, G. (2001). Rapid prototyping of an EEG−based brain−computer interface (BCI). *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *9*(1), 49–58.

[41] Guger, C., Holzner, C., & Groenegress, C. (2008). Control of a smart home with a brain−computer interface. *Brain–Computer Interface*, (i), 2–6.

[42] Dean J Krusienski, Eric W Sellers, François Cabestaing, Sabri Bayoudh, Dennis J Mcfarland, et al.. A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering, IOP Publishing*, 2006, 3 (4), pp.299−305.

[43] McFarland, D. J., & Wolpaw, J. R. (2011). Brain−computer interfaces for communication and control. *Communications of the ACM*, *54*(5), 60.

[44] Craig, A., Tran, Y., McIsaac, P., & Boord, P. (2005). The efficacy and benefits of environmental control systems for the severely disabled. *Medical Science Monitor : International Medical Journal of Experimental and Clinical Research*, *11*(1), RA32−A39.

[45] SU, M.−C., SU, S.−Y., & CHEN, G.−D. (2005). a Low−Cost Vision−Based Human−Computer Interface for People With Severe Disabilities. *Biomedical Engineering: Applications, Basis and Communications*, *17*(6), 284–292.

[46] Kilgore, K. L., Hoyen, H. A., Bryden, A. M., Hart, R. L., Keith, W., & Peckham, P. H. (2009). An Implanted Upper−Extremity Neuroprosthesis Using Myoelectric Control. *J Hand Surg Am*, *33*(4), 539–550.

[47] Cristian−Cezar P., Alexandra C., Alina P., Doru T. Evaluation of a P300−Based Interface for Smart Home Control. Luis M. Camarinha−Matos; Ehsan Shahamatnia; Gonçalo Nunes. *3rd Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS)*, Feb 2012, *Springer, IFIP Advances in Information and Communication Technology*, AICT−372, pp.179−186, 2012,

[48] Leeb, R., Friedman, D., Müller−Putz, G. R., Scherer, R., Slater, M., & Pfurtscheller, G. (2007). Self−paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic. *Computational Intelligence and Neuroscience, 2007*.

[49] Tai, K., Blain, S., & Chau, T. (2008). A Review of Emerging Access Technologies for Individuals With Severe Motor Impairments. *Assistive Technology*, *20*(4), 204–221.
[50] Wolpaw, J. R., & McFarland, D. J. (2004). Control of a two−dimensional movement signal by a noninvasive brain−computer interface in humans. *Proceedings of the National Academy of Sciences*, *101*(51), 17849–17854.
[51] (2018) Stephen Hawking, Official Website / The Computer [Online] Available: http://www.hawking.org.uk/the−computer.html
