**Media Engineering and Technology Faculty**
**German University in Cairo**

# A Brain-controlled On-screen Assistive Keyboard

**Bachelor Thesis - Expository**

Author: Maqarios Wagdy Tawadros Saleh Mohareb
Supervisors: Assoc. Prof. Seif Eldawlatly
Submission Date: 26 May, 2019

**Media Engineering and Technology Faculty**
**German University in Cairo**

# A Brain-controlled On-screen Assistive Keyboard

**Bachelor Thesis - Expository**

Author: Maqarios Wagdy Tawadros Saleh Mohareb
Supervisors: Assoc. Prof. Seif Eldawlatly
Submission Date: 26 May, 2019

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgement has been made in the text to all other material used

Maqarios Wagdy Tawadros Saleh Mohareb
26 May, 2019

# Acknowledgments

First of all, I would like to express my at-most gratitude to my supervisor Assoc. Prof. Seif Eldawlatly for his support and immense knowledge. His sincere guidance helped me a lot in the time of research, application, experimentation and writing of this thesis.

Besides my supervisor, I thank my colleagues Ahmed Alaa, Mostafa Nasr and Omar Ashraf for giving me insightful arguments and collection of data.

Last but not least, I would like to thank my parents Wagdy Tawadros Amal Fathy for their continuous support and patience.

# Abstract

In this project, an on-screen keyboard will be made. The main aim of this project is to help people with physical disability use computers more easily.

Using Peak 300 [1] (P300) based Brain–Computer Interface [2] (BCI) which uses oddball paradigm to determine the required actions to be done by the application. By displaying a matrix of characters, the user can determine the character he wants from this matrix.

By processing the recorded brain waves to Machine Learning [3] (ML) model and apply mathematical filters to accurately determine the intended actions. Thus, reaching an accuracy with an average of 80% in character recognition rate which is not bad compared to previous results obtained by previous papers [4, 5, 6, 7].

# Contents

# Chapter 1

# Introduction

Disabled people are those with impairment that can be mental, physical, intellectual, etc. However, the main target here are those with physical impairment (i.e. Spinal Cord Injury (SCI)), since their brain is functional to improve their productivity and support their independence.

The target of this project is to make on-screen keyboard using BCI instead of using keyboard or mouse. By showing items on screen and measuring brain signals, we can determine what the user wants. A BCI monitors brain activity using electrodes to detect certain patterns generated by the user's brain. After digitizing Electroencephalography [2] (EEG), signals are pre-processed and passed to ML to classify the outcome [6, 4, 7, 5].

EEG is a type of Event Related Potential (ERP) initially reported by S. Sutton [8]. P300 is the largest component of EEG and can be detected during an oddball event. An oddball event happens when the user is presented with sequence of events that can be classified based on frequency of appearance. The categories of this classification are either frequent or infrequent event. The result of stimulation happens 300ms after the event. It is worth noting that, infrequent events can only be detected if they are by surprise.

Based on Fazel-Rezai et all [9], P300 has proved to be one of the main components of BCI. Although it was almost abandoned since 1988 till 2000, in recent years, it started to gain momentum among its peers. P300 has several appealing features such as efficiency, straightforwardness, relatively fast and practically doesn't require training. P300 has proved that it can be used in wide range of functions and being easy to use in home for disabled people, although it has some concerns such as gaze shifting. However, new paradigms have been introduced as well as new ways of flashing and even improve classification methods [4] to enhance the overall experiment.

In order to identify the different brain patterns without human intervention, ML was used. By supplying the model with signals and their corresponding classes, we can identify brain patterns. The ML type we are going to use is a supervised one, which requires set of examples and their correct responses.

# Chapter 2

# Background

## 2.1 Introduction to BCI

The communication within the human body requires muscles and peripheral nerves. The start of the process happens with the user's intent which triggers a complex process where specific brain areas are triggered, and hence the peripheral nerves system sends the signals to the corresponding muscles, which performs the intended action. Efferent means sending signals from the peripheral nervous system to the muscles. However, Afferent describes the other way around. [2]

A BCI offers an alternative way to natural control and communication by bypassing the normal efferent pathways. BCI directly measures the brain waves corresponding to the user intent then translates it into signals which gets sent to the BCI applications. These translations include enhancement, noise removal, signal processing and pattern detection (classification). [2]

BCI, Brain-Machine Interface (BMI) and Direct Brain Interface (DBI) almost refers to the same thing which refer to intercepting (recording) brain waves. However, Neuroprosthesis is more general as it refers to receiving and sending signals from and to the brain. Figure 2.1 shows examples of neuroprostheses which shows that BCI is a category of neuroprostheses.

### 2.1.1 Processes of Operation

BCI goes through three main procedures, measuring the brain activity, processing it and controlling the intercepted signals.

