Capstone Project

Department of Mechanical Engineering

Final Report

2019

Project Title:	A Virtual Reality (VR)-based Brain-Computer Interface Development
Date:	March 25, 2020

Project Team Information

Identifier:	**CP[12-DOE-193]**	
Student workers:	Jiawei Liao,	756560
	Harfiyanto Dharma Santoso,	772503
	Albert Tien Hann,	758259
Academic Supervisor:	Assoc. Prof. Denny Oetomo	
Academic Examiner:	Dr. Alireza Mohammadi	

Version: 2.1 , March 25, 2020

Acknowledgements

We would like to thank our supervisor, Denny, for the great opportunities and support provided for us to challenge and develop our skills in this frontier area of studies. We would like to express deep gratitude to our mentor, Jing, for the generous support regarding project management, literature review and experiments design throughout this project. We would also like to thank Ricardo for the assistance with the VR environment development, which helped us save valuable time and avoid unnecessary detours.

Contents

Chapter 1

Executive Summary

The project is about developing a method of classifying brain signals into commands that are used to complete simple tasks in the Virtual Reality environment such as controlling a robot to pick up object on a table. This is important because there are sizeable number of people who are suffering from diseases or injuries that render their body or part of their body paralysed e.g. brain, spinal cord injuries, and cerebral palsy. [26] Through understanding how to decode brain signals into comprehensible set of commands, the patient could regain some form of control and therefore improving their quality of life in general. This project acts as a stepping stone in realizing that. By using VR environment, various experiments could be conducted in software, thereby avoiding unnecessary expense from buying equipment that might or might not end up in the final product. It also simplifies the experiment setup, making it easier to adjust the setup if necessary.

Scalp-electroencephalogram (S-EEG) method was used to collect brain signals from a subject using a non-invasive method of placing dry electrodes on the subject's scalp. Different positions of the node would yield different type of signals. Three known type of brain signals were investigated and compared to see which ones were more suitable for this project. After reviewing case studies done in the field, two brain signals were chosen: Steady-State Visual Evoked Potentials (SSVEP) and Motor Imagery (MI).

A preliminary set of experiments were conducted on the three group members to test the response of the two brain signals to their corresponding stimuli. From the data collected, an offline analysis was performed to develop two different algorithms to decode each signal. Filter Bank Canonical Correlation Analysis (FBCCA) were used to decode SSVEP signals. It is the improved version [22] of Canonical Correlation Analysis which is widely used among the research papers [12], [13], [17]. In this experiment, three different types of square wave steady state stimuli were tested: blinking, resizing and checker box. Comparing their performance, the blinking box was the most accurate, followed by the chess-grid and resizing stimuli. However, its accuracy came at the cost of comfort for the subject. It was straining the subject's eyes therefore making long-term use unpleasant. The Motor Imagery signal was decoded to differentiate right/left hand imagined squeezing tasks by calculating the β band power between C3 and C4 electrode using FCz as the reference. Welch's method ([2]) of power spectrum density was used to compute the power spectrum density and the β (12-30Hz) band power was estimated using numerical trapezoidal integration. After the experiment, the group member who achieved the highest accuracy (80% for 5 seconds processing window) was selected as the subject for the consequent experiment.

In the last experiment, the simple task was represented in the VR environment by picking up a ball randomly generated on an 8x8 grid. The robot is controlled using two navigation systems. A virtual robot arm was also developed in VR to perform the pickup. An enclosed room was fabricated within the VR environment, encapsulating the robot and navigation system that allows the group to conduct experiments using VR headset and the g.tech EEG cap. Three different combinations of the navigation systems and Brain computer interface methods were used to investigate their performances. Over 30 trials were conducted on a single subject, to arrive at a working virtual robot arm that can pick up a ball within an average time of 1 minute with 83.33% success rate.

Chapter 2

Introduction

2.1 Background and Motivation

According to a survey study of prevalence of paralysis across the United States in 2013, [26], approximately 1 in 50 people (a total of 5.4 million) in the U.S are living with paralysis, and far more are affected globally. [6] A wide range of diseases, such as brain and spinal cord injuries, brain stem stroke and cerebral palsy, could impair the neural pathways between the human brain and body muscles. Under extreme circumstances, patients could be 'locked-in' and unable to communicate with the external world.

To improve the life quality of paralysed patients, electroencephalographic (EEG) and other measures of brain activity have been used to create non-muscular channels between the brain and an external computer - a brain-computer-interface ([17]). The advance in computational power and signal processing techniques have allowed researchers to design and implement real-time BCIs to decode brain signals into commands. Paralysed patients could utilize a BCI to type letters, answer questions and give commands.[10]

Virtual reality has emerged in recent years as a novel asset of BCIs as it could provide a visually immersive environment. Such environments could improve the general user experience while expanding possible applications in a simulated environment, such as controlling a avatar to walk [27] and rowing a boat [31]. Still, the combination of VR and BCI is relatively new and more research of their applications should be investigated, such as controlling robots and other smart devices to assist paralysed patients with daily tasks.

Augmented reality (AR) will be a potential viable solution to achieve this goal, where development of VR based BCIs is a necessary research and prototyping stage between the traditional flat-display BCIs and the envisioned AR based BCIs. More importantly, the tasks and user interface design of VR based BCIs can be conveniently transferred to the AR environment given their similarities.

2.2 Problem Statement and tasks:

The objective of this project is to design and implement a scalp-electroencephalogram (S-EEG)-based brain computer interface (BCI) to control a robotic arm to achieve navigate and pick up tasks in a simulated virtual reality (VR) application setting, and to evaluate and analyse the performance of several possible controlling methods.

In order to accomplish the objectives of this project, the following tasks need to be completed:

1. Study of known brain signal types and choose suitable candidates as control signals for the project.
2. Study of known signal decoders and choose suitable methods for this project.
3. Design and implement VR applications for both signal performance testing experiment and also the navigate and pick up task experiment.
4. Evaluate performance of chosen control signals against the original research paper results on the three group members by offline processing; Best performing subject to to be tested in the final navigate and pick up task environment.
5. Design of control methods for the navigate and pick up task; Implementation of real time decoder in Matlab Simulink based on the analysis of results in (4.).
6. Conducting the navigate and pick up task experiment and analyse the results.

Chapter 3

Literature Review

3.1 Chapter Introduction

This literature review aims to investigate optimal approaches to construct a VR-based BCI by analysing the existing BCI control methods and several modern VR-based BCI applications. The body of this review is divided into the following subsections:

- **Scalp-Electroencephalogram (S-EEG):** An overview of the S-EEG brain signal collection method used in this project.
- **VR BCI control signals comparison and case studies:** A comparison of popular known brain signal phenomena used as control signals for VR BCIs, and a case study of VR BCI systems using these signals. Steady state visually evoked potentials (SSVEP) and motor imagery (MI) /induced event related synchronisation (ERS) are justified to be the control signal candidates for this project.
- **Analysis of VR-based BCIs applications:** An analysis of the studied VR-based BCIs and other applications of VR-based BCIs.

3.1.1 Scalp-electroencephalogram (S-EEG) BCI

Scalp-electroencephalogram (S-EEG) is a non-invasive method to monitor the electrical activity of the brain via non-invasive electrodes placed on the human scalp, and it is the most popular brain signal collection method given its relative safety, portability and low cost. [17] However, signal quality is poor since the signals have to cross many layers of body tissue before reaching the electrodes. [20] Electrocorticography (EcoG) and intracortical neuron recording produce much higher quality signals by placing electrodes directly on the surface of the brain and inside the grey matter respectively, but they introduces risks and the expensive equipment required limited their applications. Despite EEG's many advantages, extensive signal processing techniques must be applied to mitigate the poor signal quality. [17] An EEG BCI consists of the following main components:

- **Electrodes and cap:** Metallic electrodes are mounted on a soft cap and worn by the user during the signal collection. The electrodes can be dry or 'wet'. Dry electrodes are easy to use but physical disturbance due to hair movement could greatly affect the Signal-to-Noise Ratio (SNR). Wet electrodes use EEG gel to reduce the impedance between the skin and each electrode and achieve higher SNR but are cumbersome to use. The positioning of the electrodes on the Scalp usually follows the International 10–20 system [8], which is illustrated in Figure (3.1).

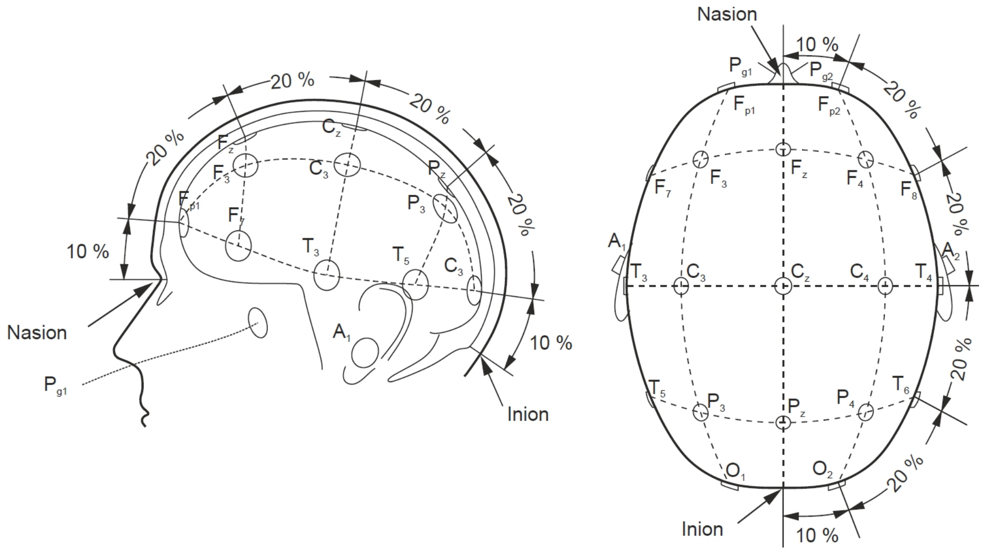


Figure 3.1: International 10-20 System [8]

- **Amplifier and ADC system:** The EEG potentials are in the micro-volt range, and need to be accurately amplified and digitally converted for signal analysis. [20]
- **Recording and analysis device:** This refers to the 'computer' in the BCI term, where a computer is used to record the collected raw signals and apply analysis to decode raw signals into usable commands. Raw signals are commonly filtered into several sub-bands before further processing, such as delta (0-4Hz), theta (4-7Hz), alpha (8-12Hz), beta (12-30Hz), and gamma (30-100Hz).

3.2 VR BCI control signals comparison and case studies:

Four main types of brain signals have been employed in existing BCI systems , visually evoked potentials, slow cortical potentials, P300 evoked potentials, and sensorimotor rhythms [17]. These four main control signals can be further divided into many sub variants. The 2013 review of BCI systems [17] provided a brief table summary of these four control signals for easier comparison.

Signal	Physiological phenomena	Number of choices	Training	Information Transfer Rate
VEP	Brain signal modulations in the visual cortex	High (enough for typing)	No	60-100 bits/min
SCP	Slow voltages shift in the brain signals	Low (2 to 4, very difficult)	Yes	5-12 bits/min
P300	Positive peaks due to infrequent stimulus	High	No	20-25 bits/min
Sensorimotor rhythms	Modulations in sensorimotor rhythms synchronized to motor activities	Low (2,3,4,5)	Yes	3 -35 bits/min

Table 3.1: Summary of control signals from (table 2, [17])

VEP, P300 and Sensorimotor rhythms have been used in VR BCI systems, where VEP and Sensorimotor rhythms are more widely adopted. A brief explanation of their principle is given below:

- **Visually evoked potentials (VEP):** VEPs are induced in the occipital lobe after receiving a visual stimulus. [4] Steady state VEPs (SSVEP) based BCIs is widely used for its high information transfer rate (ITR). [4] A visual stimulus such as a constant frequency (commonly 8-15Hz) flashing light could induce sinusoidal SSVEP at the fundamental frequency of the light source and at several subsequent harmonics. [11].

- **P300:** P300 potentials are evoked by infrequent auditory, visual, or somatosensory stimuli. Responses are elicited after paying attention to an oddball stimulus among several frequent stimuli. [3] A typical visual P300 BCI comprises a matrix of letters, numbers, or other symbols. [3] The rows or columns of this matrix are flashed at random and P300 is induced when the user sees the desired row or column flashes.
- **Sensorimotor rhythms:** Sensorimotor rhythms is associated with motor activities including both actual movements and imagined movements (motor imagery, or MI). A phenomena called event-related desynchronisation/synchronisation (ERD/ERS) occurs during motor actions and imagery of body parts movement in the corresponding brain region, where ERD refers to the amplitude increase in band rhythem and ERS refers to the amplitude decrease. [5], [7]

3.2.1 SSVEP VR BCIs

A typical SSVEP BCI displays a virtual keyboard on a flat display, where each key is a stimuli at a different frequency for the subject to gaze at. SSVEP BCIs often use canonical correlation analysis (CCA) to compute the correlation weight between raw signals and generated reference signals to determine the target. [12], [13] This method has achieved very high transfer rate and accuracy. A recent study in 2014 [18] has implemented a 48-key (7Hz to 15.8Hz, with 0.2Hz resolution) online SSVEP BCI with maximum information transfer rate (ITR) of 105 bits/min and accuracy of 84.1% for 2s stimulus duration. By using additional phase modulation techniques used in stimulus and filter bank CCA decoders, the team pushed the ITR of SSVEP BCI beyond 200 bits/min. [23], [22]

SSVEP stimulus could be be ported into virtual reality (VR) scenes as part of the head up display (HUD) where it serves as a visually selectable button. For instance, a research group created a 3-target VR SSVEP BCI in 2018 to control a virtual vacuum cleaner on a 2-D grid plane to seep virtual dust. The three SSVEP stimulus were part of the HUD overlay and were used to give commands to steer left/right and move forward. [28] They later extended the same BCI setup to remotely control a real-world robotic car. [30] Though the processing time for each command is 3.95 seconds, the overall experience can be more engaging than typing letters.