[52] Verstraete, M., Derboven, J., Gemmeke, J., Karsmakers, P., Broeck, B. Van Den, & Van Hamme, H. (2012). The design of voice controlled assistive technology for people with physical disabilities. *Proceedings of the First International Workshop on Language Technology in Pervasive Computing (LTPC)*, 2–5.
[53] Solanki, U. V., & Desai, N. H. (2011). Hand gesture based remote control for home appliances: Handmote. *Information and Communication Technologies (WICT), 2011 World Congress on*, 419–423.
[54] Reilly, R. B., & O'Malley, M. J. (1999). Adaptive noncontact gesture−based system for augmentative communication. *IEEE Transactions on Rehabilitation Engineering*, *7*(2), 174–182.
[55] Verstraete, M., Derboven, J., Gemmeke, J., Karsmakers, P., Broeck, B. Van Den, & Van Hamme, H. (2012). The design of voice controlled assistive technology for people with physical disabilities. *Proceedings of the First International Workshop on Language Technology in Pervasive Computing (LTPC)*, 2–5.
[56] Soderlund Staffan "Method For Playing Games Using Brain Waves" International Patent Application WO2004SE01778 20041129 21st July, 2005
[57] Chen, S. C., See, A. R., Chen, Y. J., Yeng, C. H., & Liang, C. K. (2013). The use of a brain computer interface remote control to navigate a recreational device. *Mathematical Problems in Engineering*, *2013*.
[58] Ekanayake, H. (2010). "P300 and Emotiv EPOC: Does Emotiv EPOC capture real EEG?" Retrieved from: http://neurofeedback.visaduma.info/emotivresearch.htm
[59] Aqeel−ur−Rehman, Arif, R., & Khursheed, H. (2014). Voice Controlled Home Automation System for the Elderly or Disabled People. *Journal Applied Environmental and Biological Sciences*, *4*(8), 55–64.
[60] Aloise, F., Schettini, F., Aricò, P., Leotta, F., Salinari, S., Mattia, D., Cincotti, F. (2011). P300−based brain−computer interface for environmental control: An asynchronous approach. *Journal of Neural Engineering*, *8*(2). https://doi.org/10.1088/1741−2560/8/2/025025
[61] Zhang, H., Guan, C., & Wang, C. (2008). Asynchronous P300−based brain − Computer interfaces: A computational approach with statistical models. *IEEE Transactions on Biomedical Engineering*, *55*(6), 1754–1763.
[62] Homepage of OpenBCI [Accessible online] http://openbci.com/
[63] Product page of Emotiv EPOC [Online]
Available https://www.emotiv.com/epoc/
[64] Kaper, M, *P300 Based Brain Computer Interfacing* 2006.
[65] (2018) EEGLAB Homepage. [Online]
Available: https://sccn.ucsd.edu/eeglab/index.php
[66] (2018) OpenViBE Homepage [Online]
Available: http://openvibe.inria.fr/