#### Measuring Brain Activity

Brain activity produces magnetic and electrical waves. That's why, sensors or electrodes are used to measure these changes at different areas and times. Brain activities can be

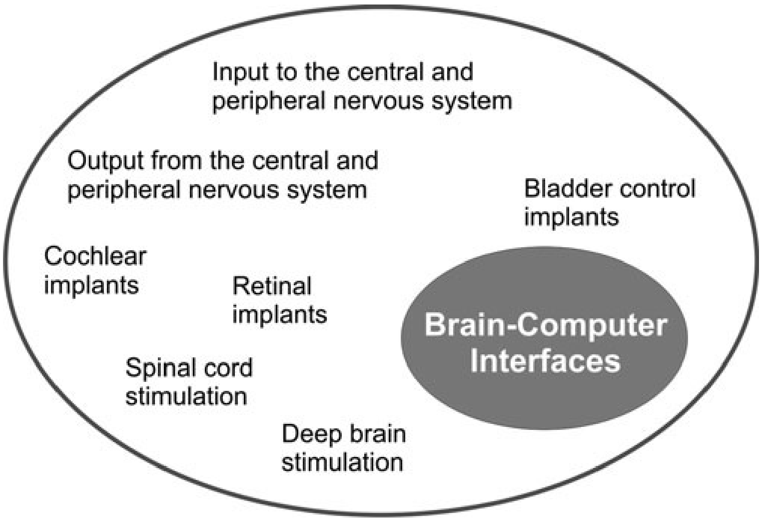



Figure 2.1: Examples of neuroprostheses and its relation to BCI [2]

recorded through sensors (non-surgical solution) or electrodes (surgical solutions). For the non-surgical solution, sensors are used to measure the electrical activity from the scalp and this solution is called EEG (Figure 2.2). They are relatively easy to deal with however they don't provide accurate measures due to external interference and are susceptible to limitations in frequency range. In order to get consistent recording, sensors are spread on the scalp through system called 10 - 20 (Figure 2.3). 10 - 20 refers to how sensors are spread 10 - 20 - 20 - 20 - 20 - 10 percent. The 6 regions are named according to their position: Fp - pre-frontal, F - frontal, C - central, P - parietal, O - occipital, T - temporal.

On the other hand, the surgical solution need to open the skull through surgical procedures and plant the electrodes. When the electrodes are planted on the surface of the cortex, it's called Electrocorticogram [2] (ECoG).

And another solution called Intracortical Recordings [2] (ICR), which plants the electrodes in the inner parts of the brain. Although the surgical solutions provide higher accuracy and frequency range compared to the non-surgical solution they have serious drawbacks such as they need surgery, finance and can have ethical problems. Difference between each solution is shown in Figure 2.4.

### 2.1.2 Brain Patterns

Because measuring brain activity is not enough, some mental strategies are used to trigger the required tasks. The most used mental strategies are selective attention and motor imagery. Since only selective attention is needed in this project, motor imagery will not be introduced.

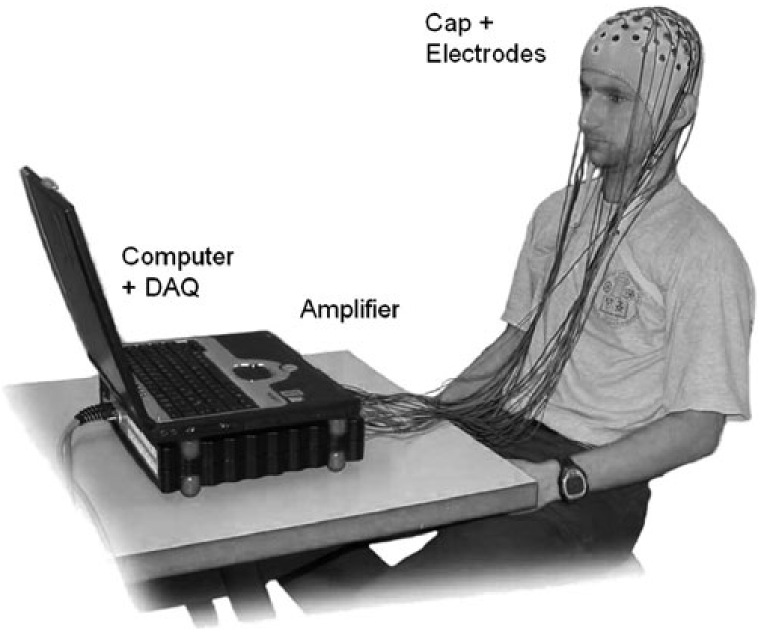



Figure 2.2: EEG based BCI [2]

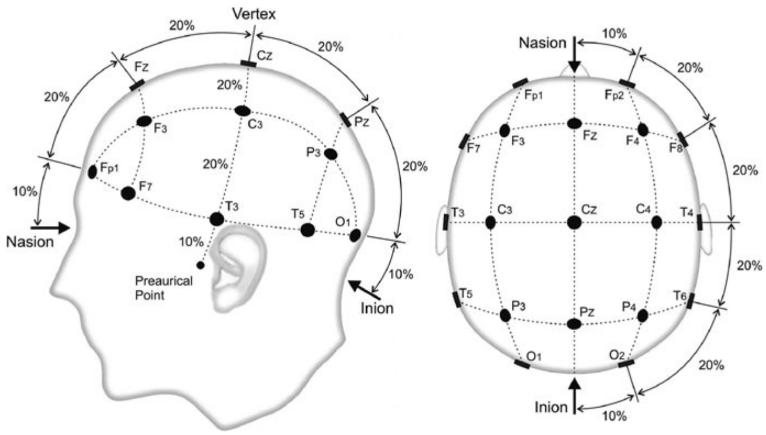



Figure 2.3: International standard of representing 10 - 20 system that shows how electrodes are spread on the scalp [2]