Another use case is to use four SSVEP stimulus blocks in the HUD for navigation. A study in 2015 [25] proposed a maze game where the user needs to control a ball to navigate through a 2D grid based maze. Unlike flat display SSVEP typing BCIs, VR-based SSVEP BCIs are conservative on the number of stimulus used, as mutiple SSVEP stimulus sources can take a lot of screen space and become obtrusive to the VR scene. VR scene's colour and texture could also have a negative impact on the SSVEP signal quality.

3.2.2 P300 VR BCIs

Not many cases of P300 based VR BCIs can be found, as a matrix type visual stimulus interface can be difficult to design for general tasks. One P300 VR BCI was developed as a gaming approach for treating ADHD in children, where a matrix of toys displayed in a virtual class room for selection. [24] A more recent published research created a smart home simulation

based on P300, where a matrix type visual stimulus interface acts as the switch of the many home appliances used. [32]
For most general use cases, P300 based VR BCIs can be replaced by SSVEP based VR BCIs.

3.2.3 Motor Imagery (MI) VR BCIs

Motor imagery is also a popular method for VR BCI, where the event-related desynchronisation/synchronisation (ERD/ERS) during imagination of movements are used as the control signals. MI does not require any on-screen visual stimulus to take up space and this is a great advantage. It has be proven in researches that only three electrodes (FCz,C3,C4) are enough to construct a MI BCI to distinguish between right hand/left hand MI with very basic threshold techniques (power difference between C3 and C4 using FCz as the reference) while achieving $> 80\%$ accuracy. [14]

Using similar setups, a research group created a MI VR BCI where the subject could use left/right hand MI to control the steering of a rowing boat.[31]

However, to get beyond two choices is often very difficult and MI BCI is often limited by the number of choices and the accuracy is lower than that of SSVEP.[17] Also, some target users could not generate classifiable EEG signals in MI, and this proportion is considered much larger than that for SSVEP. [16], [21]

3.2.4 Hybrid BCIs

It is possible to record and analyse two types signal on parallel and create a hybrid BCI, where one control signal could be used as a state machine to increase the number of choices and to decrease the false positive rate. A research on SSVEP and MI hybrid BCI found an overall accuracy increase over SSVEP or MI alone, at the cost of using very long processing window of 8s (as MI often needs longer time for accuracy). [19]

It is a viable approach to take in the need of freeing screen space occupied by SSVEP stimulus.

3.3 Summary

Given the low signal quality of EEG recording method, VR BCIs are still limited to very discrete tasks applications.
SSVEP signals is desired to be the main signal for VR BCI given its high accuracy and information transfer rate. Besides accuracy and information transfer rate, the long-term comfort of the stimuli is also important. The number and size of SSVEP stimuli should be constrained for a more open field of view.
Motor imagery signals could also be used as additional inputs alongside SSVEP in a hybrid BCI setting to save screen space if the sacrifice in timing/accuracy performance is acceptable.

Chapter 4

Hardware and Software Setup

4.1 Chapter Introduction

This chapter provides a complete list of hardware and software components and settings used for the VR BCI project. specifications of components are also listed in the main body where necessary or suppled in the appendix.
The major components of the hardware include:

- A high-end gaming laptop used for signal processing and running VR applications.
- An Oculus Rift VR headset.
- g.tech g.USBamp-RESAERCH biosignal amplifier of 16 channels.
- g.tech g.GAMMAcap EEG cap and g.SAHARA active dry electrode system.

The major software components include:

- Unity game engine for VR application development
- Matlab and Simulink R2015a compatible with gTech hardware for real time signal recording and decoding

4.2 Hardware

1. Laptop used in this project has the following specifications:
    - Model: Gigabyte Aero 15X (2018)
    - Gigabyte Aero 15X (2018)
    - CPU: 2.2GHz Intel Core i7-8750H (hexa-core, 9MB cache)
    - Onboard Graphics: Intel UHD Graphics 630
    - Dedicated Graphics: Nvidia GeForce GTX 1070 (Max-Q, 8GB GDDR5 RAM)
    - RAM: 32GB DDR4 (2,666MHz, 16GB x 2)
    - Screen: 15.6-inch Full HD 144Hz IPS Panel (1,920 x 1,080) IPS LCD
2. VR headset used in this project:
    - Model: Oculus Rifts (CV1)
    - Display: PenTile[6] OLED 2160x1200 (1080x1200 per eye) @ 90 Hz
    - Sound: Integrated 3D audio headphones (user removable/exchangeable) (Not used in this project)

- Input: 6DOF (3-axis rotational tracking + 3-axis positional tracking) through USB-connected IR LED sensor
- Controller input: Oculus Touch motion tracked controllers, Xbox One controllers.
- Connectivity: HDMI 1.3, USB 3.0, USB 2.0

3. Bio-signal amplifier and acquisition/processing system):

- Mode: g.tech's g.USBamp-RESEARCH
- 16 DC-coupled Wide-range Input Channels Per Unit
- 24-bit Resolution with Simultaneous Sampling of all channels with up to 38.4kHz, digital signal filtering ad pre-processing
- Connects via USB 2.0
- Internal digital bandpass and notch filters, built-in calibration unit and impedance checking.
- Easy configuration and setup via the software, high-speed online data processing for SIMULINK and for LabVIEW supported
- CE-certified and FDA cleared medical device, safety class: II, conformity class:IIa, type of applied part: CF

4.3 Environment

All experiments were conducted in UoM & Fourier Robotics Laboratory. The laboratory is not a dedicated library for testing BCI systems. Many BCI systems were tested in sound-proof rooms or sometimes even electromagnetically shielded rooms to minimise the environmental background noise or the noise induced by the background noise in the EEG signals. UoM & Fourier Robotics Laboratory is a busy lab with many concurrent working project groups, and the large background noise in this environment could greatly penalise the performance of BCI systems. But it also provided the project group a chance to assess the practicality of BCI systems.

4.4 Software

4.4.1 Unity

The software that is used in this project to create the Graphical User Interface (GUI) is Unity3D. Unity3D is one of the most popular game development platform to date and it is certainly not unwarranted. A quick search on the internet would reveal its wide range of features and toolkits. It also easy to use and supports a seamless integration with the VR headset that is used in this project. The native programming language, C# is a very popular and powerful language. Hence, Unity3D is well suited to handle the GUI for this project.

User Datagram Protocol UDP is used to establish a local connection and send messages between Simulink and Unity3D. UDP is suitable for this project's application because it is quick and efficient.
The following are the software (and its version) used in this project:

- Unity Version: 2019.1.10f1

- NVIDIA GeForce Driver Version: 26.21.14.3111

4.4.2 MATLAB and Simulink R2015a

MATLAB was used as an analysis tool for the analysis of collected EEG signals and offline decoders were written to test for performances.
Simulink was used to collect the EEG signals from the bio-signal amplifier in real time. Decoders written in MATLAB were ported over to perform real time decoding and the results were translated into defined UDP packets and sent to the Unity program as commands.

Figure 4.1: Lab Environment and Setup Overview

Figure 4.2: VR headset and EEG cap side view

Chapter 5

General EEG setup and Methodology

5.1 Chapter Introduction

This chapter describes the main EEG signal acquisition settings used for all the experiments conducted during the project, as well as the general methods used for SSVEP and MI BCI including stimulus types and decoding methods. The two experiments of the project have slight variations in settings but all main settings are described in this chapter for detailed reference.

5.2 Electrode positioning

A total number of 15 electrodes were used for the EEG recording. 9 electrodes were placed across the occipital lobe region (Pz,PO7,PO3,POZ,PO4,PO8,O1,OZ,O2) for SSVEP signal recording. The other 6 electrodes were placed along the primary motor cortex (FCZ, C3,C1,Cz,C2,C4) to capture motor rhythms induced by actual or imagined actions. A1 and A2 were used respectively as the ground and reference and no bipolar setting was used. The positioning follows the International 10-20 System defined in ([8])

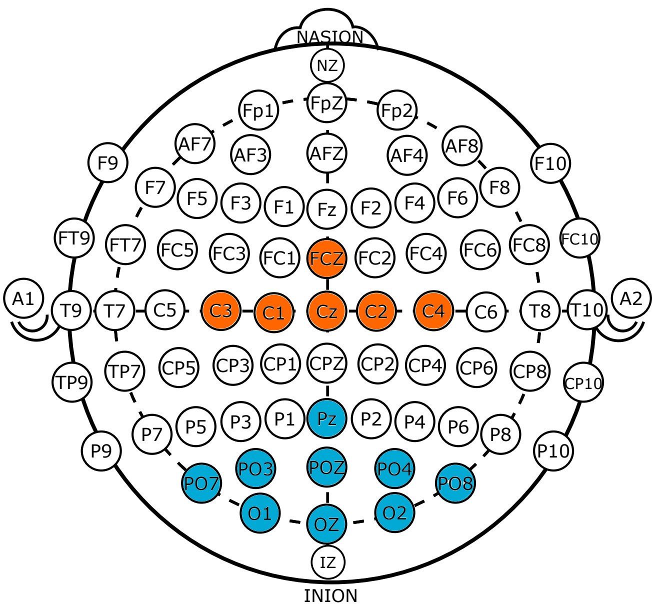


Figure 5.1: Electrodes Positioning

5.3 Amplifier recording settings

The g.USBamp-RESEARCH was configured to record at a sampling frequency of 512Hz. All channels were configured to have 0 bipolar setting, a band pass filter of 5-60Hz and and a notch filter at 50Hz. All 15 channels were selected to have a common ground and reference.

5.4 General VR BCI Prompts

A green timing bar is filled duration the duration of a processing window and a grey timing bar in the same position is emptied during the resting window. The timing bar is a part of the head up display (HUD) and rotates with the head orientation.

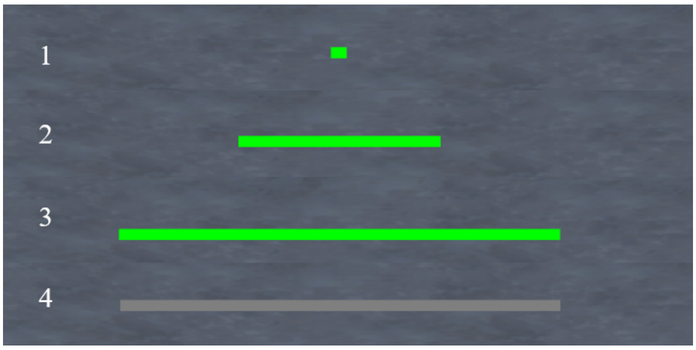


Figure 5.2: BCI timing bar

5.5 SSVEP Stimuli

Square wave SSVEP stimuli is achieved by alternating a sprite between a lighter color and a darker color, where the frequency is specified by the delay between each alternating step. Studies have shown that using generating functions dependent on the refresh rate could achieve better frequency resolution, where timer based delays could be subject to program timer error. ([15]) However, the VR-EEG setup is computationally intense and the frame rate can become unstable from time to time and thus the more basic timing method was applied. Since the lab environment is noisy and the limited SSVEP stimuli targets does not require small frequency resolutions, this approach was justified by the group.

$$f(x) = A * sgn[\sin(2\pi f(t + PD))] + B, PD = \frac{k-1}{4f_k} \tag{5.1}$$

where A refers to the change in color intensity, B refers to the mid point color intensity between each peak and valley, N is the number of stimuli. PD represents the phase modulation delay between the SSVEP stimuli targets (k refers to the index of the stimuli based on ascending frequency order), a technique proposed by ([23]) to increase SSVEP SNR.

5.6 Standard Canonical Correlation Analysis (CCA)

Canonically Correlation Analysis (CCA) has been widely used to decode SSVEP signals due to its easy implementation and robustness. ([12], [13], [17]) CCA is a method of measuring the

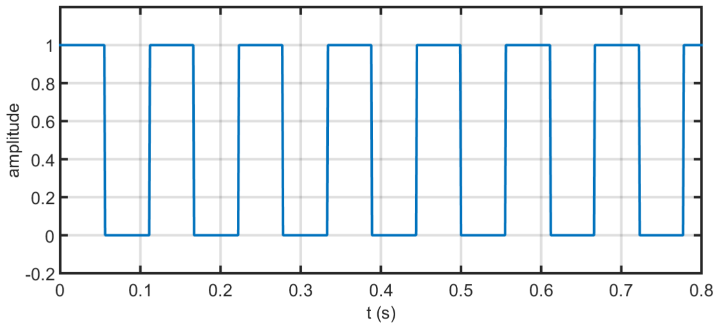


Figure 5.3: Sample Square Wave at 9Hz

linear relationship between two multidimensional variables developed by H. Hotelling. ([1]) Consider two multi-dimensional variables X, Y and their linear combinations $x = X^T W_X$ and $y = Y^T W_Y$. CCA finds the two bases W_X, W_Y for each variable that are optimal with respect to correlations and computes the corresponding correlation factor ρ. Thus the problem is to maximise ρ in the following equation:

$$\rho = \frac{E[XY]}{\sqrt{E[X^2]E[Y^2]}} = \frac{E[W_X^T XY^T W_Y]}{\sqrt{E[W_Y^T XX^T W_X]E[W^T YY^T W_Y]}} \tag{5.2}$$

The maximum of ρ is the maximum canonical correlation factor.
MATLAB function canoncorr could be used to compute the answer of such a problem. For SSVEP frequency detection applications, X represents the multiple channel SSVEP signals collected during a timing window (MXN matrix with M channels and N data samples), and Y represents the reference signals of equal data sample length N. For this project, $M = 9$ as 9 electrodes were placed around the occipital lobe region to capture SSVEP, as described in 4.2.
Sinusoidal signals are commonly used as the reference signals for SSVEP detection ([13]) and has form:

$$Y_{ref} = \begin{bmatrix} \sin(2\pi \cdot ft) \\ \cos((2\pi \cdot ft)) \\ \sin(2\pi \cdot 2ft) \\ \cos(2\pi \cdot 2ft) \\ \vdots \\ \sin(2\pi \cdot N_h ft) \\ \cos(2\pi \cdot N_h ft) \end{bmatrix} \tag{5.3}$$

where f is the stimuli frequency and N_h is the number of harmonics to be included in the reference signals. Y_{ref} has dimension $2N_h XN$ compared to X. For SSVEP frequency recognition, a reference signal is generated for each stimuli frequencies used. The canonical correlation factor ρ is then computed between the SSVEP signals and each reference signal, where the reference signal that generates the largest ρ indicates the stimuli the subject is attending to.