Appendix B: Diagrams

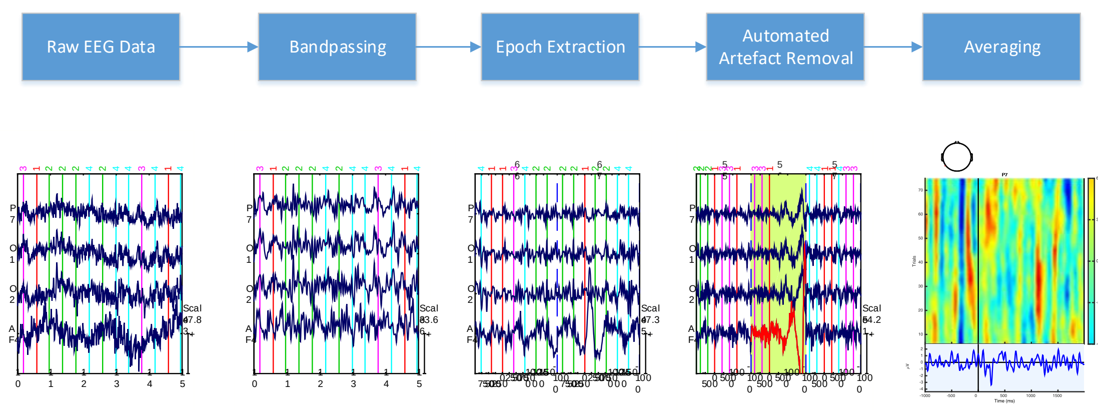


Figure 0.1 A visualization of the EEG signal processing pipeline

Appendix C: Code Listings

Data Collection Component (C++)

```
void EmotivDataCollector(void *unused) {
        try {
                EE_DataChannel_t targetChannelList[] = {
                        ED_COUNTER,
                        ED_AF3, ED_F7, ED_F3, ED_FC5, ED_T7,
                        ED_P7, ED_O1, ED_O2, ED_P8, ED_T8,
                        ED_FC6, ED_F4, ED_F8, ED_AF4, ED_GYROX, ED_GYROY, ED_TIMESTAMP,
                        ED_FUNC_ID, ED_FUNC_VALUE, ED_MARKER, ED_SYNC_SIGNAL
                };

                const char header[] = "COUNTER,AF3,F7,F3,FC5,T7,P7,O1,O2,P8"
                        ",T8,FC6,F4,F8,AF4,GYROX,GYROY,TIMESTAMP,"
                        "FUNC_ID,FUNC_VALUE,MARKER,SYNC_SIGNAL,STIM,";

                EmoEngineEventHandle eEvent = EE_EmoEngineEventCreate();
                EmoStateHandle eState = EE_EmoStateCreate();
                unsigned int userID = 0;
                const unsigned short composerPort = 1726;
                float secs = 1;
                unsigned int datarate = 0;
                bool collectEEG = false;
                int option = 0;
                int state = 0;

                if (EE_EngineConnect() != EDK_OK) {
                        std::cout << "Emotiv Engine start up failed." << std::endl;
                        running = false;
                }
                else {
                        std::stringstream filename, filename2, source, destination, srcPath,
destPath, destName;
                        //Copying paths
                        source << "C:/Users/Tharinda/Documents/Visual Studio
2013/Projects/EmotivStim/Debug/";

                        destination <<
"C:/Users/Tharinda/Documents/MATLAB/eeglab14_1_1b/eeg-common/";
                        destName << "templog.csv";


                        filename << "eeglog-" << GetTimeStr() << ".csv"; // EEG log filename

                        std::ofstream ofs(filename.str(), std::ios::trunc);
                        ofs << header << std::endl;

                        //filename2 << "preprocessed-log-" << GetTimeStr() << ".csv"; // EEG
log filename
                        filename2 << "preprocessed-log.csv"; // COMMON EEG log filename
                        std::ofstream ofs2(filename2.str(), std::ios::trunc);

                        DataHandle hData = EE_DataCreate();
                        EE_DataSetBufferSizeInSec(secs);

                        std::cout << "EEG buffer size in secs:" << secs << std::endl;

                        //std::cout.setstate(std::ios_base::failbit);
                        //std::streambuf* cout_sbuf = std::cout.rdbuf(); // save original
sbuf
                        //std::ofstream   fout("/dev/null");
                        //std::cout.rdbuf(fout.rdbuf()); // redirect 'cout' to a 'fout'
```

```
			while (running) {
				state = EE_EngineGetNextEvent(eEvent);

				if (state == EDK_OK) {
					EE_Event_t		eventType		=