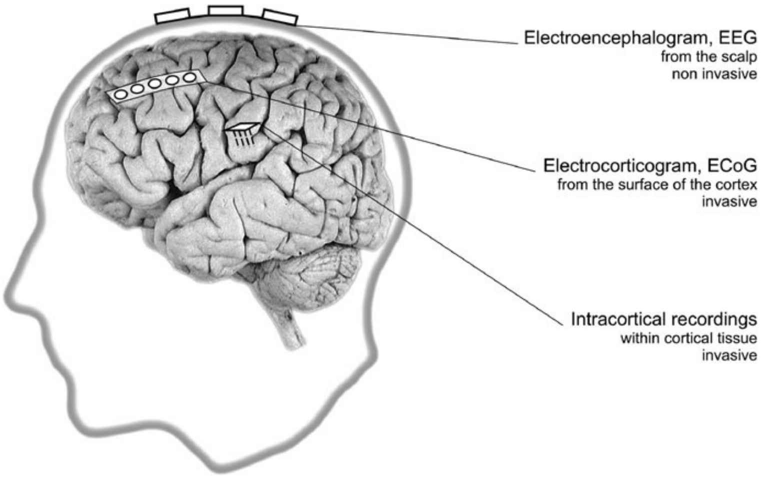



Figure 2.4: The surgical and non-surgical solutions of brain waves' detection [2]

### Selective Attention

BCI based on selective attention has to depend on external stimuli (event), either visual or auditory. In BCI application, in order for the user to select certain command, the user has to focus (select) their attention on a certain item on the screen to execute that command.

P300 based BCI, the main component behind this project, relies on visual stimuli to execute different tasks. As described in Chapter 1, P300 is an oddball paradigm that requires two kind of events, frequent and infrequent event. However, infrequent events must be by surprise to show the most effect (Figure 2.5).

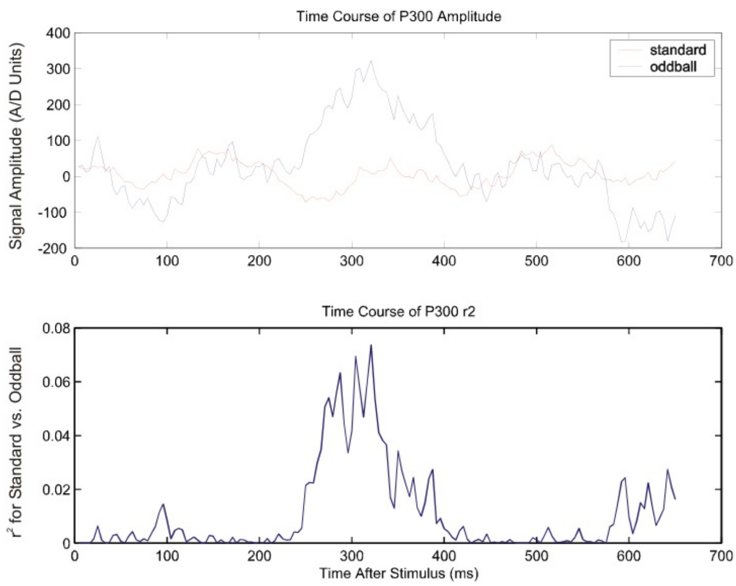



Figure 2.5: P300 Signal: Representation of an oddball paradigm [6].

Steady State Visually Evoked Potential [10] (SSVEP) based BCI is similar to P300 as it depends on visual stimuli. SSVEP requires the screen to flicker with frequencies 6 - 30 Hz to execute different commands. Focusing one's attention on a certain area with certain flickering frequency, then identifying that frequency to execute the wanted task.

# Chapter 3

# Approach

## 3.1 Data Description

In this project, BCI2000, Competition III, Dataset II was used, due to being accurate, free, provides 64 channels and quite a number of papers was based on it which provides a good reference for comparison of classifiers and the corresponding accuracy. In addition to the competition's dataset, we recorded our own dataset from 2 subjects. In total, the competition's dataset has 2 subjects A and B, the recorded dataset has another 2 subjects C and D.

### 3.1.1 Paradigm

The P300 speller in the competition was like the following; the user is shown a 6x6 matrix of characters (Figure 3.1). Each row and column is intensified in a random sequence. Then, the user focuses his gaze on one of the 36 cells of the matrix. The sequence of 12 intensifications (6 rows + 6 columns) which results in an oddball paradigm where the intensification of the focused character represents the infrequent event and the other intensifications + no intensifications represents the frequent event. The event which is infrequent will result in a P300 response, while the frequent event will have no change on the signal (Figure 2.5). [6]

### 3.1.2 Recording

#### Competition's Data

In the competition's dataset (Subjects A and B), signals are filtered from 0.1 – 60Hz and digitized at 240Hz (Each 240 samples corresponds to 1 second). The 12 sets of intensifications were repeated 15 times for each character resulting in 180 total intensifications per epoch (character) into 64 channels EEG (Figure 3.2). There are 185 epoch in total; 85 train and 100 test per subject.

