Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

An SSVEP-based BCI Gaming System.

Final Project Report Submission: Practical Course Biosignal Processing and Modeling

Joseph González Núñez[1], Ritika Gupta[1], Inés Martín Muñoz[1], and Karahan Yilmazer[2]

1Master of Science in Neuroengineering

2Master of Science in Electrical and Computer Engineering

August 5th, 2022

Abstract — By decoding human brain signals, Brain-Computer Interfaces(BCI) intends to create a new control channel between humans and other external devices.They can be used to improve the daily living activities of motor-impaired individuals. Steady-Steate Visual Evoked Potentials (SSVEP) are commonly used in BCIs. SSVEPs are periodic responses that match the frequency of the visual stimulus presented. This paper presents an SSVEP-based BCI system that can be used for gaming. The multichannel electroencephalogram (EEG) signal is recorded using an off-the-shelf device and the frequency information associated with the SSVEPs is extracted to create the control commands that control a block in an virtual game to move forward, to the left and to the right.

1 Introduction

1.1 Brain Computer Interfaces

With the help of brain-computer interface (BCI) technology, people can operate tools and programs without using their hands or other external muscles[1]. Recent developments in brain imaging technologies like electroencephalography, magnetoencephalography, and functional magnetic resonance imaging have sparked the growth of the cognitive neuroscience area.

Due to its non-invasiveness, dependability, portability, and impressive time signal resolution, EEG-based BCI has recently been employed successfully in clinical rehabilitation, assistive mobility, mental-state detection, and games[2].

The application of BCI-based assistive technology can improve the quality of life for those with severe motor disabilities[3]. Despite recent advancements, there are still several challenges to creating a practical and efficient BCI system. The principal difficulties were usability, pricing, accuracy, and speed. Current BCI systems are unreliable and transmit information slowly. This implies that sending commands to the gadget being managed by the user takes a lot of time. The price of EEG apparatus, such as an EEG hat and amplifiers, is another issue[4].

1.2 Overview of BCI Paradigms

The different types of BCI paradigms are discussed below:

1. **Spontaneous Potentials:** When the test subject is not given any stimuli, spontaneous EEG is measured. The spontaneous EEG is recorded from healthy participants over a period of time during which the brain activity transforms continuous waves into events of varying frequency. In spontaneous EEG waves, characteristics of various cognitive processes, mental states, and activation processes can be seen. One can associate a certain mental task with the appearance of particular frequency bands over a particular area of the brain. Different behaviours, ideas, and mental states have an impact on the EEG patterns which is then studied for different subjects.
2. **Event Related Potentials (ERP):** The ERP are distinct from spontaneous brain activity in a way that they emerge during stimulation of the subject and are identified through thorough data analysis. In

addition to producing constant spontaneous activity, the brain also changes its potential in a distinctive way in response to specific internal or external stimuli. If no stimulation is provided, episodic stimulation causes event-based activity to be registered but not shown. A certain response and particular EEG components are anticipated to appear in the ongoing EEG activity after the individual is exposed to an external stimulus (such a click sound or a flashing light). Utilizing triggers, or timestamps of stimulus presentation recorded in the EEG, these ERP are analysed in the time domain.

3. **Evoked Potentials (EP):** A subset of the ERP known as the evoked potentials (EP) which develops in response to or while being attentive to specific physical stimuli (auditory, visual, somatosensory, etc.). The EP result from the reorganization of phase of the ongoing EEG signals. The visual evoked potential (VEP) over the visual cortex fluctuates at the same frequency as the stimulating light is one example of how the EP can have distinguishing features connected to different stimulus properties[5]. Other EP are also utilized, such as the auditory evoked potential (AEP)[6].
   Based on the stimulation frequency, the evoked potentials can be divided into transient EP where the stimulation frequency is less than 2Hz and a steady state EP (SSEP) when the stimulation frequency is higher than 6Hz. The SSEP is a periodic response. The stimulation frequency or harmonics of that frequency must cause an increase in signal power in the band to be considered an SSEP.

4. **Steady State Visually Evoked Potentials (SSVEP):** It has been observed in many research works that as the test participant focuses his attention on the area of visual space containing the stimulus, brain activity stimulated by the visual stimulus increases.
   According to studies, when two or more stimuli are shown concurrently with varying flicker frequencies, the flicker triggers brain responses that draw the subject's attention. The brain reaction for a particular frequency can be analysed since its frequency matches that of the stimulation frequency. These steady state reactions are identified by the EEG recordings as steady state visually evoked potentials (SSVEP)[7]. The SSVEP response may be modulated by attention in many ways, depending on factors such stimulus frequency, stimulus spacing, color, and form. Low frequency flashing is known to generate more intense SSVEP, but it can also make people uncomfortable and easily exhausted.

5. **Motor Imagery:** Motor imaging is defined as performing an imagined movement rather than an actual one. Previous research has demonstrated that the brain's movement-generating regions are activated by imagination[8].
   One of the most used paradigms for motor imaging is the sensorimotor rhythms (SMR) paradigm. According to this paradigm, the term "imagined movement" refers to the imagination of kinetic movements of substantial bodily parts, such as the hands, feet, and tongue, which may cause changes in brain activity[9].
   A motor imagery paradigm called imagined body kinematics (IBK) emerged from intrusive BCI technology. IBK challenges the subject to visualize the unbroken motion of a single body part in a three-dimensional environment. After that, time domain decoding of the captured signals is performed. It's also known as a natural imaginary movement when referring to this paradigm.

6. **Hybrid BCI:** According to each BCI type's benefits, a hybrid BCI typically mixes them in series or parallel mode [10]. The major goal of hybrid BCIs is to increase pattern recognition's precision. Not all of the combinations, though, are efficient and practical. Additionally, a hybrid BCI can combine an EEG-based BCI with other BCI types as a magnetoencephalogram (MEG), electro-corticogram (ECoG), functional magnetic resonance imaging (fMRI), and near-infrared spectroscopy (NIRS).

1.3 Typical BCI architecture

The typical BCI architecture consists of the blocks shown in Figure 1. The brain activity of the patient is recorded using the acquisition devices such as Emotiv EPOC for EEG, etc. This process includes channel count and placement determination, amplification, analogue filtering, and A/D conversion. Channel locations are