EE_EmoEngineEventGetType(eEvent);
					EE_EmoEngineEventGetUserId(eEvent, &userID);

					// Log the EmoState if it has been updated
					if (eventType == EE_UserAdded) {
						//std::cout << "User added";
						EE_DataAcquisitionEnable(userID, true);
						collectEEG = true;
					}

					// Extended...
					// Log the EmoState if it has been updated
					//if (eventType == EE_EmoStateUpdated) {
					//	EE_EmoEngineEventGetEmoState(eEvent, eState);
					//	//const	float	timestamp	=
ES_GetTimeFromStart(eState);
					//	//printf("%10.3fs : New EmoState from user %d
...\r", timestamp, userID);
					//	//logEmoState(ofs,	userID,	eState,
writeHeader);
					//	//writeHeader = false;
					//}
				}


				if (collectEEG) {
					EE_DataUpdateHandle(0, hData);

					unsigned int nSamplesTaken = 0;
					EE_DataGetNumberOfSample(hData, &nSamplesTaken);

					if (nSamplesTaken != 0) {

						double* data = new double[nSamplesTaken];
						for	(int	sampleIdx	=	0;
sampleIdx<(int)nSamplesTaken; ++sampleIdx) {
							for	(int	i	=	0;
i<sizeof(targetChannelList) / sizeof(EE_DataChannel_t); i++) {
								int	channel	=
targetChannelList[i];
								EE_DataGet(hData,
targetChannelList[i], data, nSamplesTaken);
								ofs	<<	data[sampleIdx]	<<
",";
								if	(channel	==	ED_P7	||
channel == ED_O1 || channel == ED_O2 || channel == ED_AF4){
									ofs2		<<
data[sampleIdx] << ",";
								}
							}
							ofs << marker << std::endl;
							ofs2 << marker << std::endl;
							marker = 0;
						}
						delete[] data;
					}
				}
				Sleep(1);
			}

			//std::cout.rdbuf(cout_sbuf); // restore the original stream buffer
			//std::cout.clear();

			ofs.close();
			ofs2.close();
			EE_DataFree(hData);
			/*char*  sourceDir  =  "C:\\Users\\Tharinda\\Documents\\Visual  Studio
2013\\Projects\\EmotivStim\\Debug\\";
```

```
			strcat(sourceDir, "preprocessed-log.csv");
			char*				destDir					=
"C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\";
			strcat(destDir, "preprocessed-log.csv");
			copyFile(sourceDir, destDir);*/


			/*srcPath << source.str() << filename2.str();
			LPCWSTR src = (LPCWSTR)srcPath.str().c_str();
			destPath << destination.str() << destName.str();
			LPCWSTR dest = (LPCWSTR)destPath.str().c_str();
			CopyFile(src, dest, TRUE);*/


		}
		//	catch (const std::exception& e) {
		//		std::cerr << e.what() << std::endl;
		//		std::cout << "Press any key to exit..." << std::endl;
		//		getchar();
		//	}

		EE_EngineDisconnect();
		EE_EmoStateFree(eState);
		EE_EmoEngineEventFree(eEvent);
	}
	catch (...) {
		std::cout << "Exception occured in the EEG logger!" << std::endl;
		running = false;
	}
	std::cout << "Exiting from Emotiv connector..." << std::endl;
	std::cout << std::endl << std::endl << "Data collection ended" << std::endl;
	std::cout << "Press any key to continue" << std::endl;
	if (!aborted){
		//			ENTER			key			down
*****************************************************SIMULATED KEYPRESS
		keybd_event(VK_RETURN, 0x9C, 0, 0);
		// ENTER key up
		keybd_event(VK_RETURN, 0x9C, 0, 0);
	}
}
```