5.7 Filter Bank Canonical Correlation Analysis (FBCCA)

(Chen et al 2015[22]) proposed a filter bank canonical correlation analysis (FBCCA) method to improve the decoding performance by filtering SSVEP signals into N_B number of sub-bands and a correlation factor between each sub-band and each reference signal is computed and combined into a overall weighting.

5.7.1 Generating sub-bands from SSVEP raw signals;

Digital band pass filers are used to separate the raw signals into N_b sub-bands. For offline Matlab analysis in Experiment 1, band filtering was done by filtering the signals with sixth order zero-phase Chebyshev type I infinite impulse response (IIR) filters using the filtfilt() function. For online Simulink decoding implemented for Experiment 2, the filtering was done by using the channel bandpass butterworth filter block of the g.Tec g.RT analyze library, which is configured to implement fifth order Butterworth bandpass filters. Figure (5.7) provides a abstracted flow chart for using FBCCA analysis.
(Chen et al 2015[22]) proposed three main methods for generating sub-bands.

- Method 1 divides the full frequency band of SSVEP signals into sub-bands with equally spaced bandwidths (usually the bandwidth of stimuli frequencies used).
- Method 2 divides the full frequency band into individual harmonic frequency bands, $Band_K = [K \cdot Stimuli\_band, min(2K \cdot Stimuli\_band, SSVEP\_band\_upper\_bound)$.
- Method 3 generates sub-bands covering multiple harmonic frequency bands with a high cutoff frequency at the upper-bound of the SSVEP full frequency band.

Figure (5.4,5.5,5.6) provides a visualisation of the three methods for the case where the full frequency band is 9-60Hz and the stimuli frequency band is 9-15Hz. (which is the setting for Experiment 2 of this project).

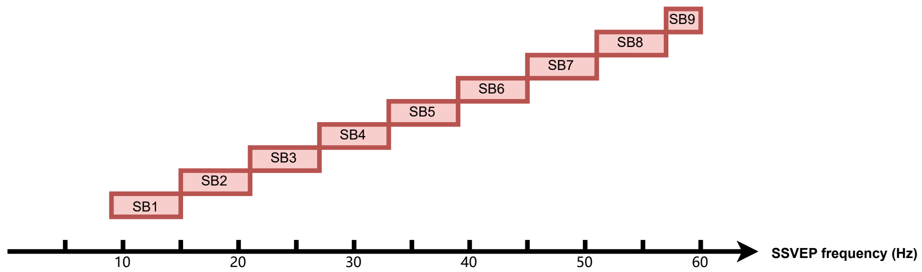


Figure 5.4: Filter Bank Method 1

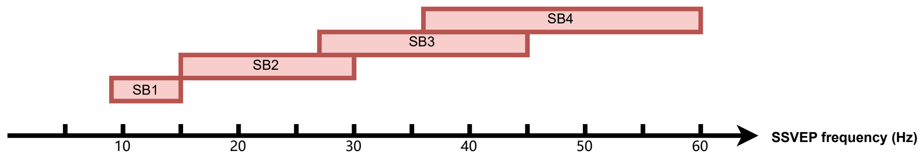


Figure 5.5: Filter Bank Method 2

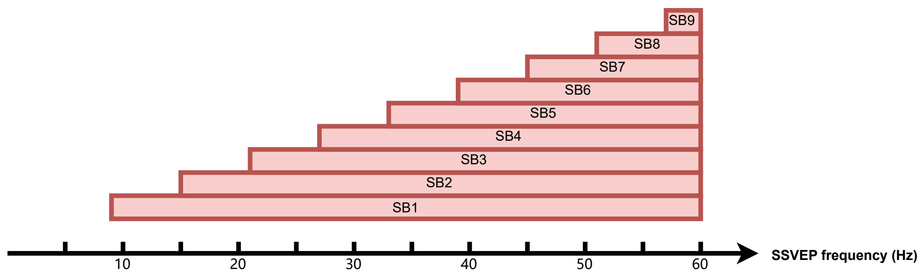


Figure 5.6: Filter Bank Method 3

(Chen et al 2015[22]) concluded that Method 3 achieved the best performance for SSVEP frequency identification. Thus Method 3 would be chosen for the offline analysis and online decoding of this project.

5.7.2 Combined score using N correlation factor generated by N CCA results for N sub-bands

Consider a total of V stimuli frequencies used.
For each stimuli frequency f_K, FBCCA generates N CCA correlation factors between each of the N sub-bands and the corresponding reference signal Y_K.

$$\rho_K = \begin{bmatrix} \rho_K^{SB1} \\ \rho_K^{SB2} \\ \vdots \\ \rho_K^{SBN} \end{bmatrix} = \begin{bmatrix} \rho(X_{SB1}^T W_{X_{SB1}}(X_{SB1}Y_{f_K}), Y_K^T W_{Y_K}(X_{SB1}Y_K)) \\ \rho(X_{SB2}^T W_{X_{SB2}}(X_{SB2}Y_{f_K}), Y_K^T W_{Y_K}(X_{SB2}Y_K)) \\ \vdots \\ \rho(X_{SBN}^T W_{X_{SBN}}(X_{SBN}Y_{f_K}), Y_K^T W_{Y_K}(X_{SBN}Y_K)) \end{bmatrix}_{(NX1)} \tag{5.4}$$

A weighted sum of squares of the correlation factors between all sub-bands (ρ_K^{SB1} to ρ_K^{SBN}) can be calculated to produce a combined correlation factor score $\hat{\rho}_K$ for each reference signal Y_K:

$$\hat{\rho}_K = \sum_{n=1}^{N} w(n) \cdot (\rho_K^n)^2 \tag{5.5}$$

where n is the index of the sub-band starting at 1. The peak magnitude in SSVEP harmonics decreases as the response increases, the weighting function $w(n)$ is defined to give higher weighting on the fundamental and lower-order harmonics:

$$w(n) = n^{-a} + b, n \in [1, N] \tag{5.6}$$

The values of a and b are to be tuned to maximize the classification performance.

5.7.3 Decoding FBCCA results

For V number of stimuli frequencies, FBCCA produces an array of $\hat{\rho}_K$ corresponding to the index of each stimuli frequency. The index of the highest value of $\hat{\rho}$ indicates the stimuli frequency the subject is attending to. This can be implemented using the max() function in MATLAB to give the maximum value and the corresponding index of an array.

$$(\hat{\rho}_{max}, decoded\_Stimulus\_index = max([\hat{\rho}_1, \hat{\rho}_2, \ldots, \hat{\rho}_V]) \tag{5.7}$$

A minimum threshold for $\hat{\rho}_{max}$ could be set to create a null target to indicate when the subject is not attending to any of the frequency stimuli.

5.7.4 SSVEP Signal Noise Ratio (SNR)

As described earlier in this chapter and also in the literature review, the fundamental principle of applying SSVEP decoders is to observe the peak amplitude at the stimuli fundamental frequencies and subsequent harmonics in the SSVEP signals. (([12], [13], [17])). This could be visualised in the FFT plot of the SSVEP signals by looking at the peaks at the fundamental frequency, second harmonic, third harmonic and so on. The less the background noise, the more harmonic peaks could be observed. ([17]) The SNR of a SSVEP stimulus frequency target could be estimated by calculating the ratio between the peak magnitude (of the fundamental frequency, first harmonic,...etc.) and the average magnitude in the FFT:

$$SNR(K) = 20\log_{10}(\frac{Peak(k^{th})}{Mean(FFT)}) \tag{5.8}$$

where K is the index of the harmonic.

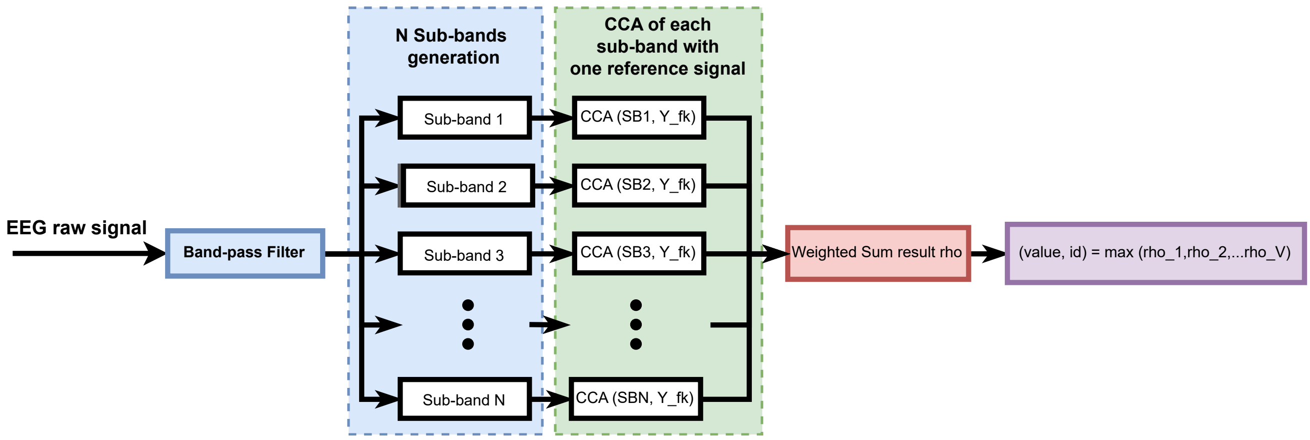


Figure 5.7: FBCCA analysis flowchart

5.8 Motor Imagery: Decoding left/right hand tasks

(Y.Wang,2017 [14]) proposed a MI-based two-target BCI using only three primary electrodes and the decoder requires no sophisticated training. The study proposed a novel method to differentiate left hand/right hand motor imagery tasks by assessing the β (12-30Hz) band power difference between C3 and C4 using FCz as the reference, where a negative difference indicates a left hand movement/imagery task and a positive difference indicates a right hand movement/imagery task. Each task was done during a 8 seconds window and can reach up to 88% accuracy. This simple two-target decoder can be expressed as the following equation:

$$target = sign(P_\beta(C3 - FCz) - P_\beta(C4 - FCz)) \tag{5.9}$$

where P_β stands for the β band signal power, and a negative result indicates a right hand movement/imagery task and a postive result indicates a left hand movement/imagery.

5.8.1 Welch's method to estimate band power

Welch's method or the periodogram method is a way to estimate a signal's power spectrum by dividing the time signal into successive blocks, forming the periodogram for each block, and averaging. ([2]) ([29]) provided an brief mathematical explanation of Welch's method. Consider m th windowed, zero-padded frame from the signal x by

$$x_m(n) = w(n)x(n + mR), \quad n = 0, 1, \ldots, M - 1, m = 0, 1, \ldots, K - 1 \tag{5.10}$$

where R is defined as window hop size and K denote the number of available frames. Then the periodogram of the m th block is given by:

$$P_{xm,M((\omega_k)} = \frac{1}{M}|FFT_{N,k}(X_m)^2| \tag{5.11}$$

Thus the Welch estimate of the power spectral density is given by

$$\hat{S}_x^W(\omega_k) = \frac{1}{K}\sum_{m=0}^{K-1} P_{xm,M((\omega_k)} \tag{5.12}$$

The MATLAB function pwelch() was used for estimating the EEG channel power spectrum. Window was set to capture 256 samples with 128 samples overlapping and NFFT was set to 512

such that 1Hz resolution is achieved for 512Hz sampling rate.
Then MATLAB function cumtrapz() was used to estimate the β signal band power by using trapezoidal numeric integration from 12 to 30Hz given by the power spectrum.

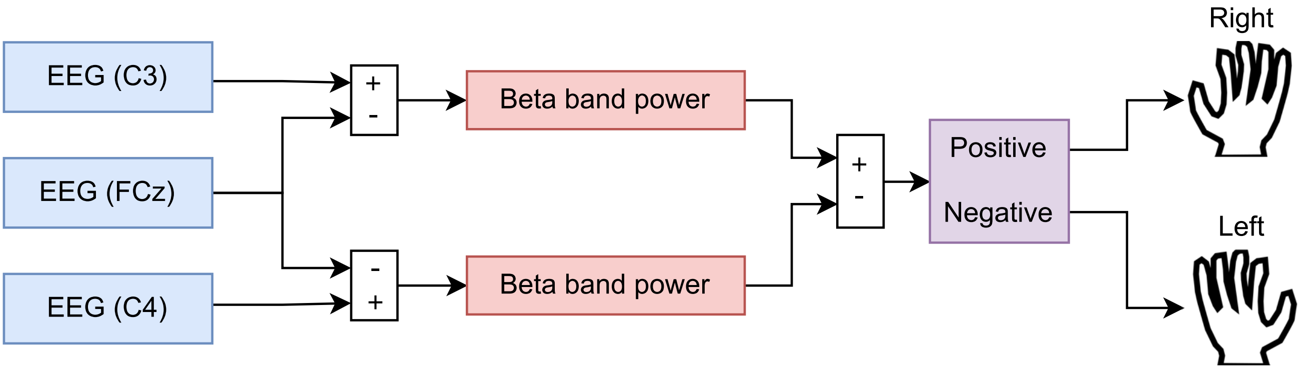


Figure 5.8: Binary left/right hand task MI decoding flowchart

5.9 Information Transfer Rate of Multi-Class BCI systems

The information transfer rate of a BCI system in bits/minute can be calculated using the following formula ([9]):

$$ITR = \frac{60}{T}(\log_2(C) + P\log_2 P + (1-P)\log_2 \frac{1-P}{C-1}) \tag{5.13}$$

where P is the classification accuracy, C is the number of classes and T is the processing time window in seconds.