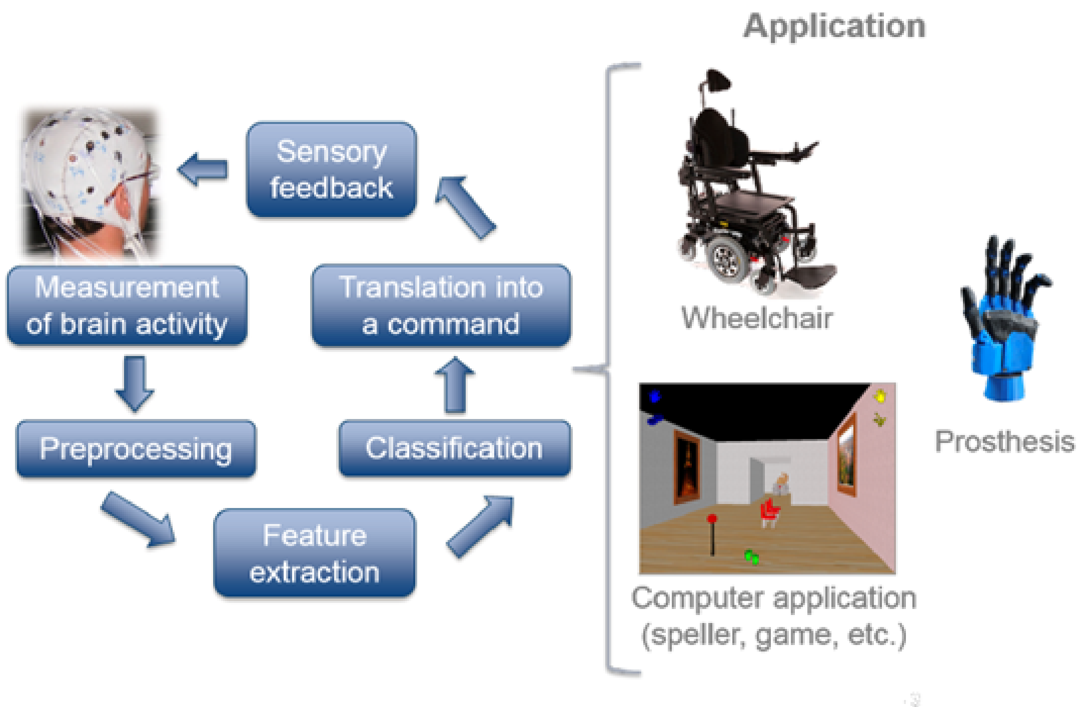


Figure 1 Closed-Loop BCI System and possible applications. The main components of a BCI system are: measurement of the signal, pre-processing of the signal, feature extraction, classification, translation into a command and sensory feedback provided to the user. Source: [11]

chosen based on the mental work being done and the paradigm being used.

The recorded data consists of artifacts due to muscle movements, eye movements, etc. so before feeding the signal to the pipeline the data has to be pre-processed. Various pre-processing techniques such as filtering the notch at 50Hz or 60Hz, dividing the signal into different sub-bands for extracting alpha, beta and gamma bands, using filters to remove the artifacts, etc. Segmenting the EEG data is done after filtering. By dividing the continuous EEG signal into time-locked windows, which typically overlap or are locked to a stimulus, this can be accomplished. Averaging is possible with epoching, which also significantly streamlines the feature extraction and classification procedure.
After pre-processing the clean data is fed to the feature extraction block where different features are calculated according to the paradigm chosen. One of the most important steps in the design of BCI systems is to locate and extract useful information from signals. The classification algorithm that uses the EEG features will struggle to categorize the user's mental states if the features are not pertinent and do not adequately describe the signal. If this happens, the correct recognition rates of mental states will be low, making it impossible or inconvenient to use the interface. Therefore, it is advised to choose and extract useful features in order to maximize system performance by simplifying the task of the following classification method. After feature extraction the feature vector is formed and is then given for classification.

There are several ways to accomplish classification, from straightforward thresholding or linear models to sophisticated nonlinear neural network classifiers. To correctly classify a previously extracted feature vector is the aim of classification. This class describes a BCI user's purpose. The classification algorithm is designed in such a way that the output of each of the classes can be given as a command to the BCI. In some cases the classification output is translated into a command to be compatible with the input of the BCI application. The process of translating a user's mental state into a command involves issuing an action that corresponds to that user's mental state, such as directing a spell-checker, manoeuvring a wheelchair, or moving the mouse pointer on a computer screen.

There is sensory feedback provided to the user so that user can alter the future commands based on the sensory feedback which enables the user to intentionally regulate brain activity to improve task performance.

There are various applications in which a typical BCI system can be used some of which include rehabilitation purposes for example wheelchairs for people with motor impairments, prosthesis, etc. Another possible application for BCI could be for computer applications such as gaming. The applications of BCI are discussed in detail in the next subsection.

1.4 Applications of BCI

BCI development is a cutting-edge area of science and technology that calls for interdisciplinary expertise from a variety of disciplines, including engineering, computer science, psychology, and clinical rehabilitation. BCI research has been effectively applied to aiding the disabled, as well as providing a second data input method for healthy individuals. It can be used as an additional channel in a variety of applications, including game control, augmented reality software, home appliance control, stress and fatigue monitoring, and many others.

The medical applications of BCI include:

1. **Prosthetic devices and rehabilitation:** Patients with mild to severe mobility difficulties use the BCI technology. The use of motor imagery (MI) BCI as a therapeutic tool is possible. Patients have attempted to hold things using BCI driven robotic prosthetic hands. The patients received feedback from robotic arms, which aided in their recovery. Despite promising rehabilitation outcomes, controlling a robotic prosthetic limb involves a lot of commands that are beyond the scope of BCI devices.

2. **Diagnosis in medicine:** BCI technology can be utilized to create health monitoring programs that might routinely check the user for early signs of neurological conditions like epilepsy and advise them to consult a doctor for a diagnosis.

3. **Assistive Mobility:** The devices that allow motor impaired people to regain their mobility are the most helpful. This is accomplished by supplying wheelchair control via BCI. Spelling letters or words is done using BCI-driven devices, enabling impaired communication. One of the most well-known BCI paradigms is the P300 speller.

4. **Identification of mental state:** Research in this field focuses on the identification of mental states that are relevant to the assessment of an operator's cognitive state, such as attention levels for the treatment of individuals with attention deficit disorder, workload, and weariness.

The BCI technology can be helpful to healthy persons even if it was primarily created with impaired people in mind. Because it is non-invasive, portable, has a good temporal precision of a few milliseconds, and is very inexpensive, EEG is especially well suited for this function. Consequently, the following are some non-medical uses for BCI:

1. **Gaming:** Every BCI paradigm has been utilized for gaming. While the game is mostly handled by conventional methods, BCI is used to conduct some user actions and/or as an additional channel for in-game communication.

2. **Virtual Reality:** The majority of currently available works concentrate on moving around in the virtual environment or adjusting the virtual camera. Camera rotation commands can be performed by the user and the keyboard is used to issue other game commands.

2 Objective

The goal of this project is to develop and implement an SSVEP-Based System to control a cursor in a virtual online game to move in three directions: forwards, left and right.
This project is divided into three main parts:

1. **Signal Acquisition:** Use of a recording device to extract the EEG signal of the subject for offline and online processing.

2. **Signal Processing and Model Training:** Pre-processing of the signal, epoching of data, implementation of classification algorithm and training of the model.

3. **Real Time Implementation:** Integrating the signal acquisition and signal processing modules to record data and classify it in real time in real time and send control commands to the virtual game.

3 Methods

3.1 Experimental Set-up

The system consisted of five parts: the acquisition system, a screen with a 60Hz refresh rate that displays the stimulation matrix, an additional screen in which the game is displayed and the control module that decodes the frequencies and sends the control commands.

The acquisition system was originally the Emotiv EPOC+ system [12] shown in Figure 2. This system has 14 saline-based electrodes and has wireless connectivity via Bluetooth. The electrodes of interest were those placed over the parietal and occipital lobes: O1, O2, P3, P4, Pz; due to the visual processing areas being located there.

Figure 2 Emotiv EPOC + recording device. Source: [12]

The stimulation matrix consisted of three squares that flickered at frequencies 12Hz, 15Hz and 20Hz and a "rest" state. Before each trial a cue appeared on the screen to indicate the stimulation square the subject should pay attention to.