Classifier (C++)

```
template<size_t R, size_t C>
int P300Classifier_AdaptWeightGauss(double center, double radius,int experiment, char mode,
int target, double (&history)[R][C]){
	std::cout << "Model based - Adaptive Hybrid Classifier With Gaussian Filtering and
Weighted Channel Ensembling" << std::endl;

	int TARGETS = 4; //hardcoded

	double SEARCH_START, SEARCH_END, SEARCH_RADIUS = radius, SEARCH_CENTER = center; //
start from around 300
	SEARCH_START = SEARCH_CENTER - SEARCH_RADIUS;
	SEARCH_END = SEARCH_CENTER + SEARCH_RADIUS;
	std::cout << "\t > Search Start > " << SEARCH_START << "\t < Search End > " <<
SEARCH_END << std::endl;

	//Getting file streams
	std::ifstream
infile1("C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\mat-to-cpp\\epochs
-1-ar.avg");
	std::ifstream
infile2("C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\mat-to-cpp\\epochs
-2-ar.avg");
	std::ifstream
infile3("C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\mat-to-cpp\\epochs
-3-ar.avg");
```

```
	std::ifstream
infile4("C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\mat-to-cpp\\epochs
-4-ar.avg");

	int count = 0;

	std::string header;
	double a, b, c, d, e;

	double buffer1[128][8], buffer2[128][8], buffer3[128][8], buffer4[128][8];

	//Scanning files and loading into buffers
	printf("\nScanning File epochs-1 \n");
	std::getline(infile1, header);
	while (infile1 >> a >> b >> c >> d >> e) {
		if (count >= 128 && count < 256){
			buffer1[count - 128][0] = a;
			buffer1[count - 128][1] = b;
			buffer1[count - 128][2] = c;
			buffer1[count - 128][3] = d;
			buffer1[count - 128][4] = e;

			//printf("time: %f AF4: %f P7: %f O1: %f O2: %f \n", a, b, c, d, e);
		}
		count++;
	}
	count = 0;

	printf("\nScanning File epochs-2 \n");
	std::getline(infile2, header);
	while (infile2 >> a >> b >> c >> d >> e) {
		if (count >= 128 && count < 256){
			buffer2[count - 128][0] = a;
			buffer2[count - 128][1] = b;
			buffer2[count - 128][2] = c;
			buffer2[count - 128][3] = d;
			buffer2[count - 128][4] = e;

			//printf("time: %f AF4: %f P7: %f O1: %f O2: %f \n", a, b, c, d, e);
		}
		count++;
	}
	count = 0;

	printf("\nScanning File epochs-3 \n");
	std::getline(infile3, header);
	while (infile3 >> a >> b >> c >> d >> e) {
		if (count >= 128 && count < 256){
			buffer3[count - 128][0] = a;
			buffer3[count - 128][1] = b;
			buffer3[count - 128][2] = c;
			buffer3[count - 128][3] = d;
			buffer3[count - 128][4] = e;

			//printf("time: %f AF4: %f P7: %f O1: %f O2: %f \n", a, b, c, d, e);
		}
		count++;
	}
	count = 0;

	printf("\nScanning File epochs-4 \n");
	std::getline(infile4, header);
	while (infile4 >> a >> b >> c >> d >> e) {
		if (count >= 128 && count < 256){
			buffer4[count - 128][0] = a;
			buffer4[count - 128][1] = b;
			buffer4[count - 128][2] = c;
			buffer4[count - 128][3] = d;
			buffer4[count - 128][4] = e;

			//printf("time: %f AF4: %f P7: %f O1: %f O2: %f \n", a, b, c, d, e);
		}
		count++;
	}
```

```
        count = 0;

        //Closing file streams
        infile1.close();
        infile2.close();
        infile3.close();
        infile4.close();

        double total1 = 0, total2 = 0, total3 = 0, total4 = 0;

        static double peaks[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

        double peakPos1 = 250.0, peakPos2 = 250.0, peakPos3 = 250.0, peakPos4 = 250.0 ;

        double peak1 = 0, peak2 = 0, peak3 = 0, peak4 = 0;

        //Peak picking: This gives the negative peak value and its position in P7 channel
        //only the peaks between search start and search end are considered
        for (int i = 0; i < 128; i++){
                double v = buffer1[i][2];
                if  (buffer1[i][0]  >=  SEARCH_START  &&  buffer1[i][0]  <=  SEARCH_END  &&  v  <