5.10 Summary

This chapter provides a detailed explanation of the EEG setup used for this project and the decoding methodology of the SSVEP and motor imagery/action signals.

Chapter 6

Experiment 1: Data collection and Performance Verification of SSVEP and MI

6.1 Chapter Introduction

This chapter explains the first experiments conducted to collect and analyse data from both SSVEP and Motor Imagery brain signals. Then the performance of the decoder is validated and compared. The results were used to design possible control scheme in Experiment 2 to control a virtual robot to pick up objects.

6.2 Aim

This experiment comprised of two sub parts:

- Test and verify the performance of SSVEP with 4 stimulus (7,9,11,13Hz) in a VR setting, using three different types of block sprite stimulus (flickering, size changing, alternating chess-grid pattern) on the three group members.
- Test and very the performance of binary decoder for motor action/imagery on the three members, using both actual movements and imagined movements.
- Conduct offline analysis for performance evaluation
- Select one of the three subjects that is most suitable for the experiment based on his performance

6.3 Methods

6.3.1 SSVEP Stimuli Types

Three types of SSVEP stimuli were designed for performance and user experience comparison. Each stimuli source was implemented based on the same generating function (5.3) in the Unity VR application run on the Oculus Rift.

1. **Blinking box:** A commonly used stimuli type for high speed SSVEP BCI, where the stimuli shifts from black to white and then back, giving it a blinking visual experience. The sequence is illustrated in Figure (6.1).

2. **Resizing box:** A spatial pattern type stimuli, where the stimuli alternates between two sizes (small and big) its size at the set frequency.The sequence is illustrated in Figure (6.2).
3. **Chess-grid box:** A stimuli using both spatial and light intensity patterns, it could be regarded as a combination of multiple blinking boxes, but using 50% light density and is less irritating to look at. The sequence is illustrated in Figure (6.3).

Figure 6.1: Blinking Box SSVEP stimuli

Figure 6.2: Resizing Box SSVEP stimuli

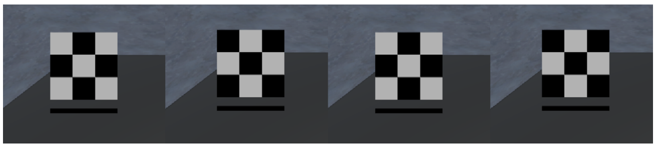


Figure 6.3: Chess-grid Box SSVEP stimuli

6.3.2 SSVEP Experiment Procedure

Interface

The purpose of this experiment is to collect brain signal data for each stimuli and analyse it offline. The four stimuli would produce four different frequencies of square signal (9, 11, 13 and 15Hz). The four stimuli targets are set up as shown in (figure 6.4).

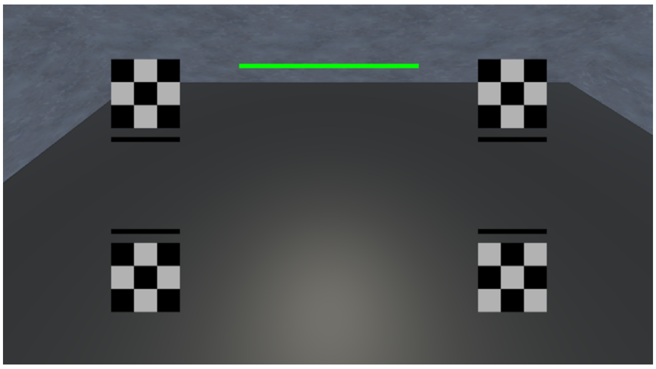
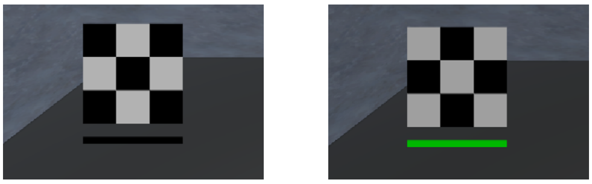


Figure 6.4: SSVEP Experiment VR Interface Setup

Procedure

To match the data collected with the corresponding frequencies, a set of instructions would be given to the subject on which box to focus on. This is done through the indicator located above or below each box. When an indicator turns from off (figure 6.5a) to on (figure 6.5b), the subject would turn his attention to the box associated with the indicator. If no indicator lights up, that means subject should not focus on any box. One experiment consists of 15 sets of five prompts (0-4) which corresponds to no box, box 1, box 2, box 3 and box 4, respectively. The sequence of prompts is shuffled per set, which means that every set contains one of each prompts (0-5) and there is exactly 15 of each command in the whole sequence. The sequence is generated in MATLAB using function randomperm().



(a) Indicator Off (b) Indicator On

Figure 6.5: SSVEP Prompt

Each prompt lasts 5 seconds in length and there is a 1 second delay between two consecutive prompts. Each prompt is given for 15 times for each stimuli (including the null target) and the whole experiment contains 75 prompts.

6.3.3 Motor/ Motor Imagery Experiment Procedure

Procedure

Similarly, to match the data collected with the associated body part, the subject would receive instructions on what to do from the prompts shown on the HUD. The prompts take in the form of a left and right hand model as shown in figure (6.6). The set of three prompts used in this experiment for motor and motor imagery are shown in figure 6.7. The prompt indicated by number one corresponds to both hands at rest. The second one corresponds to (imagine) squeezing left hand. And lastly (imagine) squeezing right hand.

Figure 6.6: Motor Experiment Setup

Figure 6.7: Motor Prompts

Each prompt lasts 8 seconds in length and there is a 1 second delay between two consecutive prompts. Each prompt is given for 15 times for each command (including the null target) and the whole experiment contains 45 prompts.

6.4 Data Analysis

The EEG signal raw data (still filtered by the amplifier settings for 5-60Hz, as described in the previous section) were recorded from Simulink as a matrix (16 rows where the first row is the time and the other 15 corresponds to the 15 channels used, see (Figure 5.1)).

6.4.1 SSVEP Analysis

Filter bank canonical correlation analysis (FBCCA) ([22]) described in (5.7) (using channel Pz,PO7,PO3,POZ,PO4,PO8,O1,OZ,O2) was used to decode the SSVEP signals, using Method 3 (5.7.1) for the filter bank generation.

Set parameters

1. **Filter Bank:** The used stimuli frequencies are 9Hz, 11Hz, 13Hz, 15Hz, covering a bandwidth of 6Hz, and the upper bound of SSVEP bandwidth is 60Hz.This results in 9 filter banks, respectively (9-60Hz, 15-60Hz. 21-60Hz, 27-60Hz, 33-60Hz, 39-60Hz, 45-60Hz, 51-60Hz, 57-60Hz, (Figure 5.6)). Note that the first filer bank was set to (5-60Hz) to avoid losing information due to low pass.

2. **Weighted Function w:** The weighting function w (Equation 5.6) for generating the FBCCA correlation factor (the weighted sum of squares in Equation 5.5) has two constants a and b. They were tuned by trial-and-error approach during the offline analysis to yield the optimal results. The values were set as $a = 1.25$ and $b = 0.5$.

3. **Processing Delay:** A 150ms delay was used to compensate for the reaction time between the stimuli prompt time window and the corresponding FBCCA processing time window.

4. **H, number of harmonics in the reference signals (5.3):** The number of harmonics included in the reference signals. It was set as 2 as only a maximum of 2 harmonic peaks could be observed in the FFT, and was presented in the results section and (Figure 6.10).

varied parameters for performance comparison

1. **Processing Window Length:** The processing time window was used to analyse the performance of the decoder against the processing time window, where it was set to 0.5s to 5s with 0.5s step.

2. **Threshold for identifying SSVEP null target:** The threshold for the FBCCA maximum correlation factor for all reference signals to detect the SSVEP null target.

Performance Metric

1. **Accuracy:** The accuracy for differentiating the four frequency stimuli
2. **Information Transfer Rate (ITR):** ITR of multi-target BCI is defined in (5.13).
3. **Sensor Noise Ratio (SNR):** To estimate the SNR of the SSVEP signals harmonics, the Fast Fourier Transform (FFT) data for each prompt for each stimulus frequency was first collected and averaged, then the following definition of SNR (5.8) was used to estimate the SNR for the fundamental frequency and all observable subsequent harmonic peaks in the FFT plot.

6.4.2 Motor Imagery Analysis

Welch's method for power spectrum density was done using MATLAB function pwelch() and β(12-30Hz) band power was done by trapezoidal numerical integration using MATLAB function cumtrapz() for computing β(12-30Hz) band power. The band power difference between C3 and C4 using FCz as the reference was used to differentiate left/right hand motor imagery tasks. The methodology was fully described in (5.8).

varied parameters for performance comparison

1. **Processing Window Length:** The processing time window was used to analyse the performance of the decoder against the processing time window, where it was set to 1s to 8s with 1s step.
2. **Threshold for differentiating the two targets:** The threshold of P(C3)-P(C4) to differentiate left/right hand tasks, ideally 0, was to be investigated. Also the value for stationary task (no movement/imagery) was to be investigated.

Performance Metric

1. **Accuracy:** The accuracy for differentiating left/right hand tasks.
2. **Information Transfer Rate (ITR):** ITR of multi-target BCI is defined in (5.13).

6.5 Results and Discussion

6.5.1 SSVEP Results (neglecting null target)

Both the results of using CCA (using band-pass filter of 5-20Hz) and the results of using FBCCA were provided below for comparison. The null-target threshold could not effectively be identified as the FBCCA correlation factor drops slowly after subject stops attending to the stimuli.

1. **CCA accuracy versus processing window length:**
(Table 6.1,6.2,6.3)
2. **FBCCA accuracy versus processing window length:**
(Table 6.4,6.5,6.6)

	Blink Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.1833	0.3833	0.3667	0.4167	0.5167	0.5833	0.6000	0.6833	0.7333	0.7333
Subject 2 Accuracy	0.3000	0.2667	0.3833	0.4667	0.4333	0.4667	0.4833	0.4500	0.5333	0.5667
Subject 3 Accuracy	0.2833	0.3833	0.5500	0.6500	0.7333	0.8833	0.9500	0.9667	0.9833	1.0000

Table 6.1: Blink Stimuli CCA decoding accuracy versus time

	Resizing Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.2500	0.3000	0.3333	0.3000	0.2833	0.3333	0.3000	0.3500	0.3333	0.3000
Subject 2 Accuracy	0.3167	0.2667	0.3500	0.3167	0.3000	0.3500	0.3333	0.3500	0.3333	0.3333
Subject 3 Accuracy	0.2833	0.3333	0.3500	0.3833	0.3167	0.3667	0.4167	0.4833	0.4500	0.5167

Table 6.2: Resizing Stimuli CCA decoding accuracy versus time

	Resizing Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.2833	0.2000	0.3167	0.2667	0.4000	0.3833	0.3667	0.4333	0.4500	0.4500
Subject 2 Accuracy	0.2667	0.2333	0.2500	0.2667	0.3167	0.3167	0.2833	0.3167	0.3500	0.3500
Subject 3 Accuracy	0.3333	0.3167	0.4167	0.5000	0.5500	0.6333	0.7500	0.8000	0.8167	0.8833

Table 6.3: Chess-grid Stimuli CCA decoding accuracy versus time

	Blinking Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.2667	0.2667	0.3167	0.3667	0.5000	0.6333	0.6333	0.7167	0.7000	0.7500
Subject 2 Accuracy	0.2833	0.2167	0.3333	0.4333	0.5167	0.5167	0.5500	0.5667	0.6667	0.7167
Subject 3 Accuracy	0.2333	0.4833	0.7000	0.8500	0.9167	0.9667	1.0000	1.0000	1.0000	1.0000

Table 6.4: Blink Stimuli FBCCA decoding accuracy versus time

	Resizing Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.2833	0.2500	0.3000	0.2833	0.3333	0.3167	0.2667	0.3167	0.3667	0.3667
Subject 2 Accuracy	0.2833	0.3167	0.2667	0.3000	0.3000	0.2833	0.2667	0.3167	0.2833	0.3500
Subject 3 Accuracy	0.2833	0.3500	0.3500	0.3833	0.4667	0.5500	0.6000	0.6833	0.6833	0.7000

Table 6.5: Resizing Stimuli FBCCA decoding accuracy versus time

	Chess-grid Stimuli Processing Time Window (seconds)									
	0.5s	**1.0s**	**1.5s**	**2.0s**	**2.5s**	**3.0s**	**3.5s**	**4.0s**	**4.5s**	**5.0s**
Subject 1 Accuracy	0.2333	0.1833	0.1833	0.2500	0.4333	0.4500	0.4500	0.5500	0.5167	0.6167
Subject 2 Accuracy	0.3500	0.3000	0.3000	0.3167	0.3833	0.4500	0.3833	0.4667	0.4833	0.4333
Subject 3 Accuracy	0.2667	0.3333	0.6667	0.7167	0.8500	0.8667	0.8667	0.9000	0.9333	0.9500

Table 6.6: Chess-grid Stimuli FBCCA decoding accuracy versus time

Three major points could be concluded from the results in (Table 6.2,6.1,6.1,6.4,6.5,6.6).

1. Subject 3 has achieved superior performance over the other two group members using all three types of SSVEP stimuli.

2. FBCCA yields higher classification accuracy over standard CCA for three test subjects for each of the three stimuli. (Figure 6.8) visualises the case for the chess-grid stimuli for Subject 3. Maximum ITR of occurs at 2.5s processing window, with 0.85 accuracy.

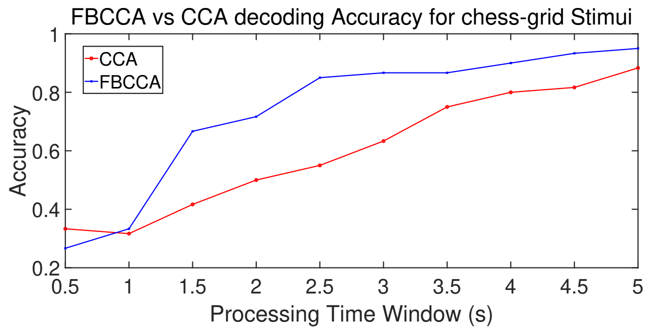


Figure 6.8: CCA versus FBCCA accuracy comparison for chess-grid Stimuli (Subject 3)

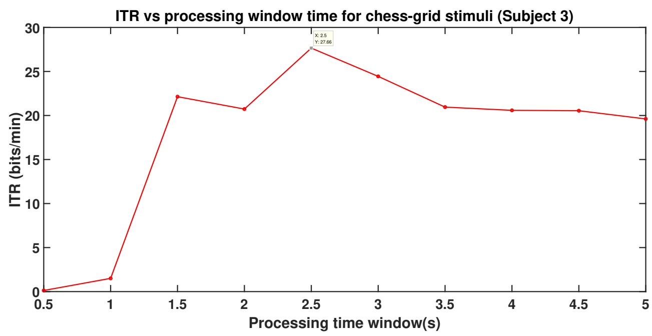


Figure 6.9: ITR vs processing window time for chess-grid stimuli (Subject 3) (Subject 3)

3. The Blinking stimuli yields superior performance over the other two stimuli, and the resizing stimuli is the least effective. The chess-grid stimuli has slightly less accuracy at all processing time length compared to the blinking stimuli. However, all three subjects (project group members) preferred chess-grid stimuli over the blinking stimuli due to relative better visual comfort, where the blinking stimuli was considered irritating to look at over several minutes of test.

SNR for each stimuli of the best performing subject (Subject 3)

The SNR for the four stimuli for each of the Subject 3 were given below for up to the second harmonic. Signal was filtered to range (5-60Hz). However, the background noise in the testing laboratory is generally large, no more than 2 harmonic peaks could be observed in the FFT of the 5 seconds processing window.
Consider the case for the 15Hz blinking stimuli SSVEP FFT average for Subject 3 for a general noise level in (Figure 6.10).

	SNR(9Hz)		SNR(11Hz)		SNR(13Hz)		SNR(15Hz)	
	H1	**H2**	**H1**	**H2**	**H1**	**H2**	**H1**	**H2**
Blink	6.91dB	3.82dB	8.41dB	4.237dB	10.00dB	3.904dB	7.83dB	5.03dB
Sizechanging	2.551dB	2.322dB	4.60dB	2.68dB	4.26dB	1.49dB	2.80dB	2.21dB
Chess-grid	3.28dB	1.29dB	5.60dB	2.38dB	4.26dB	1.88dB	3.91dB	2.90dB

Table 6.7: SNR of SSVEP (Subject 3) signal during 5s window

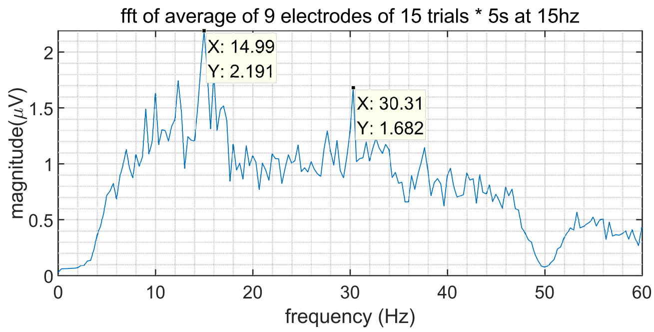


Figure 6.10: Average FFT for 15Hz blinking stimuli for Subject 3

6.5.2 Motor Imagery Result (neglecting null target)

Both subject 1 and subject 2 were considered MI illiterate as no separation in the resulting power difference across the three tasks were observed. Like the case for SSVEP, Subject 3 yielded superior results for this experiment. Using 0 as the threshold, and the data of decoder accuracy versus time was given below:

	MI processing window length (s)							
	1s	**2s**	**3s**	**4s**	**5s**	**6s**	**7s**	**8s**
Accuracy	0.7333	0.8333	0.7667	0.7667	0.8000	0.7667	0.6667	0.6333
ITR (bits/min)	9.7986	10.4970	4.3256	3.2442	3.3369	2.1628	0.7006	0.3892

Table 6.8: MI decoder performance on Subject 3

(Figure 6.11) illustrated the results for the experiment for 5 seconds processing window. The

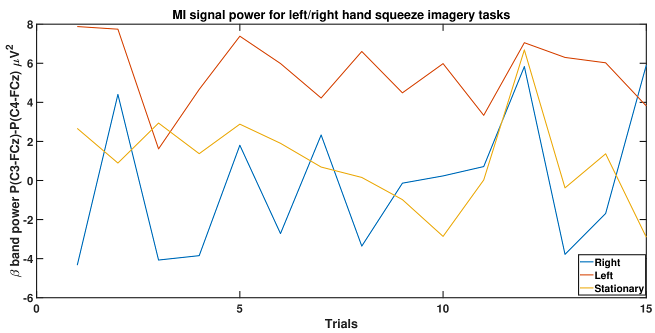


Figure 6.11: β band power P(C3-FCz)-P(C4-FCz) for left/right hand imagery (Subject 3)

high accuracy of smaller time window setting is misleading as only windows beyond 5s produces a larger separation between left hand and right hand motor imagery.
One way to decrease false positive ratio for this decoder is to use MI for a very infrequent

commands such as commanding the robot to grasp, such that we can increase the threshold to be higher for left hand imagery. Then left hand imagery could be used for a infrequent task such as commanding a robot to grasp at a given location, while neglecting the right hand imagery/stationary state.

6.6 Summary

Although only subject 3 of the three subjects (project group members) was considered relatively BCI literate, his performance using SSVEP and MI BCI was exceptionally good in the noisy lab environment, reaching up to 100% using blinking stimuli using 3.5 seconds processing window. He also achieved a high accuracy of 95% using the chess-grid stimuli using 5 seconds processing window.
He also reached up to 80% accuracy using an MI decoder to differentiate left/right hand imagery tasks using 5 seconds processing window.
For the goal of this project, a higher accuracy was considered more important than information transfer rate since commands of robots needed to be accurate in successive commands. Unlike typing tasks which could benefit from spell checking programs and tolerate 80-90% accuracy, performance of tasks using successive commands decrease exponentially with the drop in accuracy of single command.
The results and insights from experiment 1 were used to design the tasks and controlling scheme for experiment 2: To control a virtual robot to navigate on a 2-D table and pick up objects.

Chapter 7

Task Design: Robot Picking Up Ball

7.1 Chapter Introduction

This chapter discusses the design and implementation of the task which requires the subject to control a robot arm to pick up a ball using brain signal through BCI. The task setup and how the task is conducted are explained in full details.

7.2 Task Description

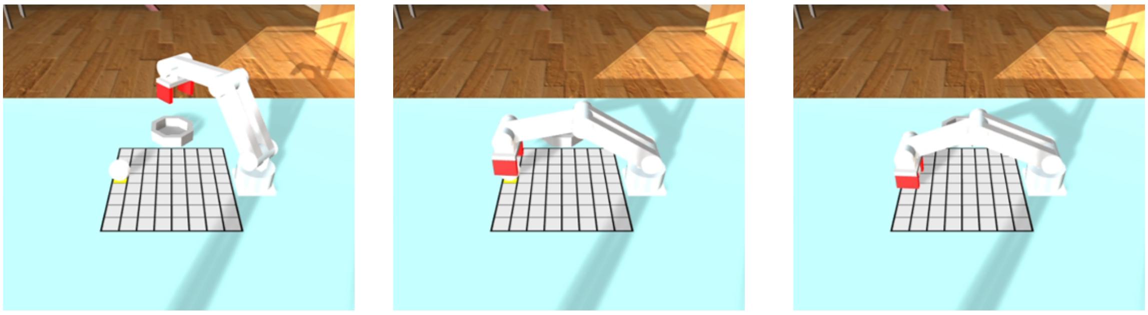
The task is to have the subject control a 'cursor' to navigate on the 'ball spawn area' and upon correctly positioning the cursor, the subject will issues command to the robot to attempt to pick up a ball at the cursor position. If ball exists, the robot will successfully pick it up and drop the it in a bucket near by. If pick up attempt was unsuccessful, the robot's end effector would still return to above the bucket at end of attempt. Given successful ball pick up and robot end effector returns to above the bucket, the ball would respawn on the ball spawn area. Respawn occurs after 6 seconds, measured from when the robot first starts moving from rest away above the bucket towards the cursor.

7.3 Task Setup

When conducting the task, the subject can see through the VR headset, the robot, the ball spawn area and the bucket on the table in the center of an enclosed room, with a layer of HUD in front, which consists of components that allows SSVEP usage and assists in successful ball pickup.

7.3.1 Robot

The robot has a 3D model made in unity, the segments length ratios are $[(segment1_{base} : segment2_{arm} : segment3_{arm} : segment4_{end\_effector}) = (1 : 2 : 2 : 1)]$, it is scaled to be fitted into the enclose room. The robot has 4 joints(refer to figure 10.1), 3 DoF, with one constraint on the end effector joint 4 to make the end effector always point down along the z axis. The robot can navigate to the center of the cursor on the Ball Spawn Area and attempt to pick up a ball. The robot's end effector starts from the starting position above the bucket and follows trajectory a to via point a above the the cursor center and lowers to end point by following trajectory b (refer to figure 7.1) . The speed of the robot's end effector through each trajectory increases by a multiplier each fixed frame in Unity, so it is not constant.



(a) Start point (b) Via point (c) End point

Figure 7.1: robot's end effector start, via and end point

7.3.2 Ball Spawn Area and Navigation System

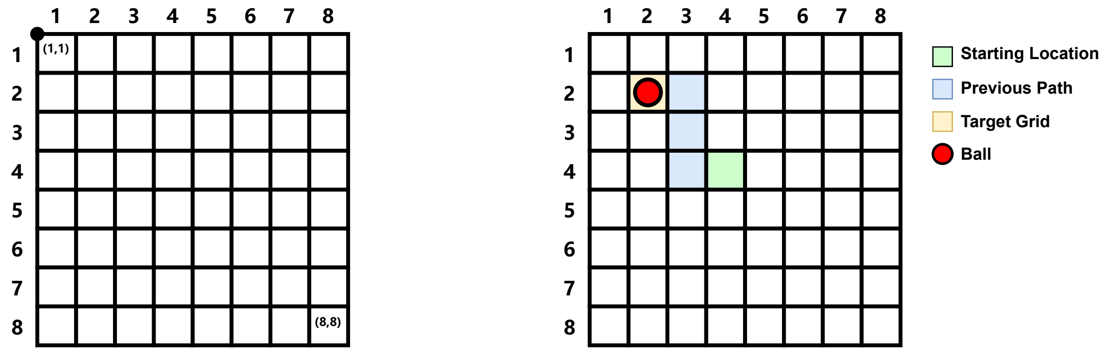
To make the tasks simpler, the ball would only spawn at one out of 64 locations on a 8 by 8 grid. It would spawn randomly across the grid with uniform distribution of 1 to 64 which corresponds to the grid number i.e. first row: 1-8, second row: 9-16, and so on. After the ball spawns, the user could direct the robot arms towards the 1 by 1 grid on which the ball lies with the use of a cursor. The cursor size, position and movement are specifically defined by a navigation system. In this report, two navigation systems (grid and quadrant) would be discussed. The motivation to use a navigation system is to make the task even simpler for the subject as it reduces the amount of commands needed to reach the target grid. This is ideal because the commands that are sent from Simulink are discrete. It also shortens the time needed to complete the tasks which is desirable.

Grid Navigation

In grid navigation system, the cursor would take in the form of a 1x1 grid from the start. The grid coordinates is specified in Fig 7.2a and the starting cursor position would be located at (4,4). This system utilizes five controls to navigate the cursor in the grid system. The controls and their UDP message equivalents are:

1. Move Up (UDP 1)
2. Move Down (UDP 2)
3. Move Left (UDP 3)
4. Move Right (UDP 4)
5. Pickup Confirmation(UDP 5)

The objective of this system is for the subject to navigate through the 8 by 8 grid towards the grid where the ball is located on using the four movement controls. Once the cursor reached the target grid, the subject would then need to confirm the selection to send a pickup command to the robot arms. As an example, Fig 7.2b shows one of the possible path the subject needs to take to reach the goal. This corresponds to these five consecutive commands: Left, Up, Up, Left, Confirm.



(a) Grid Layout (b) Grid Navigation Path Example

Figure 7.2: Grid Navigation System

Quadrant Navigation

The quadrant uses the same 8 by 8 grid system that the grid navigation system uses. However, this system uses different way to navigate the cursor. It is composed of four different stages.

1. **Stage 1**: The whole 8x8 ball spawn area would be split into four 4x4 quadrants (shown in Fig 7.2a). Once Unity received a UDP command, the corresponding 4x4 quadrant would get selected and Stage 2 commences.

2. **Stage 2**: The 4x4 selected grid from Stage 1 would be divided again into four smaller 2 by 2 quadrants. Once Unity received a UP command, the corresponding 2x2 quadrant would get selected and Stage 3 commences.

3. **Stage 3**: The 2x2 quadrant selected in Stage 2 into four 1 by 1 grids. Once Unity receive command, the corresponding 2x2 quadrant would get selected and Stage 4 commences, the pick up stage.

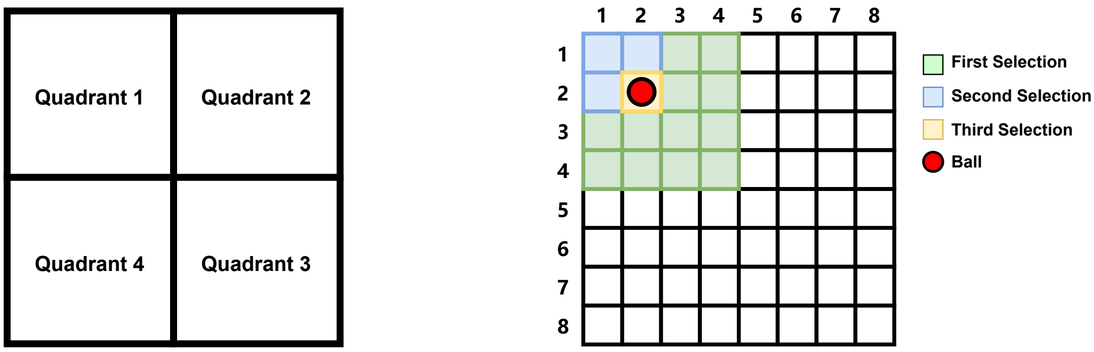
4. **Stage 4**: Once the 1x1 grid is underneath the ball and the robot is ready the ball up. Two consecutive commands is required to confirm the quadrant. Once a 1x1 grid have been confirmed, the pickup stage will begin. The location of the 1x1 grid selected in Stage 3 and a pickup command are sent to the robot.

Another difference of this system compared to the grid navigation is that it has a back button instead of confirm button as the fifth button. It allows the subject to go back to the previous stage. This feature adds an option to dismiss the previous command in cases of human error or inaccurate UDP classification. It was implemented in order to reduce the fail rate of the task. The downside is that it would increase the time needed to pickup the ball in false negative cases (correct quadrant but the back button is pressed).

Similar to the grid navigation system, the quadrant system takes in five controls which corresponds to the following UDP commands. An example command sequences for the same ball position as grid navigation path example is given in Fig 7.3b. In this example, only four command is needed to pick up the ball: Quadrant 1 (First Selection), Quadrant 1 (Second Selection), Quadrant 3 (Third Selection), Quadrant 3 (Third Selection Confirm).

1. Select Quadrant 1 (UDP 1)
2. Select Quadrant 2 (UDP 2)

3. Select Quadrant 3 (UDP 3)
4. Select Quadrant 4 (UDP 4)
5. Back Button (UDP 5)



(a) Quadrant Layout

(b) Quadrant Path Example

Figure 7.3: Quadrant Navigation System

7.3.3 VR Environment: Enclosed Room

The VR environment consists of a room enclosed by four walls, floor and ceiling. The robot is placed on a table in the center of the room.

7.3.4 Heads Up Display (HUD)

The HUD seen through the VR headset consists of components that:

- **SSVEP Checker Stimuli Block:** The stimuli block provides visual stimulus to enable the usage of SSVEP, and has a checker pattern(refer to chapter 6.3.1 figure 6.3, 6.1 and 6.2)
- **Last SSVEP Stimuli Block Indicator:** The indicator indicates the last decoder result, which corresponds to a SSVEP checker stimuli block (refer to chapter 4 6.3.2 figure 6.5)
- **Processing Window Timing Bar:** The SSVEP and motor BCI requires processing time before the decoder outputs results. The processing window timing bar indicates the start and end of a processing windows. It is a green rectangle with initial length x...(unity unit) and shrinks in distance over the processing window's duration (Refer to figure 5.2). By the end of the each processing window determined duration, the rectangle disappears, it returns to the initial length at the start of the next processing iteration.
- **Score Text:** The robot's successful or fail attempts to pick up the ball is kept score by the score text, refer to figure 7.4. 'Score' refers to successful pick up, 'Fail' refers to pick up attempts that failed.

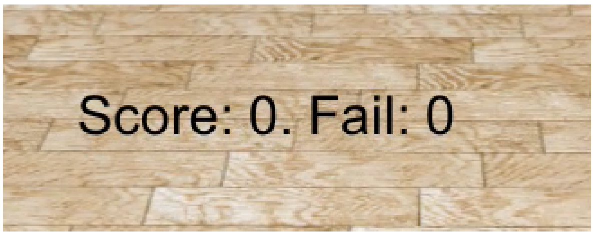


Figure 7.4: score

7.4 Task Performance specification and evaluation criteria

In order to evaluate the task, multiple trials will need to be conducted. The performance of the task will be evaluated using the evaluation criteria, and the performance specification will be based on the evaluation criteria.

7.4.1 Evaluation Criteria

The evaluation criteria to be used to evaluate the task performance are:

- **Number of Success VS Failure:** a ratio of the number of the times robot successfully picked up the ball and the number of times the robot failed to pick up the ball
- **Trial success rate:** The percetage of successful ball pick up attempts out of total attempts
- **Average Task Time:** the average time for task completion, measured from when ball spawns/re-spawn, till when the robot's end effector reaches via point 3 (above the bucket).
- **Worst Case Task Time :** the trial where the task completion time is the longest out of total trials, out of all successful ball pick up trials or out of all failed ball pick up trials
- **Best Case Task Time:** the trial where the task completion time is the shortest out of total trials, out of all successful ball pick up trials or out of all failed ball pick up trials.

7.4.2 Desired Performance specification

The Desired performance specifications to achieve for the task was suggested by the group's mentor Jing. They would be used as a guideline and will not be strictly followed.

1. The task's Best Case Task Time out of successful trials should be less than 30 seconds.

2. The task's average task time should be 60 seconds, which is double the Best Case Task Time (out of successful trials) from performance specification 1.

3. Minimum 80 percent trial success rate of the task. A suitable trial success rate is depend on the application context, given the context of this project is a virtual application for analysis purpose and there are no safety concern or real life application directly impacted, the group deemed 80 percent as the minimum trial success rate to be acceptable.

7.4.3 Summary

The task to pick up a ball by robot arm was designed in detailed in response to the project's problem statement2.2. The Objects and components used during the task, such as the robot, the VR room and the navigation system have their specification highlighted in this chapter and these information will sever as important background information reading into the next chapter.

Chapter 8

Experiment 2: Robot Picking Up Ball

8.1 Chapter Introduction

This chapter explains the experiments conducted using three distinct setup consisting of different combinations navigation system (Section 7.3.2) and BCI methods investigated in Experiment 1 (Chapter 6). Hypothesis of the performance for each setup were proposed and later validated with analysis of the results. The advantage and disadvantage of each setup were also discussed. Finally, the performance of each setup would be compared based on the task performance specifications, upon which setup 3 would be chosen as the best setup.

8.2 Aim

- To investigate the feasibility of achieving the project objective of robot picking up ball using 3 setup combinations of navigation system and BCI methods:
  1. Grid Navigation System with SSVEP and MI hybrid
  2. Grid Navigation System with SSVEP
  3. Quadrant Navigation System with SSVEP
- To investigate the advantages and disadvantages of each setup and compare their performance with each other.
- Determine the most suitable setup that closely matches the performance criteria listed in (Chapter 7.4).
- To validate the hypothesises made on the performance of the 3 setups proposed.

8.3 Experiment Setup & Method

In this experiment, grid and quadrant navigation system were combined with both SSVEP and MI BCI methods to complete the task designed from (Chapter 7). Three different combination of navigation system and BCI methods were tested: grid navigation system using a hybrid of SSVEP and MI BCI methods, grid and quadrant navigation system using only SSVEP BCI methods. Each experiment requires Test Subject 3 from Experiment 1 to pick up the ball using the corresponding navigation system 30 times. The processing window lasts five seconds followed by one second delay for the subject to adjust his focus to another box, if necessary. After the pickup command is registered, the next command will be ignored because the robot is still picking up the ball.

Figure 10.2 provides an overview of the connection between UDP and the SSVEP stimuli. The frequency used for the SSVEP stimuli were 9, 11, 13, 15, 17 Hz with additional phase delays added according to equation 5.1 from (Chapter 5.5) [23]:

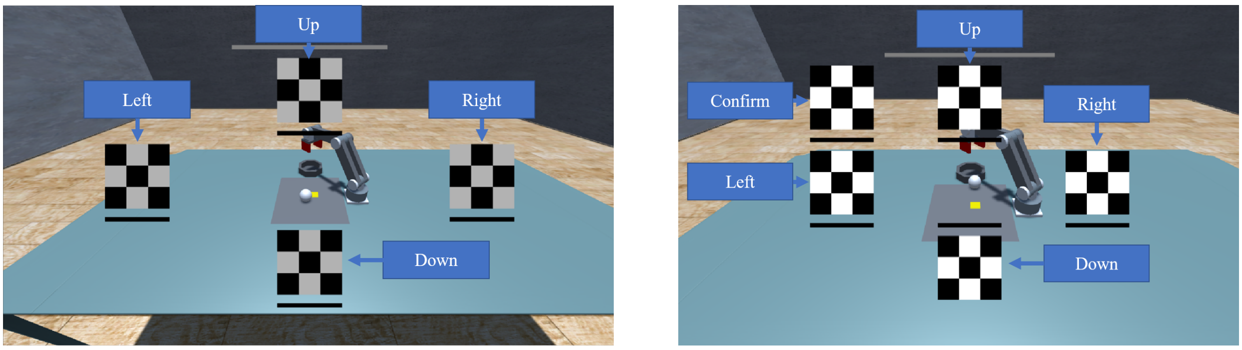
$$PD = \frac{k-1}{4f_k}$$

Where k is the box number. the stimuli would have the following phase delay:

- Box 1 (9 Hz): No phase delay
- Box 2 (11 Hz): 1/4 of its period
- Box 3 (13 Hz): 2/4 of its period
- Box 4 (15 Hz): 3/4 of its period
- Box 5 (17 Hz): 4/4 of its period

FBCCA was set to use the same weighting function in Experiment 1, with banks set at (6-60Hz,12-60Hz,18-60Hz,24-60Hz,32-60Hz,40-60Hz,48-60Hz,54-60Hz).

8.3.1 Setup 1: Grid Navigation System with SSVEP and MI hybrid



(a) SSVEP Motor Hybrid (b) SSVEP with Five stimuli

Figure 8.1: Setup with Grid Navigation

When MI beta band power difference (P(C3)-P(C4)) 5.9 crosses the threshold value, robot is triggered to pick up at cursor location. When no MI is imagined by the subject, beta band power difference oscillates around threshold value and this is undesirable. The 'right hand squeeze' through out the navigation process maintains the MI beta band power difference below threshold.

Additional changes were made to the MI setup from Experiment 1 before it's usage in Setup 1. The MI threshold was 0 μV^2 for experiment 1, in experiment 2 threshold $5\mu V^2$ was used to decrease false positives when conducting right hand squeeze, given right hand squeeze has heavy usage as it is required to be maintained through out the navigation process.

Below is the setting of Setup 1:

1. **BCI methods:** SSVEP and MI
2. **SSVEP:** 4 stimulus, 'up'(9Hz), 'down'(11Hz),'left'(13Hz) and 'right' (15Hz)
3. **MI:** 2 imagery, 'left hand squeeze' and 'right hand squeeze'
4. MI threshold: 5 μV^2

5. **Processing window** 5 seconds for both SSVEP and MI
6. **Subject:** subject 3 from experiment 1
7. **Number of Trials:** 30
8. **Setup figure:** Fig 8.1a
9. **UDP:**
    - UDP 1 <=> SSVEP checker stimuli 'Up' and 'imaginary right hand squeeze'.
    - UDP 2 <=> SSVEP checker stimuli 'Right' and 'imaginary right hand squeeze'.
    - UDP 3 <=> SSVEP checker stimuli 'Down' and 'imaginary right hand squeeze'.
    - UDP 4 <=> SSVEP checker stimuli 'Left' and 'imaginary right hand squeeze'.
    - UDP 5 <=> 'Imaginary left hand squeeze'.

8.3.2 Setup 2: Grid Navigation System with SSVEP

This setup replaces the MI used for ball pick up 'confirm' with a SSVEP stimuli.

1. **BCI methods:** SSVEP
2. **SSVEP:** 5 stimulus, 'up'(9Hz), 'down'(11Hz),'left' (13Hz), 'right' (15Hz) and 'back' (17Hz)
3. **Processing window:** 5 seconds for SSVEP
4. **Number of Trials:** 30
5. **Subject:** subject 3 from experiment 1
6. **Setup figure:** Fig 8.1b
7. **UDP:**
    - UDP 1 <=> SSVEP checker stimuli 'up'
    - UDP 2 <=> SSVEP checker stimuli 'right'
    - UDP 3 <=> SSVEP checker stimuli 'down'
    - UDP 4 <=> SSVEP checker stimuli 'left'
    - UDP 5 <=> SSVEP checker stimuli 'confirm'

8.3.3 Setup 3: Quadrant Navigation System with SSVEP

This setup is designed with the desire to maintain a consistent task time through out the trials, given minimum navigation move is 3 (excluding 'pick up' move), provided no mistakes are make.

1. **BCI methods:** SSVEP
2. **SSVEP:** 5 stimulus, 'quadrant 1'(9Hz), 'quadrant 2'(11Hz),'quadrant 3'(13Hz), 'quadrant 4'(15Hz) and 'back'(17Hz)
3. **Processing window:** 5 seconds for SSVEP
4. **Number of Trials:** 30
5. **Subject:** subject 3 from experiment 1
6. **Setup figure:** (Fig 8.2)
7. **UDP:**
    - UDP 1 <=> SSVEP checker stimuli 'quadrant 1'
    - UDP 2 <=> SSVEP checker stimuli 'quadrant 2'
    - UDP 3 <=> SSVEP checker stimuli 'quadrant 3'
    - UDP 4 <=> SSVEP checker stimuli 'quadrant 4'
    - UDP 5 <=> SSVEP checker stimuli 'back'

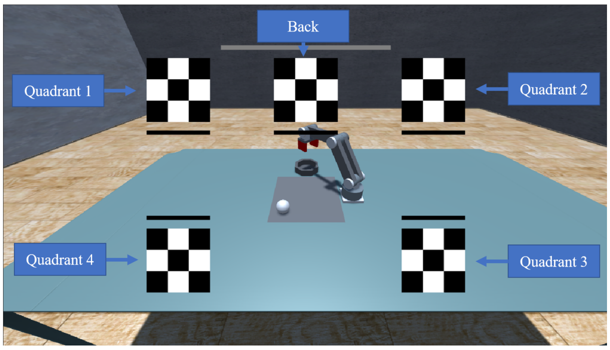


Figure 8.2: Setup with Quadrant Navigation System using SSVEP with Five stimuli

8.4 Task Performance Hypothesis

The minimum successful trial's possible best case task time is calculated using minimum moves assuming no user navigation mistakes giving the following estimation:

- Setup 1 Grid SSVEP MI hybrid : 2 moves x 5 secs = minimum 10 seconds
- Setup 2 Grid SSVEP: 2 moves x 5 secs = minimum 10 seconds
- Setup 3 Quadrant: 4 moves x 5 secs = minimum 20 seconds

The minimum successful trial's possible worst case task time is calculated using maximum moves assuming no user navigation mistakes, giving the following estimation:

- Setup 1 Grid SSVEP MI hybrid: 8 moves x 5 secs = 40 secs
- Setup 2 Grid SSVEP: 8 moves x 5 secs = 40 secs
- Setup 3 Quadrant: 4 moves x 5 secs = 20 seconds

The average task time is hypothesised to be as the following:

- Setup 1 'Grid SSVEP MI hybrid' will have a lower trial success rate than Setup 2 'Grid SSVEP' because MI using 5 second processing windows has lower accuracy than SSVEP using same processing window, and it's average task time will increase as a result.

- setup 3 using Quadrant navigation system will have average task time less than setup using grid navigation system.

8.5 Results

The results were collected for the three setups by recording screen of the VR Environment and then analysis the timing of each trial using Excel.The 30 trial times of the 3 setups are given in table Appendix10.8 figure 10.1. The Elevation criteria results are listed in the table below:

8.6 Discussion

8.6.1 Meeting Task Performance Specification

Only Setup 3 using quadrant navigation system met all three task performance specification.

	# success vs failure	trial success rate %	avg time: (sec)	avg time: success only (sec)	avg time: failure only (sec)	worst case time (sec)	worst case time: success (sec)	worst case time: failure (sec)	best case time: success (sec)
Grid SSVEP	28:2	93.33	74.067	74.714	65	154	154	93	24
Grid Hybrid	21:9	70	76.607	71.255	91.325	169.2	130	169.2	18
Quadrant SSVEP	25:5	83.33	58.867	58.856	58.92	134.4	126	134.4	20

Table 8.1: Experiment 2 Results Comparison and Analysis

Meeting Task Performance Specification	Spec 1	Spec 2	Spec 3
Setup 1 Grid + SSVEP MI hybrid	Yes	No	No
Setup 2 Grid + SSVEP	Yes	No	Yes
Setup 3 Quadrant+ SSVEP	Yes	Yes	Yes

Table 8.2: Setup Task Performance Specification Matrix

8.6.2 Performance Comparison

Setup 1 has a lower trial success rate than Setup 2 by 23.33 percents. Setup 3's average time from total trial, successful trials and failure trial is the least out of all setup as hypothesised. The worst case task time for setup 3 is the smallest because the The worst case task time using a Quadrant navigation system is smaller, at 10 seconds. The failure worst case for setup 2 is smaller because the least move required is 2 compared to minimum move of 4.
The most suitable setup would be setup 3, because it does well in time performance criteria and have trial success rate. The least suitable setup is setup 1, based on the evaluation criteria and performance specification.
Possible factors that affected the results of the experiment can be the subject losing focus after extended time. Making a navigation mistakes greatly penalises the time of the trial, because 1 mistakes requires 2 more moves to correct.

8.6.3 Disadvantages and Advantages of Each Setup

Setup 2 and Setup 3 had 5 SSVEP stimuli, which is 1 more stimulus than setup 1, the extra stimulus would take up more screen space and block the subject's view of the robot and ball spawn area. This several disadvantages, one being the user may feel frustrated to have their view obstructed by the SSVEP stimuli blocks due to visual comfort, the other disadvantages is zooming in on the ball and robot is limited because the stimuli may block the line of sight to the ball. Further disadvantages can be the extra stimulus block diverting the subjects attention and thus reducing the trial success rate.Further more, setup 2 and 3 has advantages over setup 1 because MI requires user training to and SSVEP does not. [17].

In addition, the positioning of the SSVEP stimuli blocks for quadrant and grid navigation system are different, placement of the stimuli in the corner is preferred, given the wide screen nature of the VR headset screen dimension. Hence setup 3 has an advantage over more optimal stimuli layout.

8.7 Improvement

Improvements to consider are that a bigger number of trials may be required to have a more accurate average time with setup using grid navigation system, given the ball spawns randomly. The overlap of VR headset band over the occipital lobe, applies unconformable pressure on to the dry electrodes. Addition of external support structure such as padding can be considered to improve this draw backs, and thus increase the increase the subject's willingness to conduct experiment of extensive duration.

8.8 Summary

The experiment concluded on the note of successfully implementing the overall project objective. The task of the robot picking up a ball is most suited using the setup 3, which uses the quadrant navigation system and BCI method of SSVEP. It provides the the best performance in terms of trial time and also trial success rat. on the contrary, incorporating MI is not a good solution given time performance and trial success rate will be penalised. All trials were conducted with subject 3, which limits the analysis of the task performance. Further improvements can be condcted to increase both task performance and subject comfort when conducting task.

Chapter 9

Conclusion and Recommendation

In conclusion, this project approves the feasibility to incoperat SSVEP and MI BCI methods as controls into a working VR application, in which construction of robot arms can be realized, further exploring the possibilities to AR integrating with real life robotic application. There is exciting opportunities to be realized, given the nature of a VR environment, which is completely independent of physical and costy components required to prototype products, thus enabling researchers to conduct experiment with more flexibility. Many modern BCIs focused on achieving a high information transfer rate sometimes at the cost of the decoding accuracy, such as high speed SSVEP based typing BCIs. Still, BCI technology has a long way before producing robust commercially available products.
Unlike typing that can be corrected by spellers, complex real world tasks are often comprised of many successive sub tasks, and each sub task must be implemented successfully to achieve overall success. For instance, the robot controlling task proposed in this project requires the user to navigate to the correct target location and only orders the robot to execute a pick up action at the correct target location. Although both the grid by grid navigation and the quadrant navigation system allowed the user to make mistakes, the user needed at least two correct commands to mend one incorrect mistake made by the BCI decoder. Hence, the robot's average task times using a setup with grid or quadrant navigation system are almost triple that of the best case task time for each respective setup (setup 3's average time of 58.856 sec being more than triple the best case task time of 20 sec)

Future Improvements and Recommendations:

- Investigate alternative sources of stimuli or different settings for the stimuli that are used in this project. A new setup could potentially yield better results. With more accuracy, the task can be performed even faster with lower processing time.
- A quiet lab room could produce more consistent results for BCI testing, where sound or visual noises could be mitigated or isolated easily
- Recruit a larger number of participants to test the performance of a BCI. This could provide more meaningful results for analysis to explore possibilities to increase the universality of a BCI system.
- Investigate a way to increase the number of control the user have e.g. adding more SSVEPs block or reintegrating MI. I.e, motor Imagery based BCIs could be extended to more control targets and higher accuracy by using machine learning techniques, and is also a possible direction to explore.

References

[Hot33] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[Wel67] Peter Welch. "The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms". In: *IEEE Transactions on audio and electroacoustics* 15.2 (1967), pp. 70–73.

[FD88] Lawrence Ashley Farwell and Emanuel Donchin. "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials". In: *Electroencephalography and clinical Neurophysiology* 70.6 (1988), pp. 510–523.

[Reg89] David Regan. "Evoked potentials and evoked magnetic fields in science and medicine". In: *Human brain electrophysiology* (1989).

[Jea95] Marc Jeannerod. "Mental imagery in the motor context". In: *Neuropsychologia* 33.11 (1995), pp. 1419–1432.

[MLO+96] Christopher JL Murray, Alan D Lopez, World Health Organization, et al. "The global burden of disease: a comprehensive assessment of mortality and disability from diseases, injuries, and risk factors in 1990 and projected to 2020: summary". In: (1996).

[Pfu+97] Gert Pfurtscheller et al. "EEG-based discrimination between imagination of right and left hand movement". In: *Electroencephalography and clinical Neurophysiology* 103.6 (1997), pp. 642–651.

[Kle+99] George H Klem et al. "The ten-twenty electrode system of the International Federation". In: *Electroencephalogr Clin Neurophysiol* 52.3 (1999), pp. 3–6.

[01] "Information transfer rate in a five-classes brain-computer interface." In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering, Neural Systems and Rehabilitation Engineering, IEEE Transactions on, IEEE Trans. Neural Syst. Rehabil. Eng* 3 (2001), p. 283. ISSN: 1558-0210. URL: https://ezp.lib.unimelb.edu.au/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.948456&site=eds-live&scope=site.

[Wol+02] Jonathan R Wolpaw et al. "Brain–computer interfaces for communication and control". In: *Clinical neurophysiology* 113.6 (2002), pp. 767–791.

[Odo+04] J Vernon Odom et al. "Visual evoked potentials standard (2004)". In: *Documenta ophthalmologica* 108.2 (2004), pp. 115–123.

[Lin+06] Zhonglin Lin et al. "Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs". In: *IEEE transactions on biomedical engineering* 53.12 (2006), pp. 2610–2614.

[MR07] Päivi Majaranta and Kari-Jouko Räihä. "Text entry by gaze: Utilizing eye-tracking". In: *Text entry systems: Mobility, accessibility, universality* (2007), pp. 175–187.

[Wan+07] Yijun Wang et al. "Design of electrode layout for motor imagery based brain–computer interface". In: *Electronics Letters* 43.10 (2007), pp. 557–558.

[SAG09] Indar Sugiarto, Brendan Allison, and Axel Gräser. "Optimization strategy for SSVEP-based BCI in spelling program application". In: *2009 International Conference on Computer Engineering and Technology*. Vol. 1. IEEE. 2009, pp. 223–226.

[Bla+10] Benjamin Blankertz et al. "Neurophysiological predictor of SMR-based BCI performance". In: *Neuroimage* 51.4 (2010), pp. 1303–1309.

[NG12] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. "Brain computer interfaces, a review". In: *sensors* 12.2 (2012), pp. 1211–1279.

[Che+14] Xiaogang Chen et al. "A high-itr ssvep-based bci speller". In: *Brain-Computer Interfaces* 1.3-4 (2014), pp. 181–191.

[Ko+14] Li-Wei Ko et al. "Developing a few-channel hybrid BCI system by using motor imagery with SSVEP assist". In: *2014 International joint conference on neural networks (IJCNN)*. IEEE. 2014, pp. 4114–4120.

[Tat14] William O Tatum IV. *Handbook of EEG interpretation*. Demos Medical Publishing, 2014, pp. 155–190.

[AJ15] Minkyu Ahn and Sung Chan Jun. "Performance variation in motor imagery brain–computer interface: a brief review". In: *Journal of neuroscience methods* 243 (2015), pp. 103–110.

[Che+15a] Xiaogang Chen et al. "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain–computer interface". In: *Journal of neural engineering* 12.4 (2015), p. 046008.

[Che+15b] Xiaogang Chen et al. "High-speed spelling with a noninvasive brain–computer interface". In: *Proceedings of the national academy of sciences* 112.44 (2015), E6058–E6067.

[KKH15] Ivo Käthner, Andrea Kübler, and Sebastian Halder. "Rapid P300 brain-computer interface communication with a head-mounted display". In: *Frontiers in neuroscience* 9 (2015), p. 207.

[Koo+15] Bonkon Koo et al. "Immersive BCI with SSVEP in VR head-mounted display". In: *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE. 2015, pp. 1103–1106.

[Arm+16] Brian S Armour et al. "Prevalence and causes of paralysis—United States, 2013". In: *American journal of public health* 106.10 (2016), pp. 1855–1857.

[Luu+17] Trieu Phat Luu et al. "EEG-based brain-computer interface to a virtual walking avatar engages cortical adaptation". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2017, pp. 3054–3057.

[Sta+18] Piotr Stawicki et al. "SSVEP-based BCI in virtual reality-control of a vacuum cleaner robot". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2018, pp. 534–537.

[Smi19] Julius O. Smith. *Spectral Audio Signal Processing*. online book, 2011 edition. `http://-ccrma.stanford.edu/~jos/sasp/`, 2019.

[Sta+19] Piotr Stawicki et al. “Remote Steering of a Mobile Robotic Car by Means of VR-Based SSVEP BCI”. In: *Advances in Computational Intelligence*. Ed. by Ignacio Rojas, Gonzalo Joya, and Andreu Catala. Cham: Springer International Publishing, 2019, pp. 406–417. ISBN: 978-3-030-20521-8.

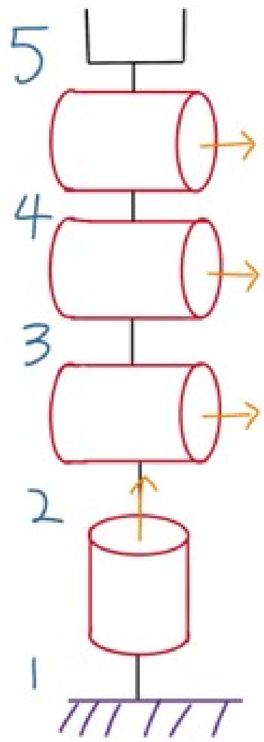
[Vou+19] Athanasios Thanos Vourvopoulos et al. “Efficacy and Brain Imaging Correlates of an Immersive Motor Imagery BCI-Driven VR System for Upper Limb Motor Rehabilitation: A Clinical Case Report”. In: *Frontiers in human neuroscience* 13 (2019), p. 244.

[Zha+19] Weili Zhao et al. “A Virtual Smart Home Based on EEG Control”. In: *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE. 2019, pp. 85–89.

Chapter 10

Appendix

10.1 Robot's Axis



[h]

Figure 10.1: Robot's Axis

10.2 Unity Application Repository

Github URL: https://github.com/harfiyanto/EEG-VR

10.3 Additional hardware and software details

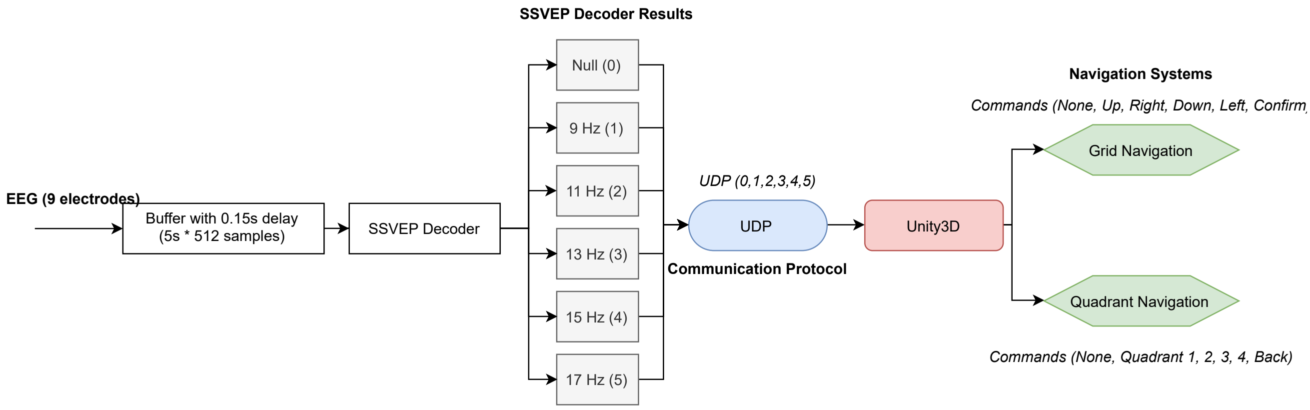


Figure 10.2: UDP Flow Chart

10.4 Matlab Script

```matlab
%% Clean start
clc;clear all;close all;

%% EEG SSVEP Processing Variables
FS = 512;                                        % sampling frequency
SSVEP_WINDOW_T = 5;                              % length of SSVEP processing window in seconds
SSVEP_WINDOW_S = round(SSVEP_WINDOW_T*FS)       % length of SSVEP processing window in samples
SSVEP_DELAY_T = 0.1465;                          % length of ssvep processing delay in seconds
SSVEP_DELAY_S = round(SSVEP_DELAY_T*FS)         % length of ssvep processing delay in samples
SSVEP_OVERLAP_T = SSVEP_WINDOW_T-1;                            % length of SSVEP processing overlap in ...
    seconds
SSVEP_OVERLAP_S = round(SSVEP_OVERLAP_T*FS)      % length of SSVEP processing overlap in samples
STI_F = [9,11,13,15,17];                             % stimulus frequencies
N_STIF = length(STI_F);                          % number of stimulus frequencies

MOTOR_WINDOW_T = 5;                              % length of motor processing window in seconds
MOTOR_WINDOW_S = round(MOTOR_WINDOW_T*FS)        % length of motor processing window in samples
MOTOR_OVERLAP_T = 0;                             % length of SSVEP processing overlap in seconds
MOTOR_OVERLAP_S = round(MOTOR_OVERLAP_T*FS)     % length of SSVEP processing overlap in samples
MOTOR_SIMPLE_THRESHOLD = 0;                     %simple threshold for left/right detection

%% FBCAA specific Processing Variables
H_REFSIG = 2;                                    % number of harmonics in refsig
N_BAND = 9;                                      % number of frequency bands
FBCCA_WN = [6,60];                               % entire usable frequency range in Hz
FBCCA_subbandwidth=(FBCCA_WN(2)-FBCCA_WN(1))/N_BAND; % each band's band width in Hz
% this is for storting the filter coefficients for the filter used
numx = zeros(9,13);
denx = zeros(9,13);

% the weighting coefficients for combining the banks' CCA weighting
% check simulink block diagrams for explicit details
a = 1.25;
b = 0.5;

% % this part is actually hard coded into simulink, just as reference here
% for fb=1:N
%      Wp = [FBCCA_WN(1)+(fb-1)*FBCCA_subbandwidth, FBCCA_WN(2)]/(Fs/2);
%      [num,den] = cheby1(6,1,Wp,'bandpass');
%      num1 = num;
%      den1 = den;
% end

%% Generating Reference Signals
% References signal with sine-cosine square waveforms
refsig_1 = refsig_sin(STI_F(1),FS,SSVEP_WINDOW_S-SSVEP_DELAY_S,H_REFSIG);
refsig_2 = refsig_sin(STI_F(2),FS,SSVEP_WINDOW_S-SSVEP_DELAY_S,H_REFSIG);
refsig_3 = refsig_sin(STI_F(3),FS,SSVEP_WINDOW_S-SSVEP_DELAY_S,H_REFSIG);
refsig_4 = refsig_sin(STI_F(4),FS,SSVEP_WINDOW_S-SSVEP_DELAY_S,H_REFSIG);
refsig_5 = refsig_sin(STI_F(5),FS,SSVEP_WINDOW_S-SSVEP_DELAY_S,H_REFSIG);

%% For offline experiments
%
% % number of trials for all stimulus frequencies
% Stimulus_N = 50;
%
% % initial_rest time in seconds
```

```
% Stimulus_start_Time = 30;
%
% % the sequence to be used
% Stimulus_frequencies_raw = [];
% for i = 1:Stimulus_start_Time
%     Stimulus_frequencies_raw = [Stimulus_frequencies_raw ,0];
% end
%
% Stimulus_frequencies_timeseries = zeros(1,Stimulus_start_Time*Fs+window*n_sti*Stimulus_N);
% Stimulus_space = 1:n_sti;
%
% %step 1 conversion to array of stimulus frequency indexes
% for i=1:Stimulus_N
%     x=Stimulus_space(randperm(length(Stimulus_space)));
%     for j=1:n_sti
%         for k = 1:window_time
%             Stimulus_frequencies_raw = [Stimulus_frequencies_raw ,x(j)];
%         end
%         Stimulus_frequencies_raw = [Stimulus_frequencies_raw ,0];
%     end
% end
% for i = 1:Stimulus_start_Time
%     Stimulus_frequencies_raw = [Stimulus_frequencies_raw ,0];
% end
% Stimulus_frequencies_raw = timeseries(Stimulus_frequencies_raw);
```

10.5 Simulink models

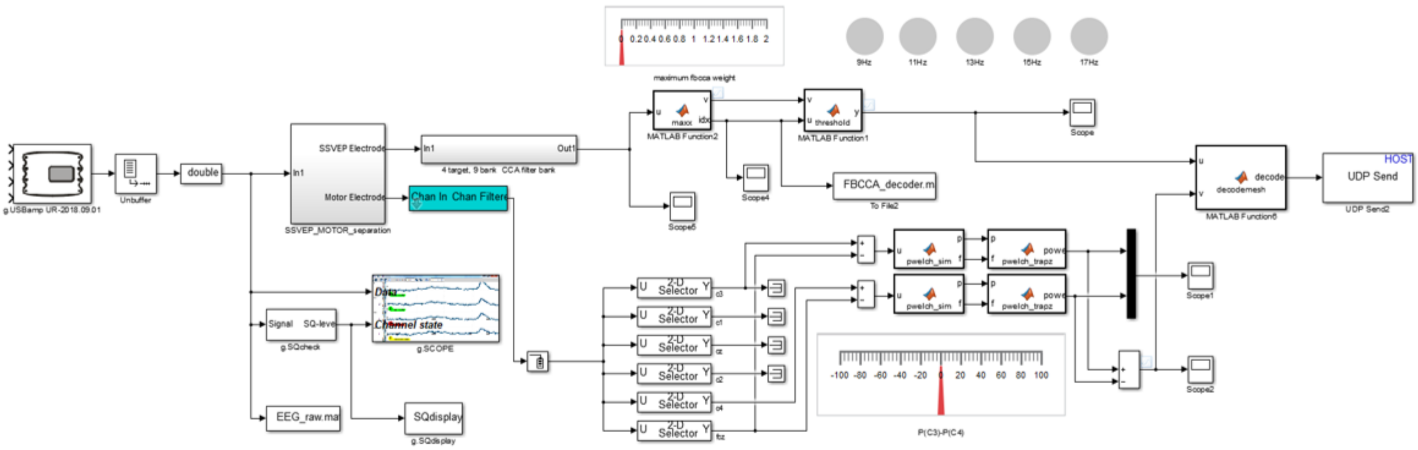


Figure 10.3: Online Simulink

10.6 Gantt Chart

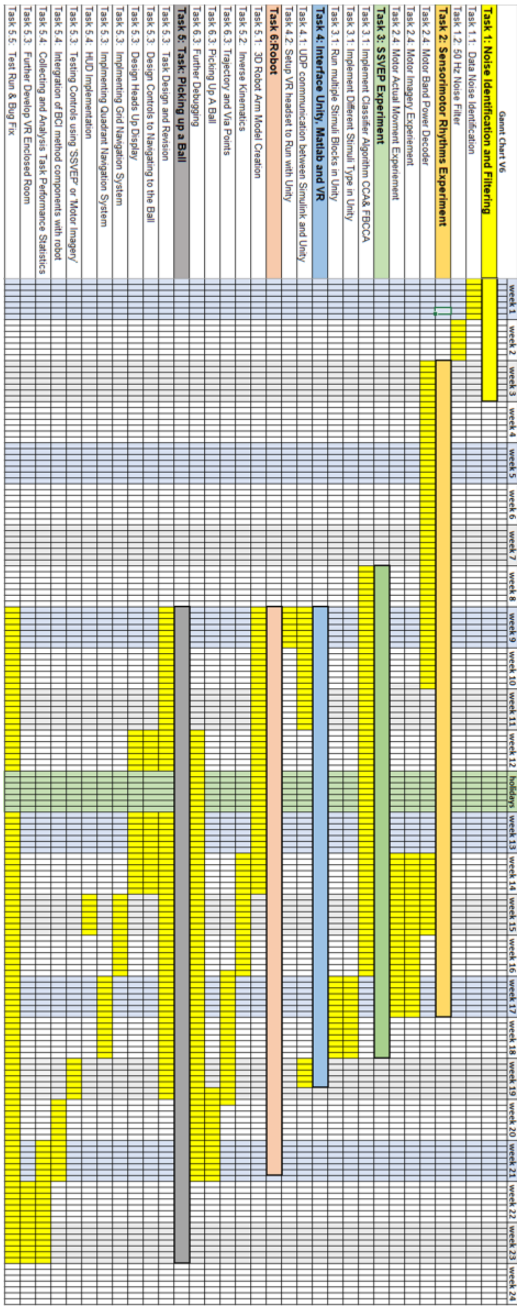


Figure 10.4: Gantt Chart version 6

10.7 Project Diary

Week, Semester	Time	Participants	Meeting Outcomes
Week 2, S1	1h	Albert, Harfi, Jiawei, Jing, Denny	Need to verify and filter possible noises from VR headset
Week 5, S1	1h	Albert, Harfi, Jiawei, Jing	Implement motor imagery by looking at left and right brain signal power to create a binary control.
Week 7, S1	1h	Albert, Harfi, Jiawei, Jing	Abandoned the left and right brain signal power method. Try to replicate Motor Imagery from a paper.
Week 10, S1	1h	Albert, Harfi, Jiawei, Jing	Turning point. Moving on to SSVEP because couldn't get motor imagery to work well.
Week Holiday	1h	Albert, Harfi, Jiawei	Implemented SSVEP. Jing suggested using FBCCA rather than CCA to imporove accuracy and performance
Week 1, S2	1h	Albert, Harfi, Jiawei	Implemented the robot. Decided to use a cursor navigation system to control where robot picks up things, automate the robot pick up trajectory generation and process.
Week 3 S2	1h	Albert, Harfi, Jiawei	Decision to investigate how to make the SSVEP stimuli more visually comforting while mainting accuracy. Decided to try checker pattern and resizing boxes (alternating between big and small).
Week 5 S2	1h	Albert, Harfi, Jiawei, Jing	Devised two navigation system for the cursor: quadrant and grid. Jing suggested to include back button for quadrant.
Week 7 S2	1h	Albert, Harfi, Jiawei	Devised the setup combinations between the two navigation systems (Grid and Quadrant) and the BCI methods.
Week 9 S2	1h	Albert, Harfi, Jiawei, Denny	Denny explained how we should go about analysis the performance of the task desiged and how to write our report.
Week 11 S2	1h	Albert, Harfi, Jiawei	Decided on the experiment evaluation criteria for the task designed and further narrowed down on what setup combination to use for the task.

10.8 Additional Experiment Results

	Grid SSVEP & MI Hybird		Grid SSVEP		Quadrant SSVEP: 5 seconds processing window		Quadrant SSVEP: 4 seconds processing window	
trial	score	time per task (sec)	score	time per task (sec)	score	time per task (sec)	score	time per task (sec)
1	success	90	success	33	success	38	success	20
2	success	73	success	71	success	76	success	84
3	success	82	success	77	success	94	success	84
4	success	48	success	66	success	24	success	31
5	fail	131	success	64	success	60	fail	139
6	fail	10	fail	30	success	88	success	61
7	success	18	success	70	success	82	fail	99
8	success	61	success	42	success	36	fail	96
9	success	69	success	83	success	112	fail	90
10	success	90	success	69	success	61	fail	64
11	success	76	success	72	success	47	fail	15
12	success	95	success	31	success	136	success	60
13	success	88	success	69	success	71	success	41
14	success	83	success	88	success	82	success	79
15	success	130	success	36	success	83	success	10
16	fail	150	success	126	success	48	success	176
17	fail	7	success	32	fail	93	success	20
18	fail	48	success	49.2	success	61	success	79
19	success	49.2	fail	134.4	success	64	success	25
20	success	18	fail	37.8	success	154	fail	26
21	success	49.2	success	20	success	78	fail	70
22	fail	137.4	success	63.6	success	88	success	75
23	success	88.2	success	60	success	87	fail	79
24	success	18	success	49.2	fail	37	success	81
25	fail	169.2	fail	14.4	success	60	fail	14
26	success	67.2	success	98.4	success	66	success	71
27	fail	78	fail	78	success	66	fail	10
28	success	94.8	success	38.4	success	111	success	15
29	success	82.2	success	25.2	success	37	fail	54
30	success	97.8	success	38.4	success	82	success	86

Table 10.1: Experiment 2 Results