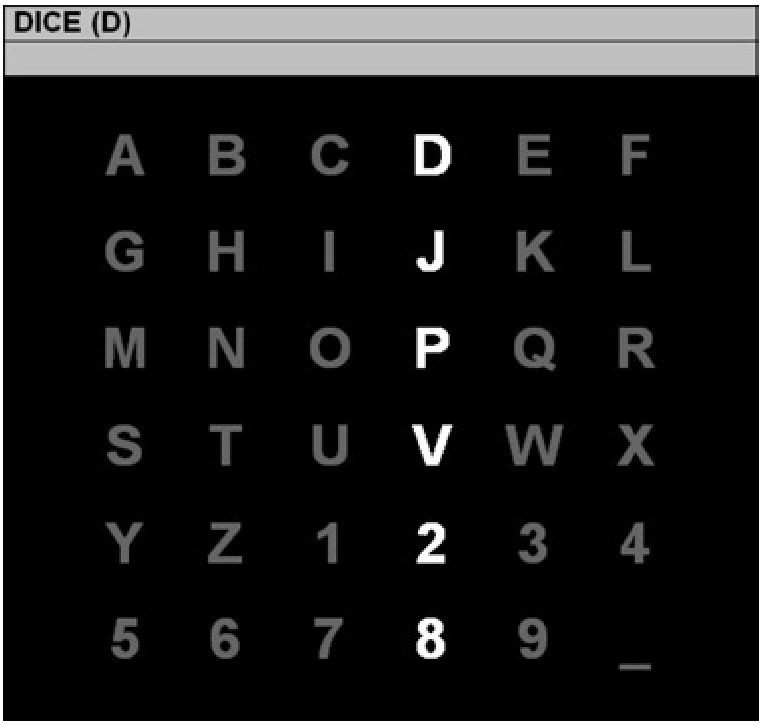



Figure 3.1: P300 Speller: Representation of the intensification of the 4th column [6].

**Recorded Data**

In the dataset we recorded (Subjects C and D), signals are digitized at 128Hz (Each 128 samples corresponds to 1 second). The 12 sets of intensifications were repeated 15 times for each character resulting in 180 total intensifications per epoch into 14 channels EEG (Figure 3.3, (a)). There are 72 epoch in total; 57 train and 15 test per subject.

. Note that, only 14 channels were recorded for subjects (C, D) due to hardware limitation (EMOTIV EPOC+). The 14 channels are AF3, F7, F3, P3, T7, P7, O1, O2, P8, T8, P4, F4, F8 and AF4 (Figure 3.3, (b))

### 3.1.3 Data Explanation

The competition's dataset has 5 arrays descried below (the description provided by the competition for dataset 2).

Signal: signal of each channel (Figure 2.5)
Type: 3D Array
Shape (Dimension): (85, 7794, 64)

TargetChar: The chosen character for each epoch
Type: String

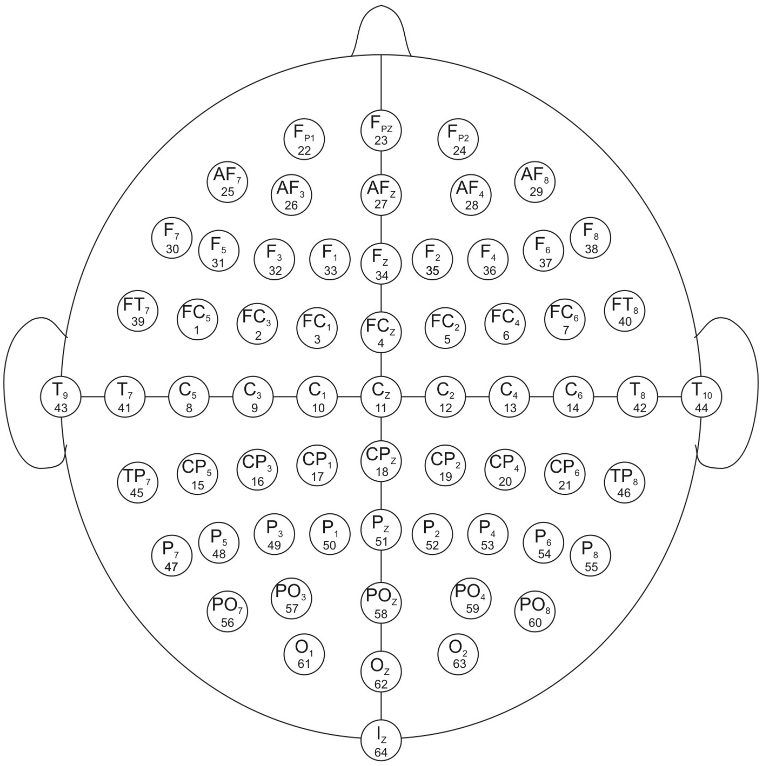



Figure 3.2: Mapping of sesnors on the scalp according to the competition's dataset description.

Length: 85

Flashing: 0 when NO row/column is intensified
1 when row/column is intensified
Type: 2D Array
Shape (Dimension): (85, 7794)

StimulusCode: 0 when NO row/column is intensified
1 ... 6 when intensified column (1 is left-most column)
7 ... 12 when intensified row (1 is upper-most row)
Type: 2D Array
Shape (Dimension): (85, 7794)
Figure 3.4

StimulusType: 0 when NO row/column is intensified
1 when row/column contains the chosen character
Type: 2D Array
Shape (Dimension): (85, 7794)

Note: This is dimension of the train dataset. If dimension of the test dataset is

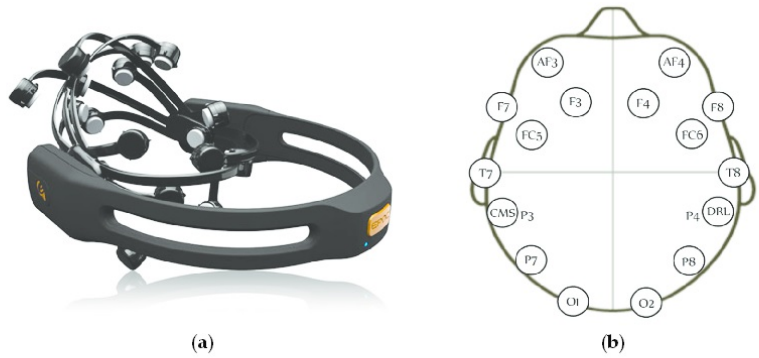



Figure 3.3: Mapping of sesnors on the scalp in EMOTIV EPOC+ headset [11].


Figure 3.4: Row/Column numbering values.

needed, substitute 85 with 100.