peak1){
                        peak1 = v;
                        peakPos1 = buffer1[i][0];
                }

        }

        for (int i = 0; i < 128; i++){
                double v = buffer2[i][2];
                if  (buffer1[i][0]  >=  SEARCH_START  &&  buffer1[i][0]  <=  SEARCH_END  &&  v  <
peak2){
                        peak2 = v;
                        peakPos2 = buffer2[i][0];
                }
        }

        for (int i = 0; i < 128; i++){
                double v = buffer3[i][2];
                if  (buffer1[i][0]  >=  SEARCH_START  &&  buffer1[i][0]  <=  SEARCH_END  &&  v  <
peak3){
                        peak3 = v;
                        peakPos3 = buffer3[i][0];
                }
        }

        for (int i = 0; i < 128; i++){
                double v = buffer4[i][2];
                if  (buffer1[i][0]  >=  SEARCH_START  &&  buffer1[i][0]  <=  SEARCH_END  &&  v  <
peak4){
                        peak4 = v;
                        peakPos4 = buffer4[i][0];
                }
        }

        if (mode == 'A' && experiment < 10){
                switch (target){
                case 1 : SEARCH_CENTER = peakPos1; break;
                case 2 : SEARCH_CENTER = peakPos2; break;
                case 3 : SEARCH_CENTER = peakPos3; break;
                case 4 : SEARCH_CENTER = peakPos4; break;
                default: break;
                }
                SEARCH_RADIUS -= 20;

        }

        // Wave analysis
        for (int i = 0; i < 128; i++){
                double  v  =  gaussian(buffer1[i][0],  peakPos1)  *  weighting(buffer1[i][1],
buffer1[i][2], buffer1[i][3], buffer1[i][4]);
                buffer1[i][5] = v;
                total1 += (-v);
        }
```

```
        for (int i = 0; i < 128; i++){
                double  v  =  gaussian(buffer2[i][0],  peakPos2)  *  weighting(buffer2[i][1],
buffer2[i][2], buffer2[i][3], buffer2[i][4]);
                buffer2[i][5] = v;
                total2 += (-v);
        }
        for (int i = 0; i < 128; i++){
                double  v  =  gaussian(buffer3[i][0],  peakPos3)  *  weighting(buffer3[i][1],
buffer3[i][2], buffer3[i][3], buffer3[i][4]);
                buffer3[i][5] = v;
                total3 += (-v);
        }
        for (int i = 0; i < 128; i++){
                double  v  =  gaussian(buffer4[i][0],  peakPos4)  *  weighting(buffer4[i][1],
buffer4[i][2], buffer4[i][3], buffer4[i][4]);
                buffer4[i][5] = v;
                total4 += (-v);
        }


        printf("\n\nScan Ended\nTotal 1: %f\nTotal 2: %f\nTotal 3: %f\nTotal 4: %f\n", total1,
total2, total3, total4);
        double totals[4] = { total1, total2, total3, total4 };
        std::sort(totals, totals + 4);

        int choice = 0;
        double total = totals[0];
        std::cout << "Decision of Classifier: -------- ";
        if (total == total1){
                choice = 1;
                history[experiment][4] = peakPos1;
                printf("MARKER 1\n");
        }
        else if (total == total2){
                choice = 2;
                history[experiment][4] = peakPos2;
                printf("MARKER 2\n");
        }
        else if (total == total3){
                choice = 3;
                history[experiment][4] = peakPos3;
                printf("MARKER 3\n");
        }
        else if (total == total4){
                choice = 4;
                history[experiment][4] = peakPos4;
                printf("MARKER 4\n");
        }

        double confidence = ((total - ((totals[1] < 0) ? totals[1] : 0)) / total) * 100;

        history[experiment][3] = confidence;

        std::cout << "Confidence: " << confidence  << "% " << std::endl;
        return choice;
}
```