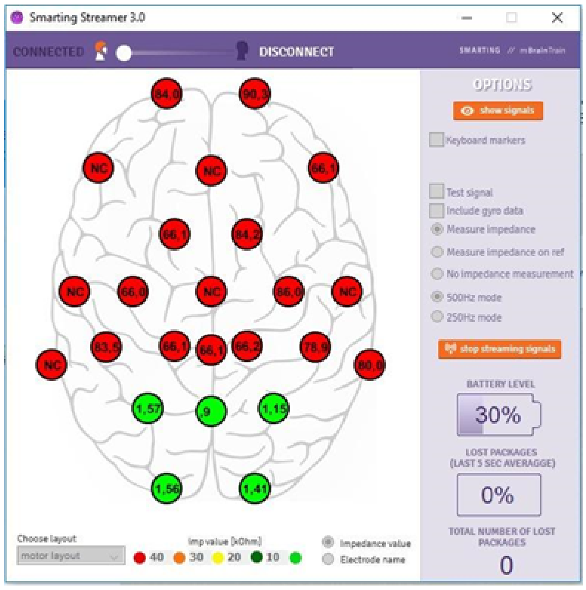
For the offline sessions carried out to train the classification algorithm the subject participated in sessions of between 5 and 10 minutes. In each session a trial consisted of 6 second of stimulation and 2 seconds of break between trials for gaze redirection.

Unfortunately, the implementation of our system using Emotiv EPOC was not possible and we had to change to the mobi Smarting 24 [13]. The Smarting 24 is an EEG cap with 24 wet electrodes and the possibility to choose between a sampling frequency of 250 Hz and 500 Hz. For our system we chose 250 Hz, as it has a more robust connection than the 500 Hz option. We used the same five electrodes as with the Emotiv system. Figure 3 shows the used cap and the user interface with which we checked the impedance of the electrodes was low enough to make a clean recording.

3.2 OpenVibe

The first approach to implement the described system was made using Open Vibe. OpenVibe is a software platform used to implement BCI systems. It includes a Signal Acquisition and a Designer Tool that allow to record data and create processing pipelines.

(a) mobi Smarting 24 EEG cap. Source: [13].



(b) Electrode Impedances.

Figure 3 EEG Cap mobi Smarting 24 and electrode impedances. 3a. The EEG cap used for the recordings. 3b. The value of the impedances of each of the used electrodes (in green).

The Signal Acquisition Tool was used to record EEG signal using the Emotiv EPOC device. The OpenVibe Designer Tool allows to create different scenarios to set up carry out the different processing steps. The stimulation scenario created consisted of the described stimulation matrix.
The OpenVibe Designer tool allows us to create different scenario for each of the processing steps. There are five main scenarios:

1. **SSVEP Configuration:** This scenario sets the parameters for the stimulation matrix and the acquisition of the signal. We use this scenario to set the three stimulation frequencies (12, 15 and 20 Hz).

2. **Training Acquisiton:** This scenario sets the parameters for the training session: the sequence of cues, the duration of the trials, the delay between trials, and the size and position of the stimulation squares in the stimulation matrix. It also saves the recorded signal in an `ov` file.

3. **CSP Training:** This scenario takes the recorded signal and filters the signal into the three stimulation frequency bands, it also performs target separation to form three classes: f=12Hz, f=15Hz and f=20Hz. The signal, separated into the three classes, is divided into epochs of 5 seconds and CSP is performed on this data to obtain the spatial filters that will transform the data at each of the frequencies to maxmize the variance between the target frequency and the rest of the data.

4. **Classifier Training:** After applying CSP the data is reduced to two channels that maximize the variance of the data to distinguish between classes. This variance is the feature that is then used to classify the data using Linear Discriminant Analysis.

The paradigm described in subsection 3.1 is created using OpenVibe as shown in Figure 4. The OpenVibe Designer user interface allows us to adjust the settings of the stimulation scenario and the training session. The sequence of targets can be defined as well as the times for stimulation and rest. Figure 5 shows the resulting stimulation matrix with the three target stimulation squares (12, 15 and 20Hz) in red and the cue in yellow.

Figure 6 shows the OpenVibe Designer scenario for filtering and epoching of data with the goal of generating CSP filters as described above. Figure 7 shows the recording done using the OpenVibe Signal Acquisition Tool and the Training Acquisition scenario. Finally, Figure 8 shows the data after applying a bandpass filter for the 20Hz frequency. We can notably see the influence of the 50Hz powerlines interference in the raw recordings,

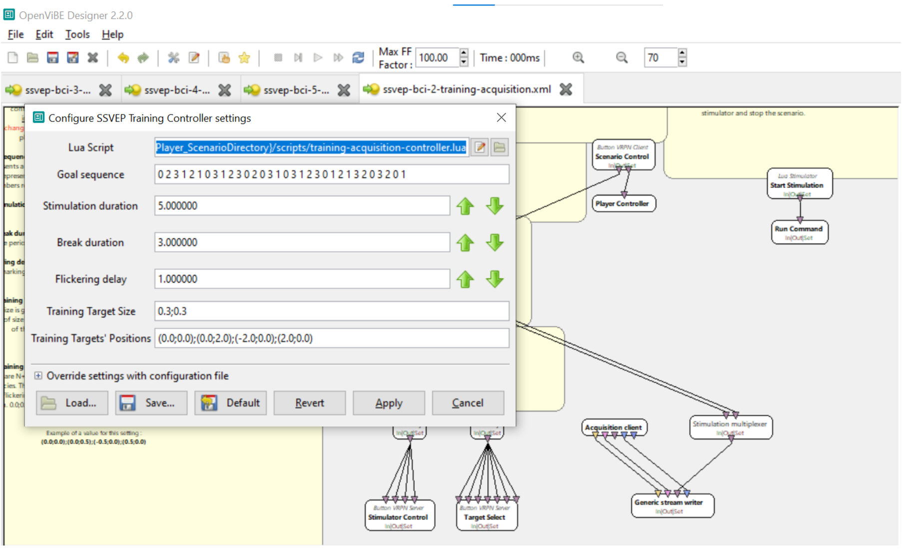


Figure 4 OpenVibe Designer: Stimulation Scenario Configuration

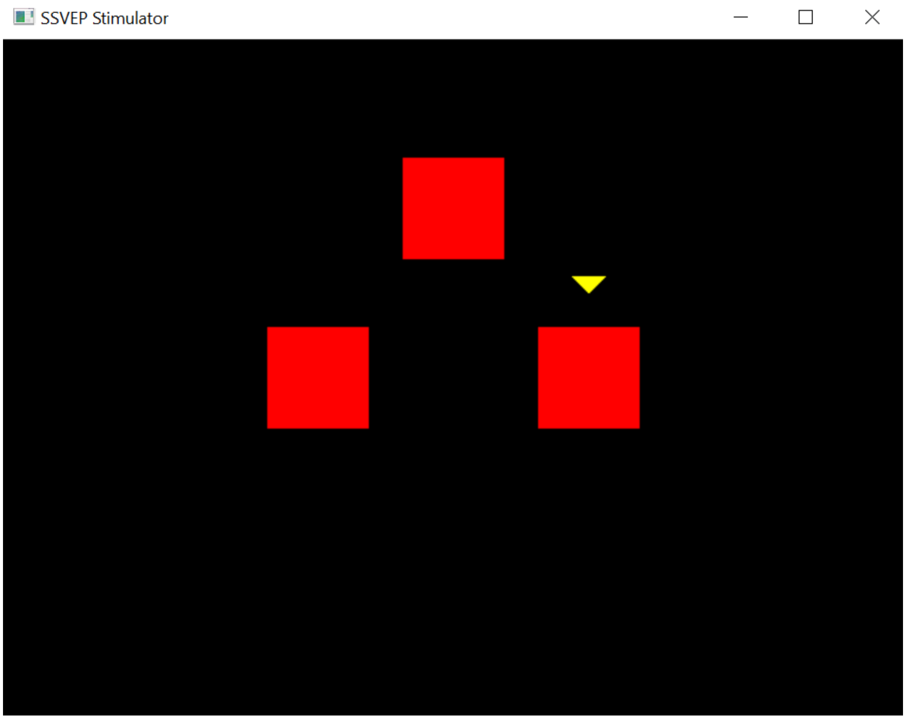


Figure 5 OpenVibe: Stimulation matrix. Three target stimulation square shown in red. Yellow triangle serves as the cue to indicate the trial target.

which is then removed in the filtered signals.

Unfortunately, we were not able to completely implement the pipeline described above and a result we tried to connect the OpenVibe Acquisition system with our own processing script on Python. However, the labels indicating the start and end of the experiment and the trials as well as the labels of each type of stimulation were not being sent correctly.
To circumvent this issue we ended up implementing the whole system on Python with custom-made scripts. This process is described in the following sections.

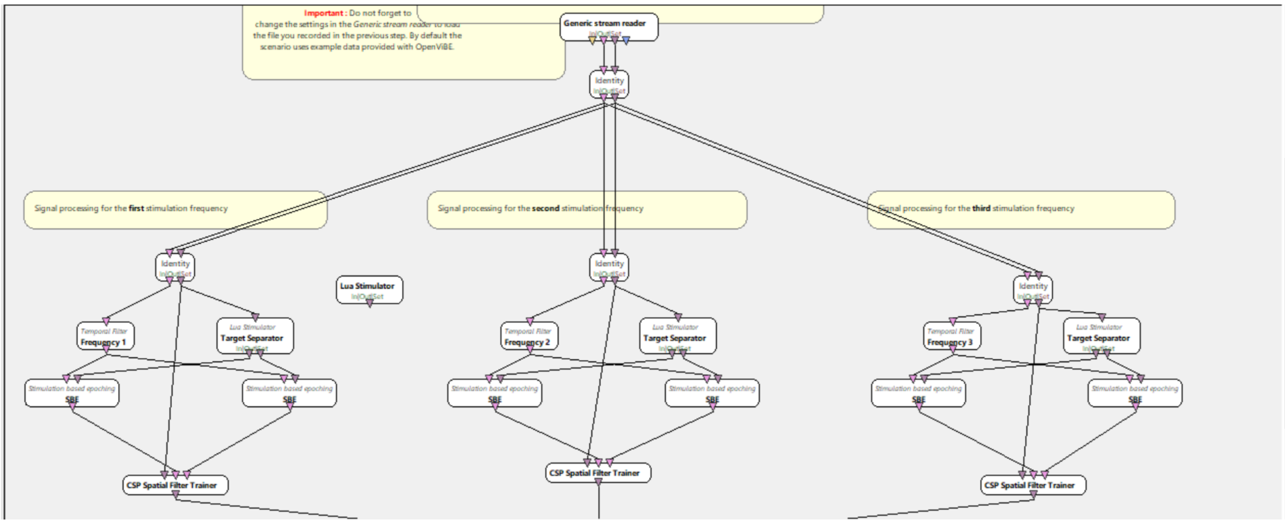


Figure 6 OpenVibe Designer: CSP Training

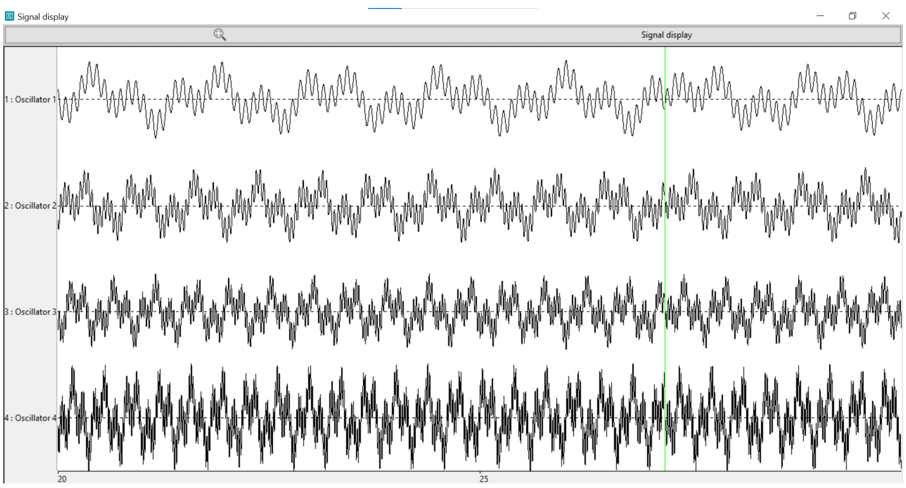


Figure 7 Open Vibe: Raw recordings.

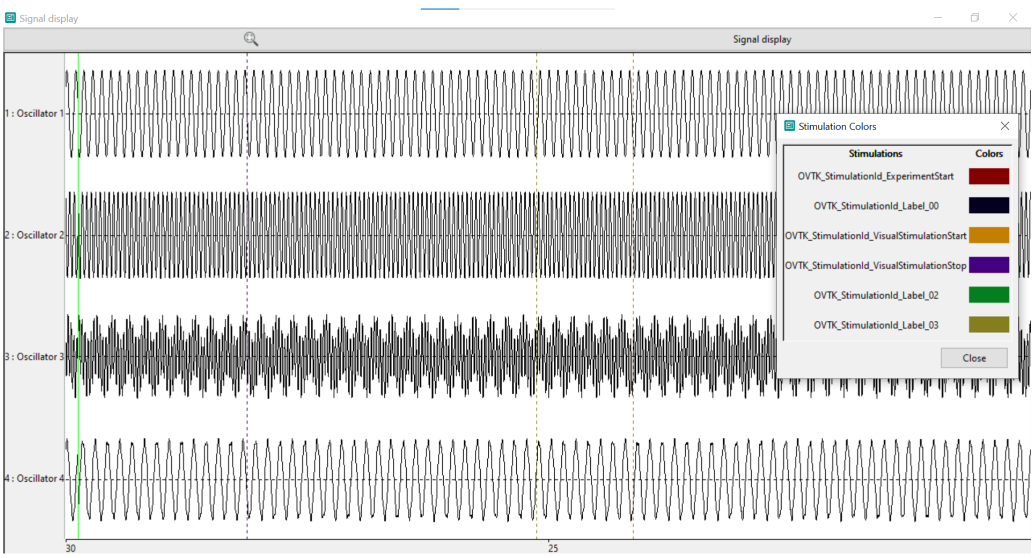


Figure 8 OpenVibe: Filtered data.

3.3 Zumobot

Initially, the goal of the project was to control the Polulu Zumo Robot [14], which is an Arduino controllable trackbot with two motors. The trackbot has an Arduino Uno Wifi micro-controller, which allows to send commands via WiFi (see Figure 9).

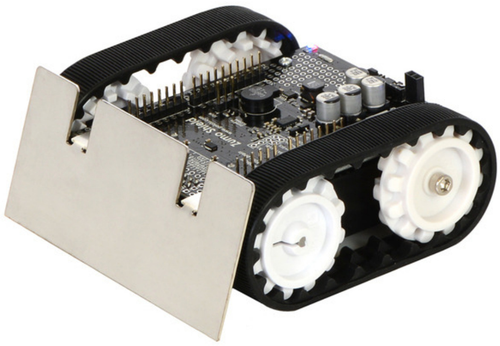


Figure 9 Polulu Zumo Robot controlled via Arduino. Source: [14].

In theory, the WiFi module can be configured to connect the Arduino and a computer to the same LAN network and send the commands through it. However, in practice we were restricted by the high traffic of the ICS laboratory network and its configuration, which resulted in a unidirectional communication channel and prevented us from successfully sending the control commands. Ideally, we would have been able to send the results of the classification algorithm as commands to the two motors in the trackbot to make it move in the desired direction.

Due to this issue with the WiFi we exchanged the final application of the BCI from the Zumo Trackbot to a virtual PyGame consisting of a maze in which the user has to navigate a virtual course to reach the finish line. The game is implemented on python and Figure 10 shows the user interface of the game. The white square is the block the user controls and the red line is the finish line. The green lines define the limits of the maze course and the goal of the game is to reach the finish line in less than 40 movements, meaning in our BCI implementation we require an accuracy high enough to complete the game in less than 40 trials. The script provided was modified to take the results of the classifier as inputs instead of the keyboards "A", "W" and "D".

3.4 Signal Processing Pipeline

Figure 11 shows the signal processing steps applied to the data in this project. The EEG data is acquired from the subject using the explained set-up. The raw EEG consists of the artifacts and is given to the filtering block. In the filtering block the notch filter and bandpass filter is applied. After filtering the EEG signals the unused channels are dropped (only 5 channels used) and Common Average Referencing is applied. The next block segments the data into epochs of 5s length. The CCA and PSD features are then calculated on the epoched data for 1 channel. A dimensionality reduction block is used to select the best components. After this the classification of the data is done. Each of the blocks is further explained in the next sections.

3.4.1 Pre-processing

The continuous EEG data was first band-pass filtered between 5-45 Hz. Then, notch filter of $f = 50Hz$ and $f = 100Hz$ was applied to remove the electric hum caused by the power line.

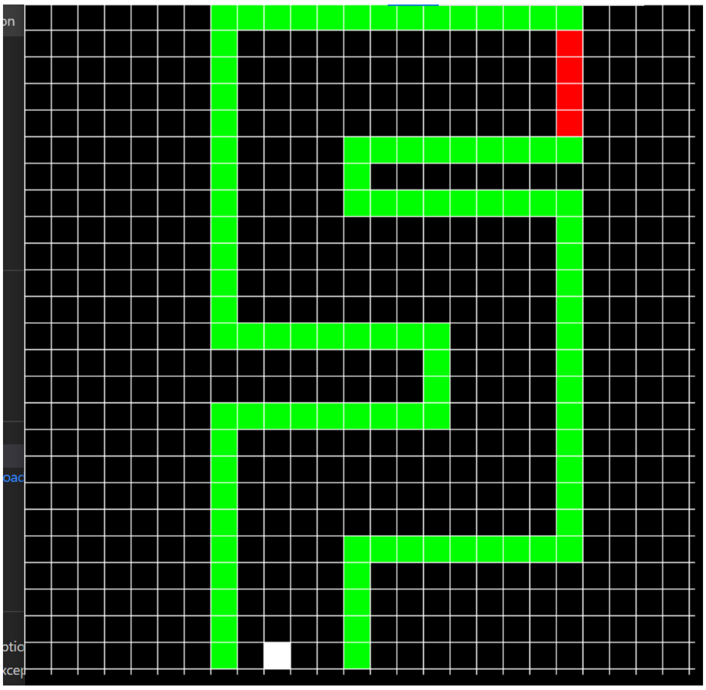


Figure 10 Virtual Maze PyGame. The white square has to be navigated thorugh the green maze until it arrives to the red finish line.

Afterwards, the channels that were not used were dropped, and we kept the channels O1, O2, P3, P4 and Pz. Once the unused channels were dropped we applied common average referencing (CAR) to the remaining channels. The spectrum of the raw EEG data can be seen in Figure 12, and the cleaned data can be seen in Figure 13.

3.4.2 Feature Extraction

Three main feature extraction methods were explored in order to achieve the optimal results:

1. **Common Spatial Patterns (CSP):** This method provides a series of spatial filters that transform the data to maximize the variance of one class and minimize the variance of the other. In this case a CSP filter is produced for each stimulation frequency which when applied to the data maximizes the variance of the data in one class (stimulation frequency) and minimizes it in the remaining classes. After applying the algorithm the variance is used as a feature.

2. **Power Spectral Density (PSD):** The PSD measures the power content of the signal at each frequency. In the case of SSVEPs we see an increased power at the frequency at which the SSVP was elicited. Therefore the maximum amplitude of the PSD of a trial can be used as a feature for classification. To increase the robustness of this method the second harmonic can also be taken into consideration.

3. **Canonical Component Analysis (CCA):** This method compares two sets of signals X and Y and finds the linear transformations that transforms the sets of signals to maximize the correlation. In this case X is the matrix composed of the recorded multichannel SSVEP epochs and Y is the reference matrix signal, made up of cosines and sines at the stimulation frequencies and their second harmonics. The correlation coefficients obtained after the CCA transformation are used as the features for classification.

Alongside, Principal Component Analysis (PCA) was used as a dimensionality reduction tool. PCA calculates the correlation between features and maximizes the variance between them, helping us choose the most discriminating components.

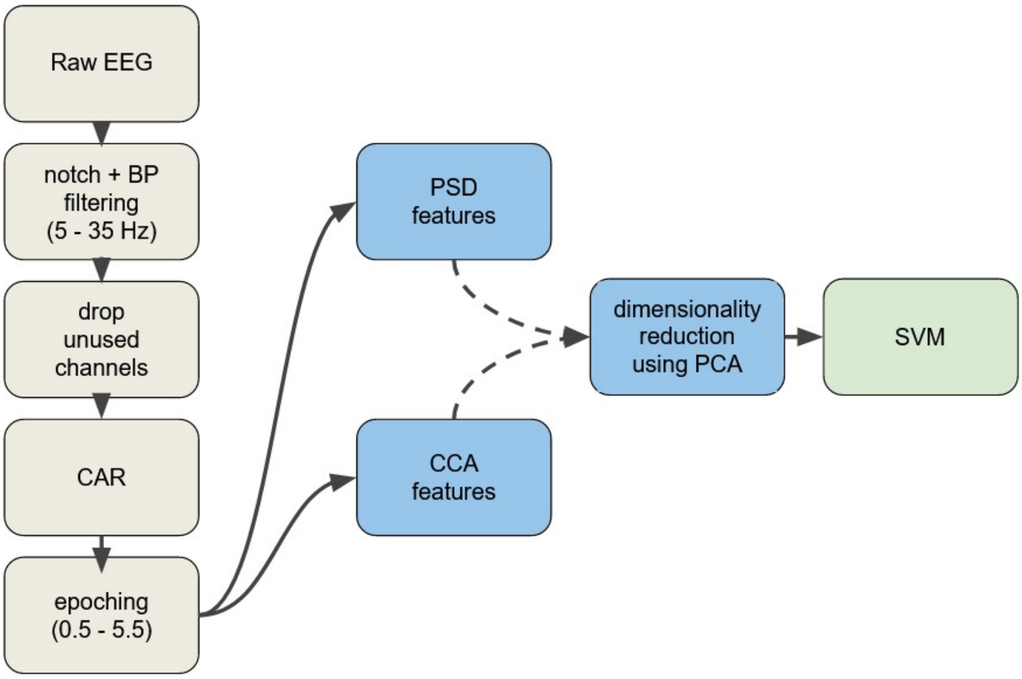


Figure 11 Signal Processing Pipeline

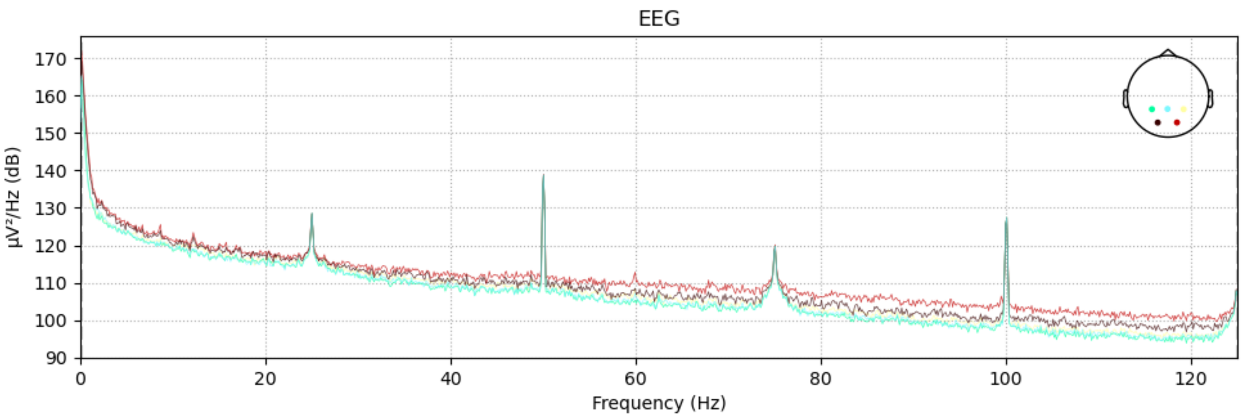


Figure 12 Raw data in the frequency domain. We can clearly see the interference of the power lines at $f = 50Hz$ and its harmonics.

Another alternative was to use the Fisher score. However, for the first few recordings, PCA performed slightly better than the Fisher score. A possible explanation for this could be that the Fisher score returns the features that help the most with the classification between different classes. But, for example, the Fisher score could select the PSD value at 9 Hz at electrode O2 as a relevant feature for the classification.

However, it is possible that the electrode O2 might slightly move during the recording, maybe due to deep breathing of the participant. This would mean that the PSD value at 9 Hz for electrode O2 would not be a reliable feature anymore for any epoch after the electrode moved.

With this possible explanation in mind, it was assumed that the Fisher score could be a powerful tool for stationary data sets where the features are not prone to change or noise. However, for our current dynamic and noisy setting, we opted for PCA.

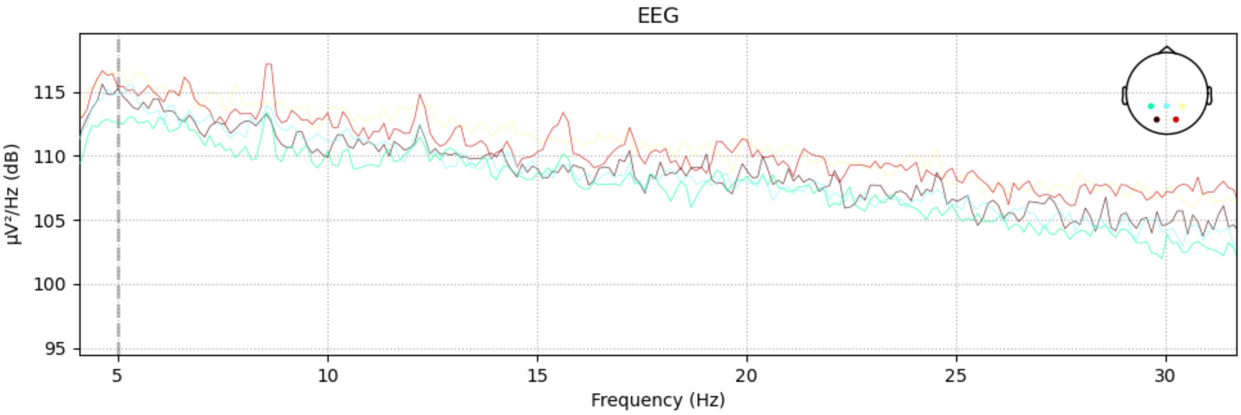


Figure 13 EEG data after bandpass and notch filtering. We can see that the components at 50Hz and its harmonics are removed and instead we can appreciate peaks at 9, 13 and 17Hz.

As discussed before, PCA projects the data matrix to a space where the variances are maximized for each direction. By using less components, this projection simplifies the data set, however it also focuses on its global tendency. Thus, it was assumed that this method would help combat overfitting, which will be the main problem of our classification pipeline in the next steps.

3.4.3 Classification

Two classifiers were used: Linear Discriminant Analysis (LDA) and Supported Vector Machines (SVM).

1. **LDA**
   Linear Discriminant Analysis (LDA) is a supervised technique to perform dimensionality reduction; it can also be applied to perform classification. This approach projects the data in a new dimension in such a way that points belonging to different classes are linearly separable. In the case of this project, there are three classes corresponding to each frequency stimulation. We used the LDA implementation of the sklearn library.

2. **SVM**
   In this work, we used the sklearn implementation of support vector classification; this function is called svc. SVC works by creating a boundary to separate samples of different classes. The C parameter allows more flexibility in the placing of the classification boundary. A large C value will create small margins on the classification boundary, meaning that the classifier will perform better on training data, but it could over-fit. A small value of C will create large margins on the classification boundary, meaning that the classifier will perform worst on training data; however, it could generalize well to test data. Therefore, there is a trade-off in choosing a C-value.

3.5 Python Implementation

3.5.1 Offline Sessions

Since OpenVibe was not used for stimulation presentation, a Pygame based script was written for SSVEP stimulation presentation. This script followed OpenVibe's stimulation window where the stimulation frequencies, stimulation box sizes and colors could be defined by the user.

Since the major problem with OpenVibe's implementation was the wrongly sent LSL time markers, a special care was given to send correct time markers. The design philosophy was to send rather more markers than less, because the unnecessary markers could be discarded in the offline analysis script. So, whenever an event happened on the screen, like start of a new trial, showing the cue or entering the pause period, an LSL marker was sent.

Fortunately, Smarting's LSL connection worked out of the box. By starting the Smarting Streamer program and then the data stream, an LSL stream for the EEG data was created.

The last step was to record both the data stream and the marker stream through LabRecorder, which synchronizes both streams in time. After the recording ends, LabRecorder writes the data into an XDF file, which can be read in Python using the pyxdf library.

3.5.2 Online Sessions

For the online session, first, a classifier had to be trained. As EEG data is non-stationary and change from session to session, an offline analysis block had to be run to calibrate the classifier with the data of the current setup. Then, the classifier could be saved for later use.

For the transfer function mapping the classifier predictions to control outputs, the task at hand was taken into account. In the maze game, the most common command was 'UP', then 'RIGHT' and lastly 'LEFT'. This is why the most predicted class was mapped to 'UP' and the least predicted class to 'LEFT'.

This choice of the transfer function squeezed in some more performance to our system. But it should be noted that, this would not be a reliable way to choose the transfer function in a setting where all the commands have the same importance.

For the stimulation another Pygame script was written. Again, for each trial, three stimulation boxes were shown to the participant. However this time, there was no cue to guide that participant and the blinking of the boxes stopped after 6 seconds. This was the time point, where an LSL marker was sent to the real-time processing script to signal that a prediction has to be made.

The data from this period of time was extracted and processed using the same signal processing pipeline described in subsection 3.4 to decode the target the user was focused on. In the final implementation, a three-trial majority vote was used to determine each command signal.

On the data acquisition side, another custom script had to be written to access the real-time EEG data stream. For this purpose, the pylsl library was used and available EEG samples were pulled in each iteration of an endless while loop.

Moreover, the script was waiting for the LSL marker, being sent by the Pygame script to initialize the prediction period. Once the 6s trial ended and the 'pred_start' marker was sent, the real-time analysis script the data last 6s data was used to extract the features and make a prediction with the pre-calibrated classifier.

4 Results

Multiple data sets were recorded from a single participant over a span of a few days in order to evaluate this system. Figure 14 shows that, in general, training and validation accuracy increase when we use more features.

Based on the same image, we claim that the best option is to use eleven features because, at that point, the system gets the *elbow* state, in which the accuracy does not considerably increase.

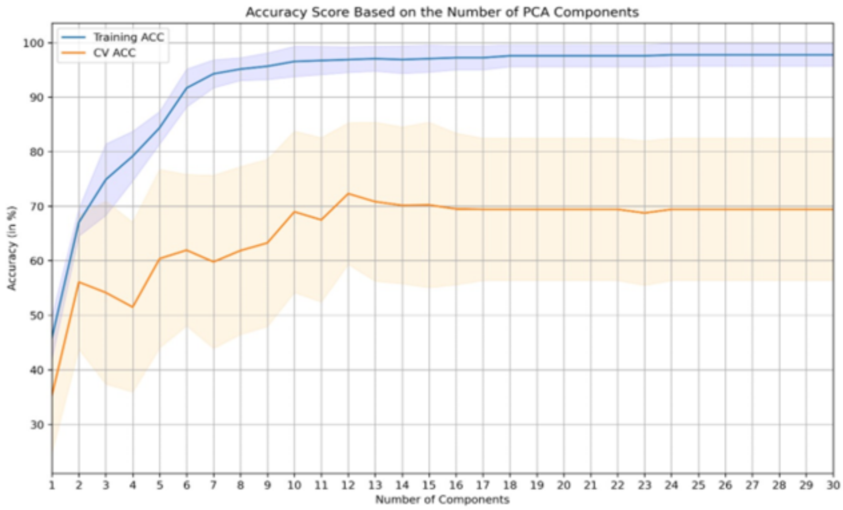


Figure 14 Training and testing accuracy for different numbers of PCA components. As the number of components increase, the training and validation set accuracies tend to increase. However, after some time, the model starts overfitting. To find a good trade-off between the training and validation errors, the "elbow" of the graph was used to get the optimal number of PCA components.

Figure 15 shows the confusion matrices for training and testing sets; we can see that in the training set, the amount of misclassification is minimal, with only two elements.

Nevertheless, for the test set, the classifier makes more mistakes; the worst class is 17Hz because, in this class, the classifier assigned erroneous labels two times. The confusion matrix illustrates that the classifier does not generalize properly; this could be because of a lack of training data or very noisy samples.

Table 1 shows the model performances over different recording sessions and participants. It can be seen that for all recordings, the training accuracy was above 85%. However, the test set accuracy and the validation set accuracy that approximates that value differ from recording to recording. For some recordings the test set accuracy was above 80%.

Furthermore, it can be seen that different stimulation frequencies were tested out for different recordings. Overall, there was no clear mapping between the chosen stimulation frequencies and a good/bad classifier performance.

As it can be seen form recordings 1.1, 1.2 and 1.4, the stimulation frequencies were the same. However, the test set accuracies changed dramatically between the sessions.

Changing the stimulation frequencies to 12, 15 and 20 Hz at recording 1.3 also resulted in a high test set accuracy, matching the performance at recording 1.2. However, it dropped again for recording 1.5.

The main takeaway from this table is that the classifier performances highly depended on the current data. Each recording was slightly different, with the participant motivation and concentration and EEG signal quality changing between sessions.

However, in the end, the pipeline consistently performed above chance level, which was 33.33% for a three-class classification problem.

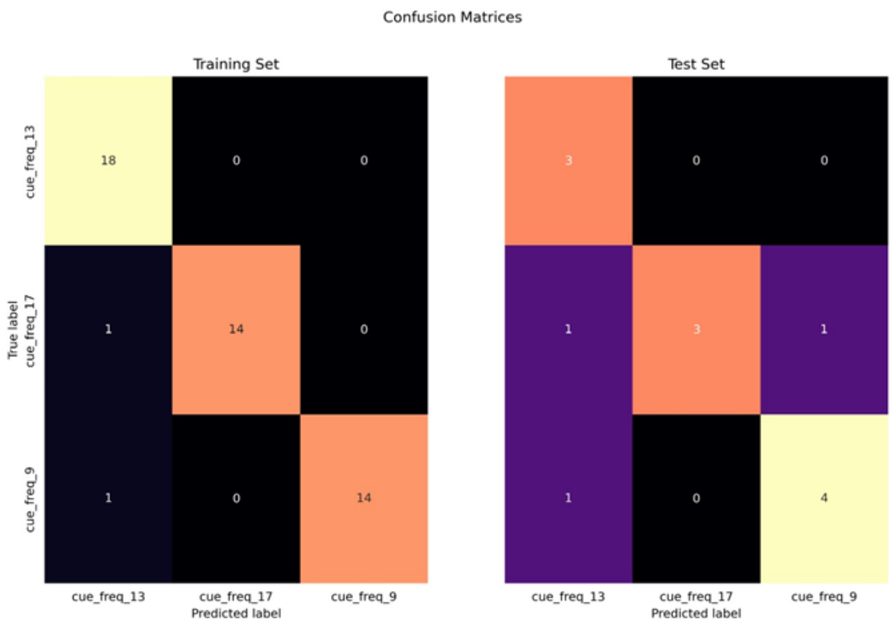


Figure 15 Confusion matrices for training and test sets

Table 1 Classification performances for each recording. The "Recording" column follows the convention of subject_number.recording_number. It can be seen that for all recordings, the training accuracy was above 85%. However, the test set accuracy and the validation set accuracy that approximates that value differ from recording to recording. For some recordings the test set accuracy was above 80%.

Recording	Stimulation Frequencies	Validation Accuracy	Training Accuracy	Test Accuracy
1.1	9 Hz, 12 Hz, 15 Hz	0.66 ± 0.10	0.92	0.58
1.2	9 Hz, 12 Hz, 15 Hz	0.62 ± 0.15	0.96	0.83
1.3	12 Hz, 15 Hz, 20 Hz	0.57 ± 0.13	0.92	0.83
1.4	9 Hz, 12 Hz, 15 Hz	0.43 ± 0.13	0.92	0.50
1.5	12 Hz, 15 Hz, 20 Hz	0.39 ± 0.13	0.88	0.50
1.6	12 Hz, 15 Hz, 20 Hz	0.39 ± 0.21	0.92	0.42
1.7	9 Hz, 13 Hz, 17 Hz	0.64 ± 0.076	0.90	0.70
1.8	9 Hz, 12 Hz, 15 Hz	0.63 ± 0.15	0.85	0.58
2.1	9 Hz, 13 Hz, 17 Hz	0.64 ± 0.076	0.90	0.70

5 Discussion

During the development and implementation of this project we encountered several limitations that hindered our progress. The first issues we encountered were related to the implementation with OpenVibe and the Emotiv EPOC recording System. In our first approach we intended to implement the system completely in OpenVibe using the OpenVibe SSVEP template pipeline changing it to adapt it to our paradigm as needed.

In a second step, we would compare the performance of the CSP+LDA feature extraction and classification method to another method like PSD or CCA. However, since we were not able to successfully make use of OpenVibe this approach was not fully implemented in the end. In regards to the signal acquisition and recording system, we found Emotiv to be an easy to use device but with worse electrode impedance. We also struggled to place the electrodes at the correct locations and maintain them there during the whole session due to the fact that the device was not fixed and upon slight user movements the electrodes moved. Additionally, in between sessions, it was hard to ensure that the electrodes were systematically placed on the same location.

Instead, we proceeded to implement the whole system in Python. However, this approach came with its own limitations due to the challenge posed by using custom-made scripts instead of a pre-established framework. The main limitations were a lack of training data, inconsistent flickering in the stimulation matrix, overfitting of the model and the components not coming together.

In order to have a good model with high accuracy it is necessary to have more training data, which we lacked mostly due to visual fatigue. Ideally, we would have 20 minute sessions with at least 40 trials per class in order to have enough data to train our classifier and also do 10-fold cross-validation and decrease the possibility of overfitting. However, the users got visual fatigue after around 10 minutes, making it impossible to have longer offline sessions.

Another issue we encountered was the fact that the signals were very unstable. Signals recorded in different days or even in different sessions during the same day often varied a lot, with the data from some sessions being non-usable due to bad quality and non-presence of SSVEPs. In addition, the fact that signals changed between sessions meant that calibration data was needed before every online session, resulting in additional recordings and time.

One of the reasons why the signal might have been more unstable than normal is the inconsistent flickering of the stimulation matrix. We experienced some issues when implementing the stimulation matrix, with some of the squares sometimes freezing momentarily. The resulting inconsistent flickering of the screen might have introduced additional artefacts and affected the SSVEPs.

The limitations discussed above prevented us from being able to complete the maze in an online session due to the model not having a high enough accuracy for all classes, meaning that it often did not classify the "left" trials correctly. Nevertheless, despite not achieving the ultimate goal of this project we were able to correctly implement the whole system both offline and online on Python with minimal latency and a good connection between the acquisition system and the processing pipeline. The system worked correctly and the principles of the system were correct, the reason we did not manage to complete the online game was due to lack of data and a bad classifier and not due to incorrect use of labels, feature extraction methods or coding issues.

5.1 Future Work

This project has several future work lines. Firstly, the BCI paradigm could be modified and gamified paradigm. In the current implementation, the stimulation matrix consists of just flashing squares at different frequencies, which after a short period of time becomes too repetitive and fatiguing for the user, which struggles to maintain focus and finish the session. A gamified paradigm could lead to a higher engagement of the user, the possibility of increasing the duration of the training sessions and overall an improved quality of the data.

Here, making the stimulation of the boxes more robust would also be very valuable. Even with the current setup, we saw clear spectral peaks at expected locations. However, depending on the stimulation computer being used, the flickering was more or less reliable. Making this work for any computer, regardless of the computer's processing power and speed, would be extremely helpful.

Another part of the system that could be improved is the online implementation. In the current implementation the user has a pre-allocated amount of time to fix their gaze on one of the targets. The data is processed in real time, but the trial periods are pre-established. An improvement would be to implement a "free recall" paradigm in which there is a continuous flashing of the targets and streaming of data for classification.

A major issue was the slight complexity of the current setup. For the offline analysis, three different programs and two different Python scripts had to be used. On the other hand, for the online setting, Two different programs and four different Python scripts were actively used.

Managing all these programs and scripts became unintuitive after some time, especially as recording sessions got longer. Lastly, to change some parameters in one script, like the stimulation frequencies, was not enough. The same frequencies had to be changed in all the other scripts. This could also add some overhead and make fast prototyping and testing difficult.

To solve all these problems, the whole pipeline could be automated. This would give us the flexibility to quickly change parameters like the stimulation frequencies and filtering coefficients. Also it can help with easily connecting all the different components together.

Finally, the feature extraction and methods could be improved. For instance, the CCA algorithm could be extended to include more harmonics of the fundamental frequency, And all methods could be implemented to be multichannel instead of selecting the best channel manually.

6 Acknowledgements

We would like to express our gratitude first to our lecturer Alireza Malekmohammadi, who provided support during the development and implementation of this project.
We would also like to express our gratitude to the Institute of Cognitive Systems (ICS) and to Professor Gordon Cheng for the opportunity to take part in this course and to develop this project.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control", *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, Jun. 2002, ISSN: 1388-2457. DOI: 10.1016/s1388-2457(02)00057-3.

[2] Q. Gao, Y. Zhang, Z. Wang, E. Dong, X. Song, and Y. Song, "Channel projection-based CCA target identification method for an SSVEP-based BCI system of quadrotor helicopter control", *Computational Intelligence and Neuroscience*, vol. 2019, p. 2 361 282, 2019, ISSN: 1687-5273. DOI: 10.1155/2019/2361282.

[3] F. Cincotti, D. Mattia, F. Aloise, S. Bufalari, G. Schalk, G. Oriolo, A. Cherubini, M. G. Marciani, and F. Babiloni, "Non-invasive brain-computer interface system: Towards its application as assistive technology", *Brain Research Bulletin*, vol. 75, no. 6, pp. 796–803, Apr. 15, 2008, ISSN: 0361-9230. DOI: 10.1016/j.brainresbull.2008.01.007.

[4] I. Martišius and R. Damaševičius, "A prototype SSVEP based real time BCI gaming system", *Computational Intelligence and Neuroscience*, vol. 2016, p. 3 861 425, 2016, ISSN: 1687-5273. DOI: 10.1155/2016/3861425.

[5] V. Raudonis, G. Narvydas, and R. Simutis, "A classification of flash evoked potentials based on artificial neural network", *Elektronika ir Elektrotechnika*, vol. 81, no. 1, pp. 31–36, Mar. 26, 2015, ISSN: 2029-5731. DOI: 10.5755/j01.eee.81.1.11029. [Online]. Available: http://158.129.0.15/index.php/elt/article/view/11029 (visited on 08/05/2022).

[6] G. Rance, F. W. Rickards, L. T. Cohen, S. De Vidi, and G. M. Clark, "The automated prediction of hearing thresholds in sleeping subjects using auditory steady-state evoked potentials", *Ear and Hearing*, vol. 16, no. 5, pp. 499–507, Oct. 1995, ISSN: 0196-0202. DOI: 10.1097/00003446-199510000-00006.

[7] D. Regan, "Steady-state evoked potentials", *Journal of the Optical Society of America*, vol. 67, no. 11, pp. 1475–1489, Nov. 1977, ISSN: 0030-3941. DOI: 10.1364/josa.67.001475.

[8] G. Pfurtscheller and C. Neuper, “Motor imagery activates primary sensorimotor area in humans”, *Neuroscience Letters*, vol. 239, no. 2, pp. 65–68, Dec. 19, 1997, ISSN: 0304-3940. DOI: 10.1016/s0304-3940(97)00889-6.

[9] V. Morash, O. Bai, S. Furlani, P. Lin, and M. Hallett, “Classifying EEG signals preceding right hand, left hand, tongue, and right foot movements and motor imageries”, *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 119, no. 11, pp. 2570–2578, Nov. 2008, ISSN: 1388-2457. DOI: 10.1016/j.clinph.2008.08.013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2602863/ (visited on 03/18/2021).

[10] G. Pfurtscheller, T. Solis-Escalante, R. Ortner, P. Linortner, and G. R. Müller-Putz, “Self-paced operation of an SSVEP-based orthosis with and without an imagery-based "brain switch:"a feasibility study towards a hybrid BCI”, *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 18, no. 4, pp. 409–414, Aug. 2010, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2010.2040837.

[11] F. Lotte, L. Bougrain, and M. Clerc, “Electroencephalography (eeg)-based brain-computer interfaces”, *Wiley Encyclopedia of Electrical and Electronics Engineering*, Sep. 2015. DOI: 10.1002/047134608X.W8278.

[12] Emotiv. (2022). EMOTIV EPOC x | 14 channel mobile EEG headset, EMOTIV, [Online]. Available: https://www.emotiv.com/epoc-x/ (visited on 07/24/2022).

[13] mBrain Train. (Dec. 11, 2020). Mobile EEG - smarting mobi | mBrainTrain, [Online]. Available: https://mbraintrain.com/smarting-mobi/ (visited on 08/04/2022).

[14] Polulu. (2022). Pololu - zumo robot for arduino (assembled with 75:1 HP motors), Polulu Robotics & Electronics, [Online]. Available: https://www.pololu.com/product/2510 (visited on 07/25/2022).