Note: TargetChar, StimulusType are only provided for training dataset.

Note: Since algorithm has been updated, only Signal, TargetChar, StimulusCode and the matrix used for training are needed to train the ML model.

## 3.2 Data Pre-processing

### 3.2.1 Average of Repetitions

Average of the all repetitions for both the competition's dataset and the recorded dataset were done according to the proposed algorithm provided by the competition in the example file. A modified version is that, we search for 2 consecutive samples in StimulusCode current and previous. If the current sample is zero, and the previous sample is non-zero, then that means there was a transition from intensification to no intensification. Then we sum the samples of the window needed (i.e. 192 samples = 800ms for competition's dataset) to the corresponding previous intensification samples.

For example, in the competition's dataset, according to Sub-Section 3.1.3, the train set has dimension (85, 7794, 64), after calculating the average the repetitions the new dimensions will be (85, 12, 192, 64).

Note: 7794 and 192 because the digitization rate for the the used headset in the competition is 240 Hz.

In the recorded dataset, the train set has dimension (57, 4600, 14), after calculating the average the repetitions the new dimensions will be (57, 12, 102, 14).

Note: 4600 and 102 because the digitization rate for the EMOTIV headset is 128 Hz.

### 3.2.2 Filters

#### Common Average Reference (CAR)

The signals were filtered using Common Average Reference [4, 5, 7] (CAR) filter (Equation 3.1). By subtracting the mean of samples across all channels at certain time from signals at the same certain time helps to remove the noise in some channels.

$$f_c(t) = s_c(t) - \frac{1}{N}\sum_{i=1}^{N} s_i(t) \tag{3.1}$$

$f_c(t)$ is the filtered signal for electrode (channel) c at time t. $s_c(t)$ is the raw signal for electrode (channel) c at time t. $\frac{1}{N}\sum_{i=1}^{N} s_i(t)$ is the mean of channels at time t.

#### Moving Average

After that, Moving average filter was applied in order to remove noise within time units per channel (Equation 3.2).

$$f_c(t) = \frac{1}{M}\sum_{i=t}^{t+M} s_c(i) \tag{3.2}$$

$f_c(t)$ is the filtered signal for electrode (channel) c at time t. $\frac{1}{M}\sum_{i=t}^{t+M} s_c(i)$ is the mean consecutive M signal where M is the moving average filter value.

In the competition's dataset, it will be set to 25 [4]. However, in the recorded dataset, it will be set to 13 [5].

### Z-Score

After applying moving average, the signals were scaled using Z-Score (Equation 3.3).

$$f_{cj} = \frac{s_c(j) - \mu_c}{\sigma_c} \tag{3.3}$$

$f_{cj}$ is the filtered signal for electrode (channel) c and feature j. $s_c(j)$ is the signals for electrode c and feature j. $\mu_c$ is the mean of signals recorded on electrode c. $\sigma_c$ is the standard deviation of signals recorded on electrode c.

### Decimation

Finally, the filtered signal is decimated by certain value. Assume decimation value of n. This means we will take 1 signal as a feature every n signals.

In the competition's dataset, it will be set to 12 [4]. However, in the recorded dataset, it will be set to 6 [5].

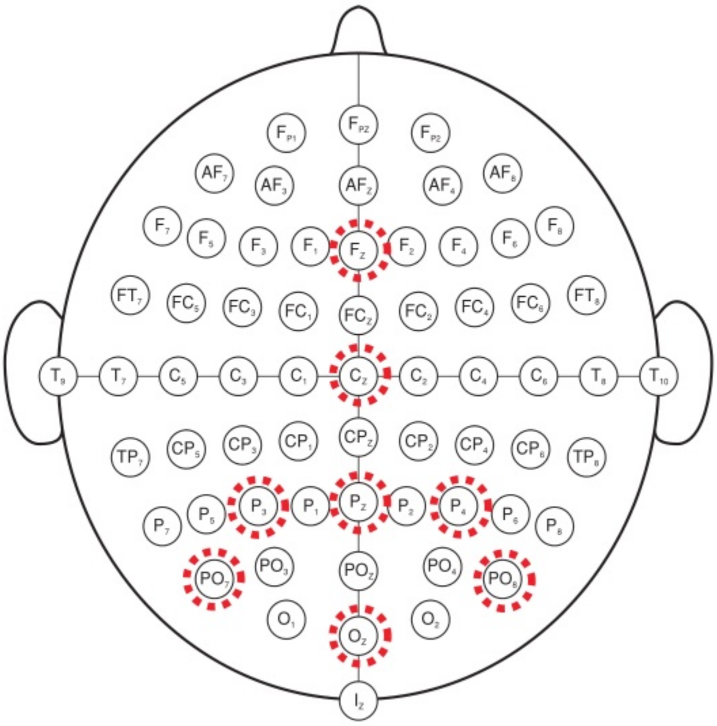



Figure 3.5: Most effective electrodes according to Dean and Sellers [6].

## 3.3 Feature Extraction

### 3.3.1 Channel Selection

Data segments of extracted data can vary a lot since the elicited signal appears from 250 - 500ms. Most papers [4, 5, 6, 7] uses data segments that vary from 600 - 1000ms. However, in this project, 0 - 800ms data segment is going to be used.

In the competition's dataset, the number of samples will be 192 sample. There are 2 sets of channels are going to be used; the first will consist of 8 channels (Fz, Cz, Pz, P3, P4, PO7, PO8 and Oz) (Figure 3.5) [4, 6]. And the other set will consist of all the 64 channels.

On the other hand, then number of sample for the recorded dataset will be 102 sample. And all the 14 channels are going to be used (Figure 3.3).

### 3.3.2 Feature Vector Concatenation

After channel selection, all the extracted channels will then be concatenated and passed to the classifier. This approach was used to allow the use of more than 1 channel to determine the P300.

## 3.4 Classification

Since P300 is an oddball paradigm, meaning that it is either frequent or infrequent event. The whole classification problem is a binary one. It is either P300 or Non-P300. In this project, linear classifiers whose decision boundary will be like Equation 3.4.

$$w^T x + b = 0 \tag{3.4}$$

$w$ is the weight (line slope). $x$ is the feature vector (sample). $b$ is the bias term (line shift).

### 3.4.1 Class Construction and Character Prediction

Since the problem is a binary classification problem, we will set the the target class to be 1 for the intensification with P300, and -1 (or 0) for the intensification with Non-P300.

Since the P300 Speller is only has 2 P300 events (1 for each dimension row/column), the character prediction will be based on the maximum value from the score function (decision function).

Note that, the higher the score (score is the predicted y label) (Equation 3.5), the further away from the corresponding line on the hyper plane, which makes it more accurate for a P300 elicit (Equation 3.6, and Equation 3.7) [4, 5, 6].

$$score = w^T x \tag{3.5}$$

$$row = max(w^T x_{7..12}) \tag{3.6}$$

$$column = max(w^T x_{1..6}) \tag{3.7}$$

$w$ is the weight. $x$ is the feature vector. $max(..)$ is a function to get max of given values. Check Sub-Section 3.1.3 for row/column numbering values.

### 3.4.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is the most basic linear classifier which is based on least squared error. The weight vector is calculated based on Equation 3.8 [4].

$$w = (X^T X)^{-1} X^T y \tag{3.8}$$

$w$ is the weight. $X$ is the matrix with feature vectors as rows. $y$ is the vector that has the labels of each row within $X$.

For Example, in the competition's dataset, $X$ has dimension (1020, 128), and $y$ has dimension (1020). Note that, we are using 85 character (train dataset), 12 intensifications, 192 sample (0 - 800ms) and 8 concatenated channel.

$$85 \times 12 = 1020$$

$$\frac{192}{12} \times 8 = 128$$

## 3.5 Graphical User Interface (GUI)

The language used to make the Graphical User Interface (GUI) is python programming language. As for the GUI package itself (classes and functions) is called Tkinter.

The user is shown the matrix in Figure 3.6a for 2.5 seconds. Then, the intensifications of rows and columns starts (12 intensifications, 15 repetitions) as in Figure 3.6b, after that, the GUI halts for 2 seconds because the P300 takes time to take effect (300ms after the stimulus).

Note that, number 5 is highlighted in Figure 3.6a to signify that the user should focus on it in order to train it.


(a) Non-intensification state of the GUI (frequent event).

(b) Intensification state of column 4 state of the GUI (infrequent event).

Figure 3.6: GUI used to record the dataset.

# Chapter 4

# Results

## 4.1 Competition's Dataset

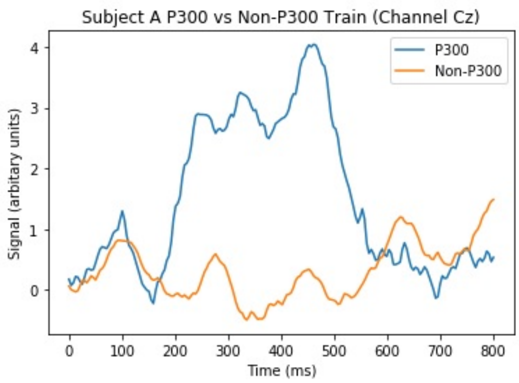
The user was shown a 6x6 matrix containing A..Z, 1..9 and _ as shown in Chapter 3, Figure 3.1. Each row and column was intensified once in a random sequence resulting in 12 intensifications in total. Each trial (12 intensification) was repeated 15 times which results in 180 total intensification per epoch. Each intensification lasts for 100ms followed by 75ms without any intensifications and so on. The above information are according to the description provided by competition 3 dataset 2.

Each subject (A and B) has 85 epoch (character) as train dataset and 100 epoch as test dataset. Figure 4.1 and Figure 4.2 shows the average of P300 plot for the 2 intensifications (row and column) that has the chosen character versus the 10 intensifications that does not contain the character.

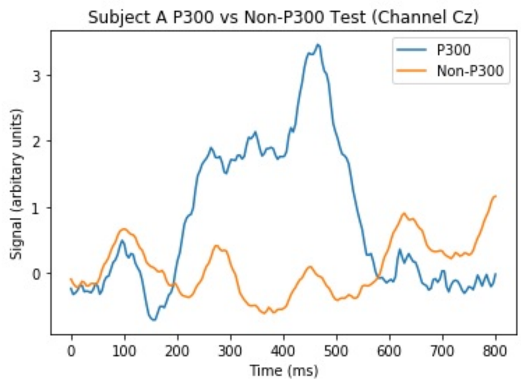
Table 4.1 shows the character recognition analysis for subjects A and B according to the proposed channels mentioned in Sub-Section 3.3.1.

| Classifier | Channels | Window | Filters | A | B | Average |
|---|---|---|---|---|---|---|
| **LDA** | 8 | $0 \rightarrow 800ms$ | - | 73 | 81 | $77 \pm 4\%$ |
| | 8 | $0 \rightarrow 800ms$ | CarMaZsD | 79 | 90 | $84.5 \pm 5.5\%$ |
| | 8 | $200 \rightarrow 600ms$ | - | 71 | 85 | $78 \pm 7\%$ |
| | 8 | $200 \rightarrow 600ms$ | CarMaZsD | 65 | 78 | $71.5 \pm 6.5\%$ |
| | 64 | $0 \rightarrow 800ms$ | - | 81 | 85 | $83 \pm 2\%$ |
| | 64 | $0 \rightarrow 800ms$ | CarMaZsD | 89 | 84 | $86.5 \pm 2.5\%$ |
| | 64 | $200 \rightarrow 600ms$ | - | 86 | 82 | $84 \pm 2\%$ |
| | 64 | $200 \rightarrow 600ms$ | CarMaZsD | 82 | 78 | $80 \pm 2\%$ |

Table 4.1: Subject A and B character recognition rate.

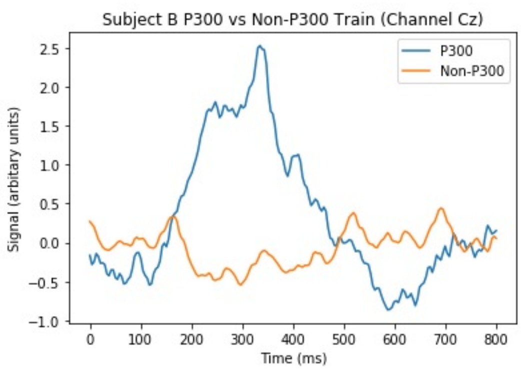



(a) Subject A P300 plot on train dataset on channel Cz.

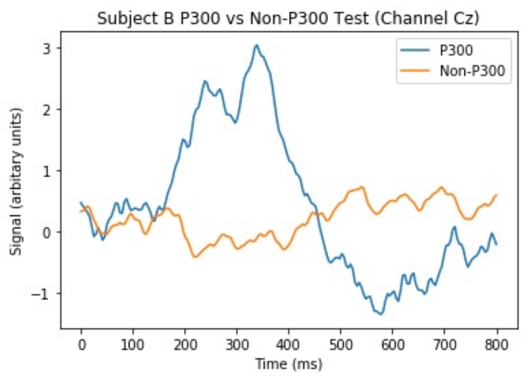



(b) Subject A P300 plot on test dataset on channel Cz.

Figure 4.1: P300 plot of subject A.




(a) Subject B P300 plot on train dataset on channel Cz.




(b) Subject B P300 plot on test dataset on channel Cz.

Figure 4.2: P300 plot of subject B.

## 4.2 Recorded Dataset

The same paradigm was applied as in Section 4.1 same matrix, 12 intensifications, 15 repetitions, 100 ms for intensification and 75ms no intensification.

However, Each Subject (C and D) has 57 epoch as train dataset and 15 epoch as test dataset. Figure 4.3 and Figure 4.4 shows the average of P300 plot all 72 epochs.

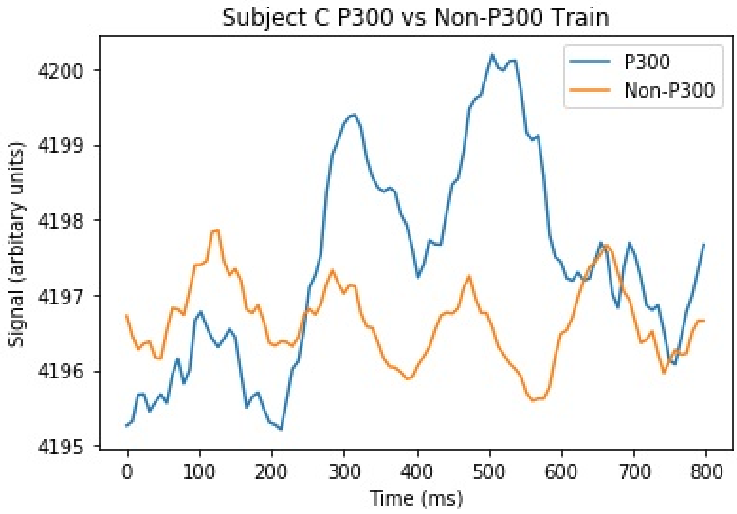



Figure 4.3: Subject C P300 plot.

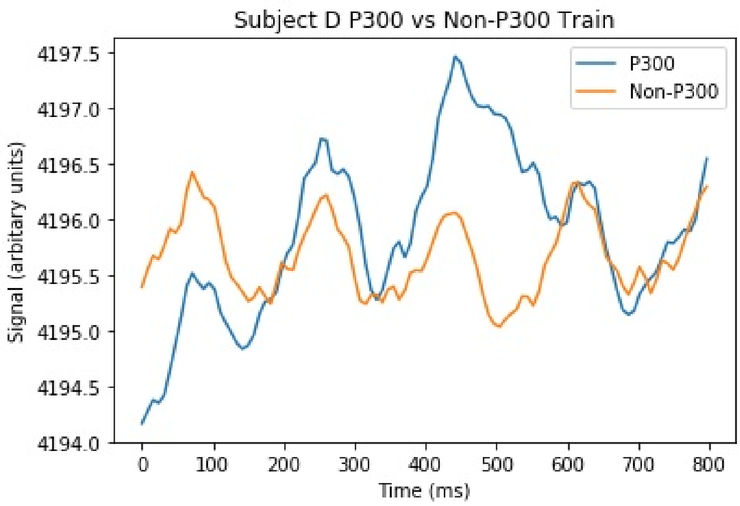



Figure 4.4: Subject D P300 plot.

Table 4.2 shows the character recognition analysis for subjects C and D using the 14 channels of EMOTIV EPOCH+ Headset (Figure 3.3).

| **Classifier** | **Channels** | **Window** | **Filters** | **A** | **B** | **Average** |
|---|---|---|---|---|---|---|
| **LDA** | 14 | $0 \rightarrow 800ms$ | - | 53.36 | 53.36 | 53.36% |
| | 14 | $0 \rightarrow 800ms$ | CarMaZsD | 86.67 | 60 | $73.33 \pm 13.33\%$ |
| | 14 | $200 \rightarrow 600ms$ | - | 46.67 | 46.67 | 46.67% |
| | 14 | $200 \rightarrow 600ms$ | CarMaZsD | 46.67 | 46.67 | 46.67% |

Table 4.2: Subject C and D character recognition rate.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

In conclusion, using P300 based BCI a P300 speller was programmed. Using ML and specifically LDA classifier, a character recognition rate of around 80% in the competition's dataset and 70% in the recorded dataset was achieved. The number of channels didn't have much effect, at least it depends on the subject, their brain waves and their state of mind 4.1.

However, one could see that applying filters did have great impact on the character recognition rate. As shown in Table 4.2, an increase of around 20% was achieved on subject C, however, on subject D there was no effect at all, mainly due to signal interference, lack of attention and lack of experience with the platform. One can see that effect from the P300 plot in Figure 4.3 versus Figure 4.4.

## 5.2 Future Work

There is a lot can be done to increase the efficiency of this project. Such as trying different combinations of filters and classifiers for each subject since ML models are like maths problems where one can achieve the desired answer with more than one way.

An auto-complete functionality can be added as well in order to save time. one character takes around 30 seconds to be determined. If we consider the average of one word to be 5 characters, then, this will take around 150 seconds in total which is a lot for one word. However, if auto-complete was added it can take half the time to determine what the user wants thus saving a lot of time.

Sentence's prediction and adaptability can be added as well. As it can check what the user tends to type so often then add it as a shortcut instead of typing the whole sentence from the beginning and predicting next word within the current sentence (same behavior as mobile keyboards).

A mobile phone application can be programmed as well since P300 speller does not require much computation. However, it will be hard to do it on mobile phones with small screens or older version platforms since it might not be compatible with the headset.

# Appendix

# Appendix A

# Lists

| | |
|---|---|
| **BCI** | Brain–Computer Interface [2] |
| **BMI** | Brain-Machine Interface |
| **CAR** | Common Average Reference [4, 5, 7] |
| **DBI** | Direct Brain Interface |
| **ECoG** | Electrocorticogram [2] |
| **EEG** | Electroencephalography [2] |
| **ERP** | Event Related Potential |
| **GUI** | Graphical User Interface |
| **ICR** | Intracortical Recordings [2] |
| **LDA** | Linear Discriminant Analysis |
| **ML** | Machine Learning [3] |
| **P300** | Peak 300 [1] |
| **SCI** | Spinal Cord Injury |
| **SSVEP** | Steady State Visually Evoked Potential [10] |

# List of Figures

# List of Tables

# References


[1] L. Farwell, “Talking off the top of your head: A mental prosthesis utilizing event-related brain potentials,” *Electroencephalography Clinical. Neurophysiolgy*, vol. 70, pp. 510–523, 01 1988.

[2] B. Graimann, B. Allison, and G. Pfurtscheller, *Brain–Computer Interfaces: A Gentle Introduction*, pp. 1–27. 01 1970.

[3] T. Mildenberger, “Stephen marsland: Machine learning. an algorithmic perspective,” *Statistical Papers*, vol. 55, 05 2014.

[4] A. Elsawy, S. Eldawlatly, M. Taher, and G. M. Aly, “A principal component analysis ensemble classifier for p300 speller applications,” pp. 444–449, 09 2013.

[5] A. Elsawy, S. Eldawlatly, M. Taher, and G. M Aly, “Performance analysis of a principal component analysis ensemble classifier for emotiv headset p300 spellers,” vol. 2014, pp. 5032–5, 08 2014.

[6] D. Krusienski, E. Sellers, F. Cabestaing, S. Bayoudh, D. Mcfarland, T. Vaughan, and J. Wolpaw, “A comparison of classification techniques for the p300 speller,” *Journal of neural engineering*, vol. 3, pp. 299–305, 01 2007.

[7] A. Elsawy, S. Eldawlatly, M. Taher, and G. M. Aly, “Mindedit: A p300-based text editor for mobile devices,” *Computers in Biology and Medicine*, vol. 80, 11 2016.

[8] S. Sutton, P. Tueting, J. Zubin, and E. John, “Information delivery and the sensory evoked potential,” *Science (New York, N.Y.)*, vol. 155, pp. 1436–9, 04 1967.

[9] R. Fazel-Rezai, B. Allison, C. Guger, E. Sellers, S. Kleih, and A. Kübler, “P300 brain computer interface: Current challenges and emerging trends,” *Frontiers in neuroengineering*, vol. 5, p. 14, 07 2012.

[10] C. Herrmann, “Human eeg responses to 1–100 hz flicker: Resonance phenomena in visual cortex and their potential correlation to cognitive phenomena,” *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, vol. 137, pp. 346–53, 05 2001.

[11] P. Batres-Mendoza, C. R Montoro-Sanjose, E. I Guerra-Hernandez, D. Almanza-Ojeda, H. Rostro, R. Romero-Troncoso, and M. Ibarra-Manzano, "Quaternion-based signal analysis for motor imagery classification from electroencephalographic signals," *Sensors*, vol. 16, pp. 1–17, 03 2016.