Arduino Code

```
#define BAUD_RATE 115200

#define B_1 2
#define G_1 3
#define R_1 4

#define B_2 5
#define G_2 6
#define R_2 7
```

```
#define B_3 8
#define G_3 9
#define R_3 10

#define B_4 11
#define G_4 12
#define R_4 13

#define R_LIMIT 250
#define G_LIMIT 50
#define B_LIMIT 25

#define CON_0 10
#define CON_1 16
#define CON_2 14
#define CON_3 15

#define GREEN 256
#define RED 0
#define ORANGE 50
#define WHITE 768

const char APPLIANCE_ID = '3'; // The ID assigned to the appliance

int brightness1 = 40;
int brightness2 = 160;
char incoming;
int led = LOW;
unsigned long currentTime = 0;
unsigned long flashStart = 0;
long duration = 50; //flash duration
char controlState = 'a'; // a: not selected b: selected c: idle d: ready
bool resting = true;

void setup() {

  Serial.begin(BAUD_RATE);

}

void loop() {
//  while(resting){
//    showOff(5);
//  }
  char incoming2;
  if(Serial.available()){
    incoming2 = Serial.read();
    serialFlush();

    //THE GAP
    flashStart = millis();
    //THE GAP

    Serial.print(incoming2);
    Serial.flush();
  }
  stimFlash(incoming2);

}

void stimFlash(char incoming){
  if(incoming == '1'){
    showRGB(1, 256, brightness2);
    showRGB(2, 0, brightness1);
    showRGB(3, 0, brightness1);
    showRGB(4, 0, brightness1);
  }
  else if(incoming == '2'){
    showRGB(1, 0, brightness1);
    showRGB(2, 256, brightness2);
    showRGB(3, 0, brightness1);
    showRGB(4, 0, brightness1);
  }
  else if(incoming == '3'){
    showRGB(1, 0, brightness1);
    showRGB(2, 0, brightness1);
```

```
    showRGB(3, 256, brightness2);
    showRGB(4, 0, brightness1);
  }
  else if(incoming == '4'){
    showRGB(1, 0, brightness1);
    showRGB(2, 0, brightness1);
    showRGB(3, 0, brightness1);
    showRGB(4, 256, brightness2);
  }

  else if(incoming == 'c'){
    duration = 5000;
    showRGB(1, 128, brightness1);
    showRGB(2, 128, brightness1);
    showRGB(3, 128, brightness1);
    showRGB(4, 128, brightness1);
  }

  else if(incoming == 'd'){
    duration = 50;
    showRGB(1, 256, 0);
    showRGB(2, 256, 0);
    showRGB(3, 256, 0);
    showRGB(4, 256, 0);
  }

  currentTime = millis();
  if(currentTime > flashStart + duration){
    showRGB(1, 256, 0);
    showRGB(2, 256, 0);
    showRGB(3, 256, 0);
    showRGB(4, 256, 0);
  }
}

void showOff(int breathe){
  for(int i=0; i<2000; i++){
    showRGB(1, (i + 0)%768, brightness1);
    showRGB(2, (i+192)%768, brightness1);
    showRGB(3, (i+384)%768, brightness1);
    showRGB(4, (i+576)%768, brightness1);
    delay(breathe);
  }
  for(int i=0; i<40; i++){
    showRGB(1, (i*192 + 0)%768, brightness1);
    showRGB(2, (i*192+192)%768, brightness1);
    showRGB(3, (i*192+384)%768, brightness1);
    showRGB(4, (i*192+576)%768, brightness1);
    delay(200);
  }
}

void showRGB(int LED, int color, int brightness1){
  int redIntensity;
  int greenIntensity;
  int blueIntensity;

  if (color < 256) {          // zone 1
    redIntensity = 255 - color;    // red goes from on to off
    greenIntensity = color;        // green goes from off to on
    blueIntensity = 0;             // blue is always off
  }
  else if (color < 512){
    redIntensity = 0;                     // red is always off
    greenIntensity = 255 - (color - 256); // green on to off
    blueIntensity = (color - 256);        // blue off to on
  }
  else if (color < 768){
    redIntensity = (color - 512);         // red off to on
    greenIntensity = 0;                   // green is always off
    blueIntensity = 255 - (color - 512);  // blue on to off
  }
  else{ // white
    redIntensity = 255;         // red off to on
    greenIntensity = 255;                   // green is always off
    blueIntensity = 255;  // blue on to off
  }
```

```
  startFlash(LED, redIntensity, greenIntensity, blueIntensity, brightness1);
}

void startFlash(int LED, int R, int G, int B,int brightness1){
  LED = LED *3;
  analogWrite(LED+1, brightScale(R,brightness1,R_LIMIT));
  analogWrite(LED, brightScale(G,brightness1,G_LIMIT));
  analogWrite(LED-1, brightScale(B,brightness1,B_LIMIT));
}

int brightScale(int intensity, int brightness1, int limit){
  float fintensity = float(intensity);
  float fbrightness1 = float(brightness1);
  float flimit = float(limit);
  float result = fintensity * (fbrightness1/255.0) * (flimit/255.0);
  return int(result);
}

void serialFlush(){
  while(Serial.available() > 0) {
    char t = Serial.read();
  }
}
```

MATLAB Data Processing Automation Code

```
M = csvread('preprocessed-log-07.10.2018-16.02.47.csv');

%csvread

eegdata = M;
eegdata = eegdata';

eeglab
EEG = pop_importdata('data',eegdata,'srate',128);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,0,'setname','eegdata','gui','off');
EEG = eeg_checkset( EEG );

EEG = pop_chanevent(EEG,5,'edge','leading','edgelen',0);
EEG     =     pop_chanedit(EEG,     'load',{'emotivbox-6electrode-2.ced'     'filetype'
'autodetect'});
EEG = pop_eegfilt(EEG, 1, 0, [], [0]);
EEG = pop_eegfilt(EEG, 0, 20, [], [0]);
eeglab redraw;

%>>>>>>>>>>>>>>>>>marker 1
EEG = pop_epoch(EEG, {'1'}, [-1 2], 'newname','epochs_1');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,1,'gui','off');
EEG = pop_rmbase( EEG, [-1000 0]);
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>auto reject
EEG = pop_eegthresh(EEG,1,[1:4] ,-40,40,-1,1.9922,0,0);
EEG = eeg_checkset( EEG );
%eeglab redraw;
EEG = eeg_rejsuperpose( EEG, 1, 0, 1, 0, 0, 0, 0, 0);
EEG = pop_rejepoch( EEG, EEG.reject.rejglobal ,0);
eeglab redraw;

pop_export(EEG,'C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\epo
chs-1-ar.avg','erp','on','transpose','on','precision',4);


%>>>>>>>>>>>>>>>>>marker 2
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 3,'retrieve',1,'study',0);
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
EEG = pop_epoch(EEG, {'2'}, [-1 2], 'newname','epochs_2');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,4,'gui','off');
EEG = pop_rmbase( EEG, [-1000 0]);
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>auto reject
EEG = pop_eegthresh(EEG,1,[1:4] ,-40,40,-1,1.9922,0,0);
EEG = eeg_checkset( EEG );
%eeglab redraw;
EEG = eeg_rejsuperpose( EEG, 1, 0, 1, 0, 0, 0, 0, 0);
```

```
EEG = pop_rejepoch( EEG, EEG.reject.rejglobal ,0);
eeglab redraw;

pop_export(EEG,'C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\epo
chs-2-ar.avg','erp','on','transpose','on','precision',4);



%>>>>>>>>>>>>>>>>>marker 3
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 5,'retrieve',1,'study',0);
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
EEG = pop_epoch(EEG, {'3'}, [-1 2], 'newname','epochs_3');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,5,'gui','off');
EEG = pop_rmbase( EEG, [-1000 0]);
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>auto reject
EEG = pop_eegthresh(EEG,1,[1:4] ,-40,40,-1,1.9922,0,0);
EEG = eeg_checkset( EEG );
EEG = eeg_rejsuperpose( EEG, 1, 0, 1, 0, 0, 0, 0, 0);
EEG = pop_rejepoch( EEG, EEG.reject.rejglobal ,0);
eeglab redraw;

pop_export(EEG,'C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\epo
chs-3-ar.avg','erp','on','transpose','on','precision',4);

%>>>>>>>>>>>>>>>>>marker 4
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 7,'retrieve',1,'study',0);
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, CURRENTSET);
EEG = pop_epoch(EEG, {'4'}, [-1 2], 'newname','epochs_4');
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG,7,'gui','off');
EEG = pop_rmbase( EEG, [-1000 0]);
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>auto reject
EEG = pop_eegthresh(EEG,1,[1:4] ,-40,40,-1,1.9922,0,0);
EEG = eeg_checkset( EEG );
%eeglab redraw;
EEG = eeg_rejsuperpose( EEG, 1, 0, 1, 0, 0, 0, 0, 0);
EEG = pop_rejepoch( EEG, EEG.reject.rejglobal ,0);
eeglab redraw;

pop_export(EEG,'C:\\Users\\Tharinda\\Documents\\MATLAB\\eeglab14_1_1b\\eeg-common\\epo
chs-4-ar.avg','erp','on','transpose','on','precision',4);
```

Appendix D: User Feedback Form

Post-experiment questionnaire (to be filled by the participant)

#	Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	The headband was comfortable to wear					
2	I did not experience eye fatigue					
3	I am okay with the application of conductive gel that was used					
4	I believe that this system is safe for the users					
5	The extra safety explanations and procedures helped me be more confident about using this					
6	The extra safety explanations and procedures scared me more, rather than making me more confident					
7	Using this did not increase my stress levels					
8	The investigator was polite and respectful towards me during the experiments					
9	This method of input is user friendly for me					
10	This method of input is effective for healthy users in general					
11	This method of input might be more effective and suitable for disabled and paralyzed individuals					
12	I would recommend more people to try this out					

Other comments and suggestions for improvement: