ERROR DETECTION AND NEW STIMULUS MECHANISMS IN BRAIN-COMPUTER INTERFACE

By
Hamza ALTAKROURY

Submitted to the Graduate School of Engineering and Natural Sciences in
partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

June 2013

ERROR DETECTION AND NEW STIMULUS MECHANISMS IN BRAIN-COMPUTER INTERFACES

APPROVED BY

Assoc. Prof. Müjdat Çetin
(Thesis Supervisor)

Prof. Dr. Aytül Erçil

Asst. Prof. Hakan Erdoğan

Assoc. Prof. Volkan Patoğlu

Assoc. Prof. Berrin Yanıkoğlu

DATE OF APPROVAL 01/07/2013

ACKNOWLEDGEMENTS

I would like to thank my advisor Müjdat Çetin Hoca for being very helpful in both the academic life and the social life in Sabancı University. It was a great opportunity to be with him in this period. It is hard to describe the characteristics of Müjdat Çetin Hoca here, a person should see him to know how much Müjdat Hoca is humble and moral person.

I want also to thank Mrs. Elif Tanrıkut who is working in the Student Resources department in Sabancı University , she was the first Turkish citizen with whom I started my communication to come to Turkey, she was really very helpful in giving information about the applications for Sabancı University.

Also, I should not forget Güniz Evirgen Hoca for accepting me in her Turkish classes, by having these classes I could understand people with great morals and great talents. These classes made my life in Turkey much beautiful.

I would not forget to remember Sabancı University for supporting me during the Master period my professors in Sabancı University who stood in front of us for hours giving us from their knowledge and experience also all my partners in VPA Lab, who were very helpful and open to any kind of questions and discussions, among them I want to give a special thanks to Ela Koyaş who accepted to translate the Abstract into Turkish.

In addition, I want to thank my brothers and sisters in Sabancı University who really makes my staying in the university, in particular, and in Turkey, in general, meaningful and valuable. They were and are always considered as my family.

For my parents, thanking is really meaningless in front of their great and indescribable efforts and support which did not stop until this stage of life. I ask ALLAH to protect them and give them long and beautiful life.

It is really hard to leave this community, this environment and those people after these three years. I hope that, as we met in life once, we meet in the other life, and for that I ask ALLAH to protect those people and show them the way that he likes.

Finally, I should mention that, this work was partially supported by Sabancı University under Grant IACF-11-00889, and by the Scientific and Technological Research Council of Turkey under Grant 11E056.

Contents

List of Tables

List of Figures

Abstract

ERROR DETECTION AND NEW STIMULUS MECHANISMS IN BRAIN-COMPUTER INTERFACE

Hamza ALTAKROURY

Electronics Engineering and Computer Science, MS Thesis, 2013

Thesis Supervisor: Associate Prof. Müjdat Çetin

Keywords: Brain Computer Interface, P300 paradigms, Error related Potentials.

Brain Computer Interfaces (BCIs) constitute a research field whose motivation is to help disabled individuals to communicate with the environment around them directly through the electrical activity of their brain rather than by the usual muscular output mechanism of the human body. The idea of non-invasive BCI is based on collecting brain signals using medical electrodes placed on the scalp of the patient and then trying to understand what the patient is trying to do/say by automatically analysing the collected signals. In other words, BCI can be imagined as a way to compensate the damaged internal nerves that used to carry signals from the brain, by using external cables connected with the computer.

Although extensive research continues to be carried out in the field of BCI, still BCI is working only inside laboratories. This is due to the weakness of the brain signals that are acquired. It is impossible to understand always the meaning of the signals without error. The existence of errors in such systems means that it is impossible to depend totally on them to control the life of disabled individuals.

One of the well-known BCI types is called the P300 paradigm. It provides individuals with a method to choose any target only by concentrating on this

target while it is flashing. The flash on the screen is considered as a stimulus for the brain, and the brain's response to this stimulus is known as the P300 signal and can be detected in the acquired signals from the brain. P300-BCI is one of the most well-known paradigms in the BCI field.

One way to reduce the number of errors in any BCI system in general, and in P300 paradigms in particular, may be by using Error-related Potentials (ErrP). These ErrP signals are generated when the subject detects an error in the system. Therefore, these signals could be used as a feedback for the BCI system to verify its last response. If the BCI system, for example, generates a wrong output, then an ErrP will be generated from the subject's brain which could be exploited to generate a message that the last output generated is not correct. Another way to reduce the number of errors, in the context of P300 paradigms, may be by making the neighbour non-target items have the same job of the target item. By using this idea, whether the subject gives attention to these non-target items or not, the output will be as the subject expects.

In this research, we have experimentally examined two different scenarios for generating ErrP signals. Having ErrP signals from two different scenarios makes it possible for us to see if the ErrP signals have the same characteristics under different scenarios. In addition, we have implemented a new P300 paradigm motivated by a BCI-based robotic control application, in which the target's neighbour items have the same job of the target itself. In this new implementation, we get better classification performance through an analysis that compensates for the change in the number of classes.

Özet

BEYİN-BİLGİSAYAR ARAYÜZÜNDE HATA TESPİTİ VE YENİ UYARAN MEKANİZMALARI

Hamza ALTAKROURY

Elektronik Mühendisliği, Yüksek Lisans Tezi, 2013

Tez Danışmanı: Doç. Dr. Müjdat Çetin

Anahtar Kelimeler: Beyin Bilgisayar Arayüzü, P300 Paradigmaları, Hata İle İlgili Potansiyeller

Beyin Bilgisayar Arayüzleri (BBAlar), engelli bireylerin, çevreleri ile insan vücudunun normal kas mekanizması tarafından değil de doğrudan beyinlerindeki elektrik faaliyeti ile iletişim kurmalarına yardım etme motivasyonuna sahip bir araştırma alanıdır. İstilacı olmayan BBA'lar beyin sinyallerini hastanın kafa derisine yerleştirilen tıbbi elektrotlarla ölçmek ve sonrasında hastanın ne istemeye/söylemeye çalıştığını toplanan sinyalleri analiz ederek otomatik olarak anlamaya çalışmak üzerine kuruludur. Diğer bir deyişle, BBA beyinden gelen sinyalleri taşımak için kullanılan hasarlı iç sinirlerin yerini bilgisayara bağlı harici kablolar ile doldurmak için bir yol olarak düşünülebilir.

BBA alanında kapsamlı araştırmalar yapılmasına rağmen, BBA hala sadece laboratuvarlar içinde çalışıyor. Bunun sebebi elde edilen beyin sinyallerinin zayıflığıdır. Sinyallerin anlamını her zaman hatasız olarak anlamak mümkün değildir. Bu gibi hataların sistemlerdeki mevcudiyeti, engelli bireylerin yaşamlarını tamemen bunlara bağlı kılmamızın mümkün olmadığını gösterir.

İyi bilinen BBA türlerinden biri, P300 paradigmasıdır. Bu paradigma, bireye kendi belirlediği herhangi bir hedefi yanıp sönerken o hedefe odaklanıp seçmesi için bir yöntem sağlamaktadır. Ekrandaki hedefin yanması

beynin bir uyaranı olarak değerlendirilir ve beynin bu uyarana yanıtı P300 sinyali olarak bilinir ve beyinden elde edilen sinyallerde tespit edilebilir. P300 tabanlı BBA, BBA alanında en tanınmış paradigmalardan biridir.

Genel olarak herhangi bir BBA sistemindeki ve özellikle P300 paradigmalarındaki hata sayısını azaltmak için hata ile ilgili potansiyelleri (ErrP) kullanmak bir yol olabilir. ErrP sinyalleri birey sistemde bir hata tespit ederse oluşur. Bu nedenle, bu sinyaller BBA sisteminin son yanıtını doğrulamak için bir geri besleme olarak kullanılabilir. BBA sistemi, örneğin, yanlış bir çıktı üretirse, bireyin beyninde oluşan ErrP kullanılarak, yanlış bir çıktı elde edildiğine dair bir mesaj üretilebilir. P300 paradigmaları bağlamında, hata sayısını azaltmak için bir başka yol da hedef olmayan komşu öğeleri, hedef öğe ile aynı işleve sahip yapmak olabilir. Bu fikri kullanarak, birey hedef olmayan bu öğelere dikkat versin veya vermesin, çıktı bireyin beklediği gibi olacaktır.

Bu araştırmada, ErrP sinyalleri üretmek için deneysel olarak iki farklı senaryoyu inceledik. İki farklı senaryodan elde edilen ErrP sinyallerine sahip olmak, ErrP sinyallerinin farklı senaryolar altında aynı özelliklere sahip olup olmadığını görebilmemizi mümkün kılar. Buna ek olarak, komşu öğeleri hedef öğe ile aynı işleve sahip olan ve motivasyonu BBA tabanlı robotik kontrol uygulaması olan yeni bir P300 paradigması geliştirdik. Bu yeni uygulamada, sınıf sayısındaki değişikliği telafi eden bir analiz ile daha iyi bir sınıflandırma başarımı elde ettik.

Chapter 1

Introduction

This chapter introduces the idea of Brain-Computer Interface, and talks about some of the problems that it faces. Also it mentions some of the previous works that have been done to solve these problems. Then, it describes the contribution that this thesis provides.

1.1 Motivation

Brain-Computer Interface (BCI) is a new and fast-growing field that aims to help disabled individuals. Around the world there are many research groups working on this field in its various forms. Although many articles have been published under this title, still brain computer interface is working only inside laboratories. The major barriers that stand in front of bringing the BCI to work in the real-world are: First, the low signal-to-noise ratio of the acquired signals. This makes it impossible to have a robust system that can work without errors. Second is the non-stationarity nature of the brain signals. This non-stationarity makes it essential to modify the parameters of the system before each run. This calibration takes time and the subject should be part of it.

In this research we try to find a way to reduce the number of errors that occur in the BCI-systems by using the so called Error Related Potentials in one way, and by modifying the interface of the P300 paradigm in the other way. In addition we suggest a way to use these potentials in making the BCI-systems adaptive.

1.2 Existing Research

One of the well-known BCI applications is the P300 speller, this speller was found to be one of the most robust applications in the field of BCI [1]. The first main purpose of this speller was to enable disabled individuals to type letters just by focusing on a specific letter, Chapter 2 gives the details about this speller. However the P300 is still slow and training before each use is required.

To eliminate the operation of training before each use, there exist ongoing research efforts, under the name of adaptive BCI [2] [3] [4]. Adaptive system means to have a system that is able to work well without training even if the time between the first training and the present run is long. Our research started with the aim of making the P300 speller adaptive using a cognitive signal called Error Related Potential (ErrP). Using ErrP in the P300 speller for adaptivity purpose is a new idea.

Several researches have worked on acquiring and detecting ErrP signals [5] [6] [7]. But results show that there are differences in these signals depending on the scenario in which they are acquired. There is no previous research that performed a comparison of ErrP signals acquired from two different scenarios. Here, in this research, the results that were found out from ErrP under P300 speller pushed us to see (before making P300 adaptive as was the aim) if the pattern of the ErrP acquired in P300 experiment can be found in ErrP signals that are acquired using another scenario.

On the other hand, the implementation of the first P300 speller [8], urges the workers in this field to use this idea not just in keyboards and speller. Many works have been done using the P300 signals in different paradigms to move wheelchairs, robots or any kind of mechanical devices [9] [10]. In this research, a new paradigm, which, to the best of our knowledge is original, has been implemented so that the P300 interface could be used to control a robot that moves in four different directions. The new paradigm, as it will be shown, reduces the errors of the P300-paradigm.

1.3 Contribution

This thesis makes two contribution in the field of BCI that can be used in the future to implement a robust and adaptive BCI-system. The first contribution is the implementation of two paradigms that are able to generate ErrP signals from subjects. The first paradigm has a similar interface with the P300 speller. The ErrP acquired from this interface can be used to implement an adaptive P300-speller. The first paradigm that was used in this research to generate ErrP signals is similar to that used in [7]. However to see if ErrP signals are similar to those that can be generated from another interface, i.e., to see whether the properties of the ErrP are independent of the scenario, another paradigm was implemented to generate ErrPs.

The second paradigm that was implemented in this work is similar to that found in Chavarriaga et al., [6], but the main difference here is having a more stable paradigm. The cursor in [6] is moving continuously which may cause an Electrooculography (EOG) contamination. These EOG signals make it hard to classify the ErrP signals. And also in [6] the steps that the box passes is only three which may not attract the concentration of the subject. In this work the cursor on which the subject should concentrate is constant and the box should walk 10 steps to reach the target.

The second contribution is the implementation of a modified P300-paradigm for controlling a robot. This paradigm was implemented based on new idea of having multiple choices for the same target. By having this paradigm we could decrease the number of errors and at the same time having higher bit rate compared to the usual P300 speller. To the best of our knowledge, such a modification has not been considered before. In most previous work involving the use of P300 for robot control, including, e.g., [11] and [9], a single choice in the stimulus matrix corresponds to a single target for the robot. Such systems are prone to too many errors.

1.4 Thesis Outline

The work in this thesis is organized as follows: In the Background Chapter (Chapter 2), a general picture is shown about the field of BCI, some of the EEG signals are characterized, and some problems that BCI faces are de-

scribed. It also talks briefly about one of the most used classifier in the BCI field.

In the Recognition of EEG signals Chapter (Chapter 3), the idea of detecting EEG signals is discussed, it also mentions some of the methods that were used in the literature for detecting EEG in general, and ErrPs in particular. Then it ends describing the methods that were used in this work to detect the ErrPs.

The Detection of Error Related Potentials Chapter (Chapter 4), describes first the motivation for detecting ErrP potentials in BCI systems, then it reviews some of the previous work that has been performed for detecting ErrP potentials. And finally, it illustrates the work of this thesis for detection and classification of ErrP.

P300 and Mechanical Devices Chapter (Chapter 5), shows some of the previous work that was carried to use the P300 paradigm for controlling mechanical devices. Then it presents the work that has been carried out in this thesis to modify the P300 paradigm for using it later to control four-direction robot.

Finally, the thesis ends summarizing the work that has been performed, commenting on its results, and proposing some ideas for those who want to continue in this field.

Chapter 2

Background

This chapter explains the idea of Brain-Computer Interface. It also focuses on P300 speller and how it works. Then it moves on to describe Error Related Potentials and how they could be used in BCI and how adaptivity could be achieved using ErrPs.

2.1 Introduction

BCI is a new research field [1]. It aims to help disabled individuals. Research on BCI is based mainly on the techniques of signal processing and machine learning. BCI could be defined as an electrical medium that connects the disabled individual with the environment through the computer, and compensates his damaged neural and/or muscular communication channels.

In general, disability could be due to diseases such as Cerebral Vascular Accident (CVA), Spinal Cord Injury (SCI), Traumatic Brain Injury (TBI), Multiple Sclerosis (MS), Amyotrophic Lateral Sclerosis (ALS) and Parkinson's disease.

The idea of BCI is based on acquiring brain signals, which are related to specific acts, then using these signals after processing as examples for the computer (which has a classifier). The role of the computer then is to understand these acts whenever it notices similar signals.

The history of recording brain signals had begun in 1929 when the German scientist Hans Berger recorded the electrical brain activity from the

human scalp. At that time, the required technologies for measuring and processing brain signals were still too limited [1].

The first BCI was described by Dr. Grey Walter in 1964 [1], where he had succeeded in detecting the action of pressing a button from the brain signals before the button being actually pressed. (The detection of the brain signals was faster than the action of moving the hand). Before that all BCI was just a matter of science fiction.

2.2 Brain Signals

There are three ways to acquire brain signals. In the first one the signals are acquired from the scalp and no surgery is needed, the signals acquired by this method are called Electroencephalography (EEG). In the second way the signals are called Electrocorticogram (ECoG) and they are acquired from the cortex. This approach necessitates a surgery for opening the skull. The third way needs the electrodes to penetrate the tissue of the brain. Signals acquired by this method are called Intracortical Signals.

Acquiring EEG is the safest and most common in the field of BCI research, however it is the most challenging due to the weakness of power of these acquired signals. On the other hand, Intracortical Signals are stronger and have better signal-to-noise ratio, but the need for a surgery in the case of Intracortical Signals is dangerous and it is hard to find volunteers for the research. Regarding safety and signal-to-noise ratio, ECoG signals are in the middle.

2.3 Electrodes

The sensors that are used to acquire the medical signals are called electrodes. These electrodes are made of conductors, and they acquire signals from non-metallic mediums placed on the skin of the patient. The common electrodes used for acquiring EEG signals are the Ag/AgCl electrodes. These electrodes could have many shapes.

One type of these is called the reusable disk, where it is made from silver or gold. These disks needs sticks to be fixed on the specified place of the patient's skin. Another type of electrodes are designed in a way to match a special cap that the subject wears. This special cap is designed to facilitate the procedure of putting the electrodes in its place accurately.

The conductivity between the electrodes and the the scalp of the patient is reduced by the patient's hair. Because of that in most types of electrodes, a conducting gel is used to ensure the conductivity between the skin and the electrodes. However, when using the new-produced dry electrodes there is no need to apply any conducting gel.

2.4 Types of BCI

BCI-systems can be classified in many ways. They could be classified according to the paradigm under which the signals are acquired; and here there are many classes, like the P300-BCI, the Steady State Visually Evoked Potential-BCI (SSVEP-BCI), the motor imagery BCI and so on. In another classification, BCI-systems can be classified into two classes. The first is the Synchronous BCI and the second is Asynchronous BCI.

In the first class, i.e., synchronous BCI, there are markers (cues shown to the subject) for marking the beginning of the signal. For example in the motor imagery BCI where the subject tries to imagine the movement of his/her extremities, the marker usually marks the beginning of the imagination of motion. This marker could be used also to mark the beginning of the event in the case of Event Related Potential (ERP) paradigms, the event usually is in the form of sound or visual stimulus. In ERP-BCI, the brain's responses to these events are studied.

One of the most robust BCI-based systems is the P300, which is classified under the ERP-BCI. This paradigm is based on having a low probability target stimulus among high probability non-target stimuli to generate a positive-going signal approximately after 300 ms from a visual or auditory stimulus [12]. This kind of BCI is explained in detail in the following section.

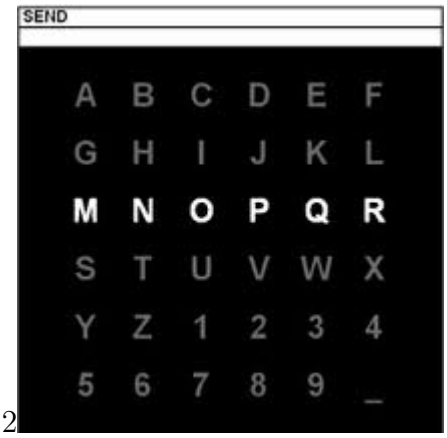

Figure 2.1: Farwell and Donchin Matrix.

Another simple and robust BCI system is known as the Steady-State Visual Evoked Potentials (SSVEP). Under this type of BCI the subject is asked to focus on a target stimulus with specific frequency among non-target stimuli with different frequencies. It has been found that the acquired signals' frequency is equal to the stimulus frequency. In Asynchronous BCI there is no marker to follow. The process of splitting the signals depends on windowing technique. This case is used mostly in the motor imagery BCI.

2.5 P300 Stimulus

One of the most common BCI systems is based on the P300 signals. These signals are named so because they appear as a positive-going component after 300 ms of a low probability stimulus [12]. The idea of having a distinguished signal appearing after a low probability stimulus had urged Farwell and Donchin to implement a keyboard based on the P300 signals [8]. Fig. 2.1 shows this P300-based keyboard which is known as the P300 speller.

In this matrix, each row and column flashes for a specific period of time in a random manner. The target letter is located in the junction of the target row and the target column. The target row and the target column have low probability; because of this each of them elicit a P300 component. By knowing the times of the P300 components the system can refer to the times of the flashes and know where the target letter is.

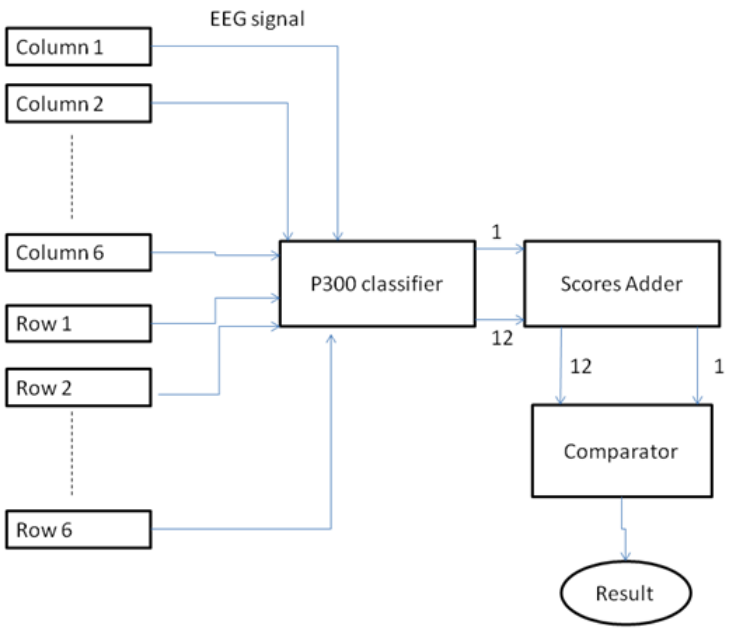

Figure 2.2: Classifying of a single signal component: Here the signals of each trial is first classified as P300 signal (labeled as 1) or non-P300 (labeled as 0), the other trials then support or oppose the previous ones through the score.

The flashing of the entire matrix is called a trial. Unfortunately, because the EEG has low signal-to-noise ratio the target signals (P300) can't be certainly distinguished from the non-target signals only by depending on one trial; the system should detect the output of more than one trial so that the classifier can be more robust.

Combining the results of the flashes could be done in two ways. In the first, the signals are directly entered to the classifier which classifies each signal as P300/non-P300. The row and the column that have the highest probability of being P300 gets 1-score and the others 0-score, after completing the trials, the comparator decides which is the target letter according to the scores. Fig. 2.2 summarize the idea in a simple graph.

In the other way, the signals that are generated from all the trials are averaged, so that the signal-to-noise ratio becomes larger. After that, the

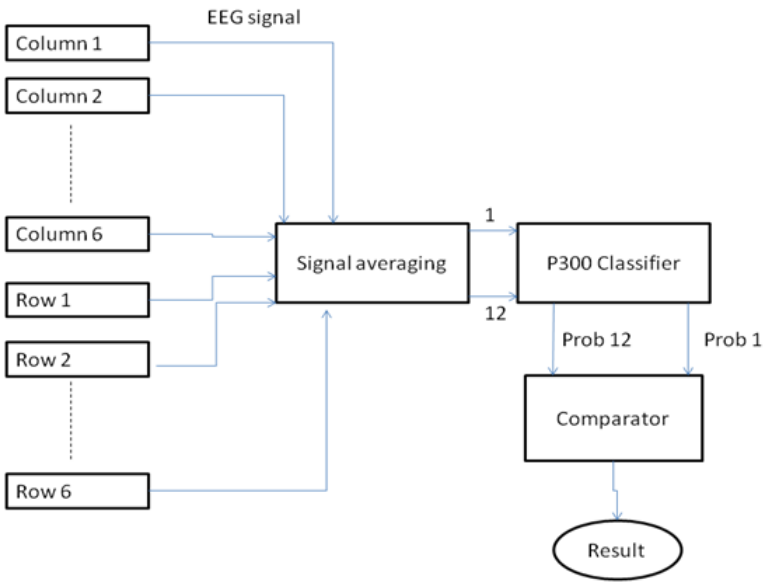

Figure 2.3: Classifying the average of the signal components: Here the signals first are being averaged, then entered to the classifier, the output of the classifier is normally a posterior probability indicating the likelihood that the signal is a P300 or not.

average is being used as an input to the classifier which finds the likelihood for a signal to be P300. The signal that has the highest probability will be classified as P300, then the target letter is found. Fig. 2.3 shows this technique. In either case, having to use more than one trial in the P300-speller is one of its major disadvantages. This repetition of the trials requires time; and that makes the system slow.

P300 paradigms could be used also to control various mechanical devices. This can be done by using directions, words, picture or any suitable elements in the matrix instead of letters. When P300 scenario is used to control mechanical devices, the cost of error should be considered. It is easy to notice that errors in moving mechanical system are much more costly that of those errors occurs in typing letters.

2.6 Error Related Potentials

One kind of EEG signals is called Error Related Potentials (ErrP), these signals are generated when the subject faces an error, and this error could be due to the subject himself/herself or from the machine. In previous articles it is stated that the shapes of the ErrP acquired from different experiments are different, i.e. in the P300 experiment the shape of ErrP signal will be different than that in the Motor Imagery experiment [5]. Also it is found that the shape of the error signals that are generated when the subject makes the error is different from the shape of the signals that are generated when the error is due to the system, i.e., when the system misinterprets the subject's command. [13].

Detection of ErrP could be used to correct the errors, this can be done in a binary classification problem. For example in the Motor Imagery a classification of left hand versus right hand could be done. If the system decides the output to be the left hand and detects ErrP, it can directly change its decision to right hand. But this cannot be done in a multi-class classification problem. Here the detection of ErrP can only be used to ignore the erroneous result (the one that generates the ErrP).

The detection of ErrP under the P300 paradigm could be done in two ways. First it could be used as the backspace button. In this case, if the machine displays a letter then the classifier detects ErrP, the machine will directly delete that choice. Doing so makes the P300 speller faster because deleting a letter normally requires the subject to concentrate on the backspace choice and wait one more trial (flashing of all the rows and columns around 5 to 10 times).

The second way is similar to the first in deleting the choice which is followed by ErrP, but also it has the advantage of making the speller faster by choosing the letter that has the second rank, which could be with high probability the right letter. In this case the detection of ErrP will force the system to choose the second high score letter instead of the highest score letter.

But if the classification is based on the average of the components, the column that has the highest posterior probability and the row that has the

highest posterior probability will be chosen. However if an ErrP is detected the second highest probability row and the second highest probability column will be compared, and the one that has higher posterior probability of them will considered.

2.7 Adaptivity

The EEG signals are random and contain stationary components and non-stationary components. Due to these non-stationary components the BCI system acts differently on different subjects and even on the same subject on different sessions [6] [12].

Often this problem is being solved by retraining the classifier of the signals before each session. Retraining makes the system capture the general shape of the signals before putting it under work. Repeating the operation of training is a time-consuming and inconvenient process. Many ideas were proposed to make the system adaptive; i.e., to make the system able to change its parameters during the test session to keep the classification performance high without repeating the training session.

Chapter 3

Recognition of EEG signals

This chapter talks about the idea of signals recognition, also it describes a number of methods that are used in recognizing EEG signals, in general, and ErrP in particular. This chapter contains also a description of the methods that were used in this work to recognize the signals.

3.1 Introduction

As mentioned previously, EEG signals have low signal-to-noise ratio, this makes the job of detecting such signals hard and requires the use of "clever" techniques to recognize these signals. The techniques that are used in detecting signals, in generals, and EEG signals, in particular, could be decomposed in two parts, the first part is known as the Feature Extraction and the second is the classification part.

In Feature Extraction only the components that are needed to identify the signal are considered. For example, instead of looking at all the parameters of the signal and analyse them, it is sufficient to know if the amplitude of a specific frequency crosses a predefined threshold, in this case the feature is only the amplitude of that frequency, and the Feature Extraction method could be the Fourier Transform of the signal. Generally speaking, the input signal is of high dimensionality and can often be summarized by a set of features, or a feature vector, that capture the essence of the signal and can help the classification of the signal. For instance, a small number of Fourier descriptors can represent a time signal efficiently.

The classification part, is the part in which decisions are made, here the class of the signal is recognized. In our previous example, the predefined threshold is considered as the classifier of the signal. The classification part comes always after the feature extraction part where the classifier looks at the features of the signal, and according to predefined parameters, the classifier decides in which class the signal should be placed.

Last thing to note here, that to find the good features of signals and to determine the parameters of the classifier in a good shape, there should be a large number of signals that are known to be from different classes. By having these large number of signals it is easy to find what is common in the signals that are in one class, and what is different between the signals that have different classes.

3.2 Methods for Detecting EEG Signals

It can be said that most of the methods and techniques that were developed in Machine Learning and Pattern Recognition for extracting features and detecting and classifying signals were used in the BCI field. In literature it can be seen that people have used simple techniques in analysing EEG signals like using Pearson's correlation for classifying P300 signals [14] , while others have used a complex techniques, like Hidden Markov Models and Support Vector Machines [15].

For detecting ErrPs, Schmidt et al. have used the linear discriminant analysis [16], and Llera et al. have used the logistic regression model [17]. However Combaz et al. have tested both the Fisher Linear Discriminant Analysis (FDLA) and the linear Support Vector Machines (linear-SVM) for detecting the ErrPs. Combaz et al. have found that the FLDA shows more balanced performances although the linear SVM seems to outperform the FDLA for the global accuracy [5]. Millan and his group have used the Mixture of Gaussians in most of their research about the ErrPs [13] [6]. It is important to note that all the previously mentioned research used the signals as an input to the classifier after down-sampling it without any feature extraction, the features were the amplitudes of the signal at different sample points.

In our analysis, we have used the both the Gaussian classifier and the Mixture of Gaussians with clustering (described next) to classify the ErrP signals that we have. In general, except when the PCA was used, the features that we considered are the amplitudes of the signals directly after down-sampling the signals.

3.3 Gaussian Classifier

The Gaussian classifier is based on the idea of Hypothesis Testing when the data are assumed to be normally distributed. If the data are to be classified into only two classes, then the Binary Hypothesis Testing is used. In this approach it is assumed that there is a null hypothesis and an alternate hypothesis, and each has its own parameters.

If the null hypothesis H_o has a posterior probability $Pr[H = H_o|\mathbf{y}]$, and the alternate hypothesis H_1 has a posterior probability $Pr[H = H_1|\mathbf{y}]$, and assuming that cost of the two types of errors is equal for simplicity, then

$$Pr[H = H_o|\mathbf{y}] \gtrless_{H_1}^{H_o} Pr[H = H_1|\mathbf{y}] \tag{3.1}$$

According to Bayes Rule the posteriors can be written as:

$$Pr[H = H_o|\mathbf{y}] = \frac{p(\mathbf{y}|H_o)}{p(\mathbf{y})} \cdot Pr[H_o] \tag{3.2}$$

Using Eq. 3.2, Eq. 3.1 could be written as:

$$\frac{p(\mathbf{y}|H_o)}{p(\mathbf{y}|H_1)} \gtrless_{H_1}^{H_o} \frac{Pr[H_1]}{Pr[H_o]} = \frac{p_1}{p_o} \tag{3.3}$$

As mentioned previously, the classifier is called Gaussian because:

$$\begin{aligned} H_o &: \mathbf{y} \sim N(m_o, \Lambda_o) \\ H_1 &: \mathbf{y} \sim N(m_1, \Lambda_1) \end{aligned} \tag{3.4}$$

Where m is the mean of the distribution and Λ is the covariance matrix, it should be non-singular, and its dimension is equal to the dimension of the

data vector.

In general, the Gaussian distribution for $N(m, \Lambda)$ is written as:

$$L(y) = \frac{1}{(2\pi)^{N/2}|\Lambda|^{1/2}} \exp[-\frac{1}{2}(y-m)^T\Lambda^{-1}(y-m)] \tag{3.5}$$

And because using the assumption in Eq. 3.4, the likelihood of the data from the null class can be written as in Eq. 3.6 and from alternative class can be written as in Eq. 3.7:

$$p(\mathbf{y}|H_o) = \frac{1}{(2\pi)^{N/2}|\Lambda_o|^{1/2}} \exp[-\frac{1}{2}(y-m_o)^T\Lambda_o^{-1}(y-m_o)] \tag{3.6}$$

$$p(\mathbf{y}|H_1) = \frac{1}{(2\pi)^{N/2}|\Lambda_1|^{1/2}} \exp[-\frac{1}{2}(y-m_1)^T\Lambda_1^{-1}(y-m_1)] \tag{3.7}$$

Finally substituting Eq. 3.6 and Eq. 3.7 into the Eq. 3.3, the resulting inequality takes the following form:

$$1/2\ln(\frac{|\Lambda_0|}{|\Lambda_1|}) - 1/2(y-m_1)'\Lambda_1^{-1}(y-m_1) + 1/2(y-m_0)'\Lambda_0^{-1}(y-m_0) \gtrless_{H_0}^{H_1} \ln(\frac{p_0}{p_1}) \tag{3.8}$$

Special Cases

The first special case appears when the dimensions are assumed to be independent, i.e., the covariances between different dimensions are zeros. In this case the shape of the covariance matrix will be:

$$\begin{pmatrix} \sigma_1^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & 0 & \sigma_N^2 \end{pmatrix}$$

Where σ_k^2 is the variance of the kth dimension, and N is the number of dimensions.

In this special case (the covariance matrix is diagonal), Eq. 3.8 can be simplified using the following equation:

$$\begin{aligned} \ln(|\Lambda|) &= \ln(\prod_{i=1}^{N}(\sigma_i)^2) \\ &= 2\sum_{i=1}^{N}\ln(\sigma_i) \end{aligned} \tag{3.9}$$

To make the equations much simpler, let x be equal to the following:

$$x = (y - m). \times [\frac{1}{\sigma_1} \cdots \frac{1}{\sigma_N}] \tag{3.10}$$

Where $.\times$ is the element by element multiplication. Then:

$$1/2(y - m)'\Lambda^{-1}(y - m) = xx' \tag{3.11}$$

Finally Eq. 3.8 becomes:

$$\ln(p_1) - 1/2||x_1||_2^2 - \sum_{i=1}^{N}\ln(\sigma_{1i}) \gtrless_{H_0}^{H_1} \ln(p_0) - 1/2||x_0||_2^2 - \sum_{i=1}^{N}\ln(\sigma_{0i}) \tag{3.12}$$

The second special case of the Gaussian classifier comes when it is assumed that the covariance matrix is equal to identity for both classes, that is:

$$\Lambda_1 = \Lambda_2 = \mathbf{I} \tag{3.13}$$

Here, it is assumed that the difference between the classes is the only the mean. Actually having covariance equal to the identity matrix means that the variance of each dimension is equal to one and the covariance between the dimensions are zero, that is, they are uncorrelated.

Using the case of identity matrix, Eq. 3.5 will reduce to:

$$L(y) = \frac{1}{(2\pi)^{N/2}}\exp[-\frac{1}{2}(||(y - m_1)||_2^2)] \tag{3.14}$$

Where, $||x||_2$ is the second norm of x. Finally, by substituting Eq. 3.14 into the inequality Eq. 3.3 the result will be:

$$\frac{1}{2}[||y - m_1||_2^2 - ||y - m_o||_2^2] \gtrless_{H_1}^{H_o} \ln(\frac{p_1}{p_o}) \tag{3.15}$$

To see the idea of the classifier, let's assume that the prior probabilities are equal ($p_1 = p_o$), by doing so the above Eq. 3.15 becomes:

$$||y - m_1||_2^2 \gtrless_{H_1}^{H_o} ||y - m_o||_2^2 \tag{3.16}$$

It is clear from Eq. 3.16 that the classifier depends on measuring distance. In this simplest case the classifier is called the *minimum-distance* classifier. This classifier simply determines the distance between any data point and both the mean of the first class, and the mean of the second class. Then it assigns the data point to the class which has the nearer mean.

3.4 Mixture of Gaussians

The idea of Mixture of Gaussians is based on the Gaussian classifier described in Section 3.3. But here instead of assuming that each class has a Gaussian distribution with a specific mean and covariance matrix, in one class, each subgroup of data is assumed to be normally distributed with a specific mean and specific variance. Then the overall distribution of the data within one class becomes a mixture of such Gaussian components. In this case, in each class we could have more than one label (indicating the mixture component) and if any given data point is classified to any subgroup (i.e., mixture component) of a class, then this data point is considered to be from this class.

Because the number of data points in one class could be small to estimate the covariance matrix efficiently, dividing these points into subgroups where each subgroup has its own covariance matrix makes the estimation of the covariance more unrealistic. In this case, all the data points from one class, after defining the mixture component, could be used to estimate a covariance matrix, and then, this covariance matrix is used as a common covariance matrix for all the subgroups (all the Gaussians) of a specific class.

According to the above, if we let C_k to be the class-conditional probability density function, then the activity α_k^i of the ith prototype (mixture component or subgroup) of class C_k for a sample x is given by:

$$\alpha_k^i(x) = (\pi|\Lambda_k^i|)^{-1/2} \exp(\frac{-1}{2}(x-\mu_k^i)^T(\Lambda_k^i)^{-1}(x-\mu_k^i)) \tag{3.17}$$

Where μ_k^i and Λ_k^i is the center and the covariance of the ith prototype of class C_k respectively. In the case of common covariance matrix for all the prototypes in class k, $\Lambda_k^i = \Lambda_k$. By finding the activity α_k^i, the posterior probability of x to be in the class C_k is given by:

$$p(C_k|x) = \frac{\sum_{i=1}^{N_p} w_k^i \alpha_k^i(x)}{\sum_{k'=1}^{M} \sum_{j=1}^{N_p} w_k^i \alpha_{k'}^j(x)} \tag{3.18}$$

Where N_p is the number of the prototypes in the class k, w_k^i is the weight of the ith prototype in the k class, and M is the number of classes.

Note that in our work M was equal to two because we have only two classes (error and correct signals), also for simplification, we did not use the Expectation Maximization method for finding the mean and the variance of the clusters, instead, we estimate the mean and the covariance of data directly after clustering. The weight of each prototype was found using the amount of data in this prototype from training data.

3.5 Clustering

Clustering is the operation of dividing a group of data points into subgroups, these subgroups are called clusters. Each cluster is given a label and the samples that are in the cluster are given the same label. Here instead of dealing with each data point individually, it is possible to deal with a limited and manageable number of groups.

Clustering aims to expand the uniqueness of the signal in an efficient way by transforming the pattern of the signal into a sequence of labels, also it can be considered as a tool to reduce the size of the data. There are many theories that perform the clustering operation. In this work both the K-means clustering and the Fuzzy C-means clustering are considered.

3.5.1 K-means Clustering

K-means clustering is an algorithm used for clustering data. In this algorithm clustering (grouping) the samples is based on the distance, i.e., any sample

in the dataset is assigned to the subgroup (or cluster) that it is nearest to. K-means clustering is a simple algorithm consisting of the following three steps:

1. Choosing K random samples and considering them as the means of the clusters.

2. Assigning the other samples to the clusters according the distance. By using the $||sample - mean||$ equation we can find the nearest mean to the sample.

3. Re-estimate the mean of the cluster considering all the points assigned to the cluster. This estimation is performed using $\frac{\sum_{i \in n} x_i}{n}$ where n is the number of samples in the cluster

4. Return to step 2.

3.5.2 Fuzzy C-means Clustering

What distinguishes Fuzzy C-means is that it assigns gradual memberships of the data points in the cluster, instead of assigning the data points completely in one cluster as the Hard C-means theory does. [18]

If the data points are given the following symbols:

$$X = \{x_1, x_2, x_3, ..., x_n\} \tag{3.19}$$

And clusters are:

$$\Gamma_1, \Gamma_2, ..., \Gamma_c \tag{3.20}$$

Then it is possible to define the degree of membership as U, where u_{ij} is a degree of membership of the jth data point into the ith cluster. The degree of membership is given a value between 0 and 1, where zero degree of membership means no membership, and 1 means full membership.

Two constrains must be under consideration when studying Fuzzy C-means. The first states that there must be no empty cluster, this is clarified by the following equation:

$$\sum_{j=1}^{n} u_{ij} > 0, \qquad \forall i \in \{1, ..., c\} \tag{3.21}$$

The second constrain states that each datum receives the same weight in comparison to all the other data, this appears in the following equation:

$$\sum_{i=1}^{c} u_{ij} = 1, \qquad \forall j \in \{1, ..., n\} \tag{3.22}$$

The Fuzzy C-means algorithm depends on the Objective Function, this function is defined as follows:

$$J_f(X, U, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{3.23}$$

where d_{ij} is the distance between the i center and the j element. m is the weighting exponent and it is usually equal to 2 for the Fuzzy C-means Clustering.

Note that the distance is used as a parameter of similarity between the data and the cluster also note that the membership is inversely proportional with the distance. It is easy to be aware that the best result of clustering occurs when the highest value of membership u_{ij} encounter the smallest value of distance d_{ij} so the objective is to minimize the squared distance of data points to their cluster centers and to get the maximum degree of memberships.

The algorithm that is used to get the best result of Fuzzy C-means Clustering is called the Alternating Optimization (AO) Scheme, the is u_{ij} are optimized for fixed cluster centers, then the cluster centers are optimized for fixed memberships as the equations clarify

$$U_\tau = J_U(C_{\tau-1}) \tag{3.24}$$

$$C_\tau = J_C(U_\tau) \tag{3.25}$$

The J_C and J_U are obtained by differentiating the Objective Function J_f and make it equal to zero. By doing so the following equations evolved:

$$u_{ij} = \frac{d_{ij}^{\frac{-2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{\frac{-2}{m-1}}} \tag{3.26}$$

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{3.27}$$

Note in Eq. 3.27 that it does not depend only on the distance between the data points and their center but also it depends on the distance between data point and the centers of other clusters. Initially cluster centers are determined randomly before the first update of the membership equation Eq. 3.27.

Fuzzy C-means clustering has been used in many publications like [19] that used the Fuzzy C-means to maximize the separability of different signals after using the wavelet space to extract the feature of EEG signals.

Chapter 4

Detection of Error Related Potentials

Error related Potential (ErrP) is one kind of EEG signals that is generated due to the subject's perception of error [20]. Even though it is hard to detect this "single-trial" signal, its discovery has urged many researchers to work on it because it could be used as a feedback in the BCI system to show if the system is working according to the patient's intention [5]. By having this feedback the system would be more robust.

This chapter discusses the concept of Error related Potential, what makes classifying this signal different from other EEG signals , how could it be used in the environment of Brain-Computer Interface and the advantages of using it. It also shows the work that was performed in our study to generate ErrP signals, compares the method that has been chosen with those used in the literature, mentions the classification techniques that have been applied, and compares the results with those in the literature.

4.1 Motivation for using ErrP

Despite the significant amount of research carried out in the field of BCI, still BCI is working mostly within the four walls of the laboratories and hospitals. BCI systems are not reliable enough to be connected to patients for daily use. One of the main problems is that the BCI systems have - as most systems - a probability of making errors.

Having many errors in systems recruited to help patients and disabled individuals and improve their life could be dangerous and may reach the degree of being fatal, because these systems could be used to control essential things in the life of the patients. So in such systems the number of errors should be as small as possible.

As mentioned in Chapter 2, Electroencephalogram-based (EEG-based) BCI systems depend on low-power EEG signals; this leads the worker in the field to excavate for finding the best features that are able to perfectly represent these signals and the best classifiers that can identify them.

Unfortunately, building an error-free classifier is impossible in practice; having classification errors is inevitable and a normal property of all the classifiers. So, whatever the researchers do, errors will not be completely eliminated, they will be just reduced in the best case.

However the BCI system could be joined with another system that monitors the decisions the BCI takes. If the BCI system makes non-logical decisions, the other system will either neglect them or modify them. Here instead of depending totally on the classifier of EEG signals in making blind decisions, the system becomes more clever by depending on other sensors.
For example a wheelchair that is controlled by a BCI system could be enhanced by a sensor to detect the walls. A "Move Right" order will not be applied as long as there is a block in the right side.

Another idea of combining could be done by depending on another biosignal, in this case the result of EEG classification is combined with a classification of another biosignal, like combining the EEG signals with the eye gaze [21], or combining SSVEP BCI with the heart rate variation [22]. In these cases the resulting systems are called hybrid BCI [23].

Hybrid BCI systems can also be implemented by combining more than one type of EEG signals, for example, one kind of EEG signals could be combined with Error related Potential (ErrP), which is another kind of EEG signals. To implement a robust classifier capable of classifying ErrP signals with a high accuracy, experiments should be designed to generate sufficient samples of these signals. Detecting such signals means errors could potentially be

corrected, error responses for the system could potentially be neglected, or the system could learn from these error to avoid them in the future.

4.2 ErrP and Adaptivity

One way to get an adaptive BCI system could be based on using the idea of ErrP signals. By having ErrP signals in a system based on binary-classification, all classified signals could be used to update the classifier.

In general, the system which discriminates between two classes, its output is either 0 or 1. But because it is not guaranteed that the classified signal is related to the true class (because every classifier has an error rate), this classified signal cannot be used to update the classifier (we mean by updating the classifier, changing its parameters according to the new signal) all the time.

If the ErrP is considered, then the probability that the classified signal is in its true class is high as long as there is no detection of ErrP (we say the probability is high, and not necessarily for sure to be in the true class, because the detection of ErrP is not perfect). Therefore the classified signals can be used to update the parameters of the classifier. By having this procedure the classifier can be made adaptive.

For example in the P300 speller case, the signal that is classified as P300 could be used again as an example of P300 signal to update the parameters of the classifier (retrain the classifier) as long as there is no ErrP signal, on the other hand, if there is ErrP the same signal could be used as an example of non-P300 signal to update the classifier. Furthermore, to have an online adaptive system (in real-time run), the classifier of the main signals should be simple (like a linear classifier) to be trained on online.

4.3 Review of recent work on ErrPs

Building a dedicated classifier for any kind of signals requires having samples of these signals to be used as examples for training the classifier. Moreover

the number of these samples should be large enough to have a robust classifier [24]. Therefore, to build a classifier for detecting error signals, many researchers have built experiments to generate these signals. These experiments are based on different interfaces, but their purpose are the same.

In his research Ferrez et al. has implemented a scenario in which the subject is asked to move a robot toward a specific target[13]. This target could be to the right or the left of the robot. The experiment was designed to study ErrP in isolation of other signals, because of that the subject was sending control commands manually (left/right buttons) not mentally. To generate a sufficient error signals they added an error probability to the system (a probability that the robot will move away from the target). Here they argued that the error signals that are generated in that experiment is due to the system and not due to the subject, i.e., if ErrP was generated that are because system failed to interpret the subject's command, and they call this kind of ErrP "Interaction ErrP".

In [6] Chavarriaga et al. designed an experiment over which the subject has no control (neither mental nor manual control). In this experiment the subject has to observe and criticize the performance of an external agent. It consists of a screen in which a square moves toward a specific target located three steps away. Each step has a specific probability (which is the probability of error) of going in the wrong direction (i.e., away from the target location).

Combaz et al. have depended on their previously implemented P300 speller to generate error signals [5]. Normally the P300 speller works pretty well when the number of trials is high, so in order to generate a sufficient amount of error signals, they decreased the number of trials.

In their experiment, Visconti et al. designed a P300 speller over which the subject has no control [7]. The subjects, at the beginning of the experiment, were asked to concentrate on a given letter at the beginning of each block, and they were told that the system was recognizing their attention. But actually, the system was programmed to choose the given letter with probability of 80% and a different letter (to generate an error signal) with probability of 20%. In their research, they considered the ErrP potentials to be generated after the screen shows the non-target letter.

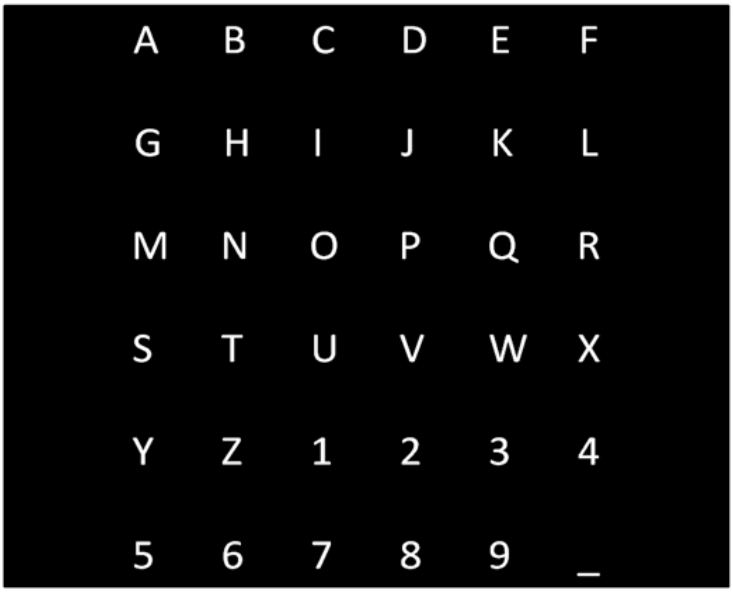

Figure 4.1: The shape of the P300 speller paradigm used to generate ErrP signals.

4.4 Generation of ErrPs using a P300 scenario

In this thesis, the first approach to error signals was based on the P300 speller. An experiment similar to the one performed in [7] was implemented.

Based on the Presentation Software® designed by the Neurobehavioral Systems company, a P300 speller, as shown in Fig. 4.1 consisting of 26 letters, 9 numbers and underscore was designed. The P300 speller was designed only to monitor generated ErrPs, so it can be called the "ErrP generator" instead of the P300 speller, because it has nothing to do with P300.

In this experiment, the subject was asked to focus on a given letter so that the system can understand the subject's intention and print that letter. But actually the system was giving the subject a letter to concentrate on, then showing the matrix with its flashes, and finally, the system was choosing a letter again without caring about the P300 signals of the subject. The error signals here could be viewed as "Interaction ErrP" [20], because the subject

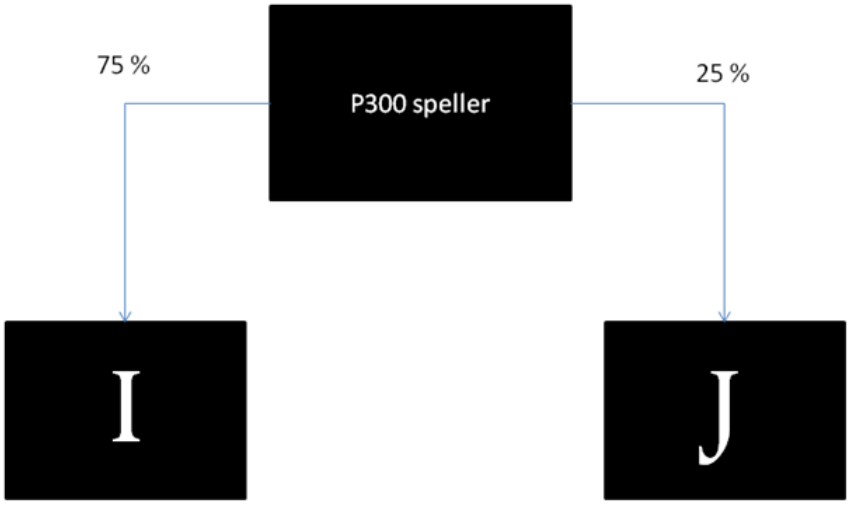

Figure 4.2: After one trial of flashing, the screen displays the correct letter with probability 75% or a letter next to it with probability 25%. This example shows the possible results of spelling the first letter.

thinks that he/she is involved in the experiment, and his/her aim is not just criticizing the system as in [6].

In a session of the experiment, the system shows the nine characters contained in the sentence "I_LOVE_SU", where each character is shown at the beginning of each block. After giving one letter the matrix starts to flash (each column and row flashes 5 times), then the result appears on the screen. The result could be the target letter with probability 75% or a different letter with probability 25% as shown in Fig. 4.2.

To make the experiment realistic and to make the subject trust on the system, the error letters were not chosen randomly, but most of them were chosen to be near the target letter. Because the target letters were chosen to be "I_LOVE_SU" in order, the error letters were "J9FUUK9MH" in order. By looking at the paradigm shown in Fig. 4.1, it could be seen that most error letters are exactly near the target letter.

The P300 speller was designed using the Presentation Software®. The experiment starts by showing the target letter for 2000 milliseconds, then the screen shows the matrix without flashes for 3300 milliseconds, after that the

matrix starts to flash; the flash period is 125 milliseconds and the off-period is 300 milliseconds. The matrix flashes for 5 trials (i.e., each column/row flashes 5 times) after that a letter is displayed to the subject (it could be the target letter or another letter), on the screen for 2000 milliseconds, finally the screen goes black for 2000 milliseconds and the experiment continues with a new letter. It is important to note that the matrix flashes in a random order; where all the rows/columns have the same probability of being flashed at the beginning.

Because the aim of this P300 speller was not to analyze the P300 signals, no more restrictions were made; like avoiding two flashes in series from one column/row (this happens when the last flash of a trial is x and the first flash of the next trial is also x). This case was avoided in other pieces of researches, because they found that two P300 components signals cannot be generated well from two flashes directly after each other.

4.5 ErrP shape in P300 scenario

In the literature it is common to examine the grand average of the error-minus-correct, i.e., the difference between the signal after an error response (or feedback) and the signal after the right response/feedback, to compare the shape of the error signals with other studies.

Ferrez et al. found that the error signal has a first sharp negativity (Ne) around 270 ms after the feedback. A latter positive peak appears between 350 and 450 ms after the feedback. Finally a negative peak appears around 550 ms after the feedback. But they suggest that the distinctive feature of the ErrP is the negative peak that appears around 250 ms after the feedback [13]. It is important to note that in their experiment the subjects sent control commands manually.

Chavarriaga et. al. found that when the subjects were just criticizing the moving square in the screen [6], the ErrP signal has approximately similar shape with the experiment mentioned above. It has one negative peak around 270 ms after the feedback and two positive peaks around 200 ms and 330 ms after the feedback.

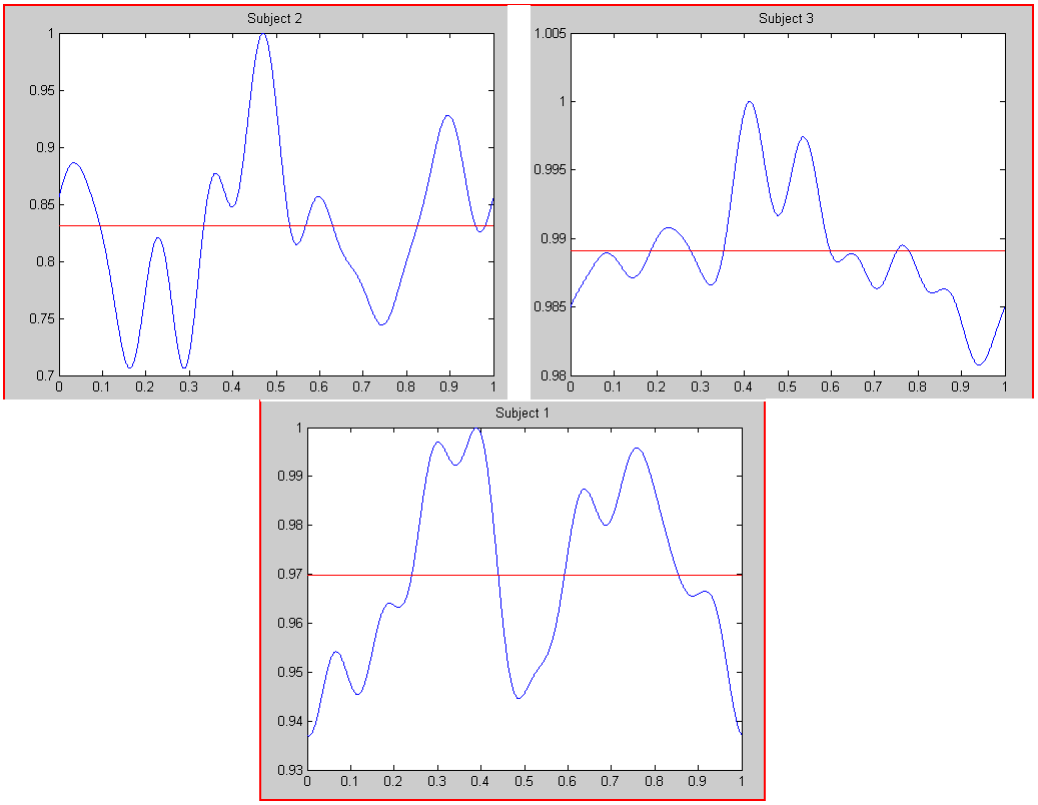

Figure 4.3: Error minus correct for the three subject

Using the P300 speller, Combaz et. al. and Visconti et. al. have found similar results of having a first negative peak around 300 ms after the feedback and positive one around 400 ms after the feedback [5], [7]

In this thesis, as shown in Fig. 4.3, it is found that the similarity the signals share is the positive peak around 400 ms after the feedback. The difference in the amplitude could indicate that some subjects have more attention than others. This positive peak around 400 ms is similar to the results of previous studies about ErrP in P300 speller [5], [7].

4.6 Processing of ErrP

In this thesis, ErrP signals were generated using the experiment described in Section 3.3. The signals were acquired using Biosemi® device according to the well-known 10/20 international system and saved in files for later off-line processing.

The signals were acquired from three subjects, and each subject is new to the idea of BCI, and each subject participates in one session. Each session lasts around 15 min for typing the sentence "I_LOVE_SU" only once. The preparation of the experiment for each session lasts around 15 min.

Signals were acquired using 9 electrodes, but in processing only the signal form the Cz electrode was used, because most of the approaches in the literature consider the anterior cingulate cortex (ACC) [6], [13] as a source of the ErrP signals. To label the error and the correct signals appropriately, different triggers were used in the experiment after the feedback. In this work the beginning of the error signals is considered to be the instant in which the error letter (a letter different than the target letter) is displayed on the screen of the computer. Showing correct letters (target letters) as a feedback on the screen is also accompanied by a trigger sent by the system to the file. Triggers given to error feedback is different than triggers given to correct feedback.

Before splitting the signals a Common Averaging Reference (CAR) referencing was applied [6]. Error and correct signals were taken as the signals recorded in the 1-second interval after the beginning of the corresponding trigger signal (2048 sample points). Before classification two processes had considered. First, signal components were filtered between $2Hz$ and $10Hz$. Then signals were downsampled from 2048 to 256 sample per second.

4.7 Classification and Results of P300-based ErrP

For classifying the acquired ErrP signals, the generated data were then split into three parts to apply 3-fold testing, each time a 2/3 portion of the data

was used for training the feature extractor and the classifier, and the other 1/3 for testing.

The Gaussian classifier mentioned in Chapter 3 was used as a classifier. The signals were used as an input to the classifier as a time sequence after being down-sampled. The classifier was tested in three different forms, identity covariance matrix, diagonal covariance matrix and the general case were the covariances among the dimensions was estimated. But it is found that the diagonal case is the best and its results are mentioned in this thesis.

In these problems the global accuracy cannot be considered because the data are not balanced, i.e., the correct signals are much more than the error ones. Because of that, to understand the strength of the classification the confusion matrix showing the True Positive, True Negative, False Positive and False Negative should be considered. Table 4.1 and Table 4.2 show the results of the first two subjects. The global accuracy shows that the accuracy of the classification is 77% for the first subject, this accuracy is approximately similar to what other people got in their researches [13], [20]. But for the second it is 66%. However, as shown in Tables 4.1 and 4.2, the accuracy of classifying error signals is low, and that is due to the small number of samples used in the classification.

Table 4.1: Confusion Matrix of the first subject

	Correct (real)	Error (real)
Correct (classified)	4	2
Error (classified)	1	2

Table 4.2: Confusion Matrix of the second subject

	Correct (real)	Error (real)
Correct (classified)	4	2
Error (classified)	2	1

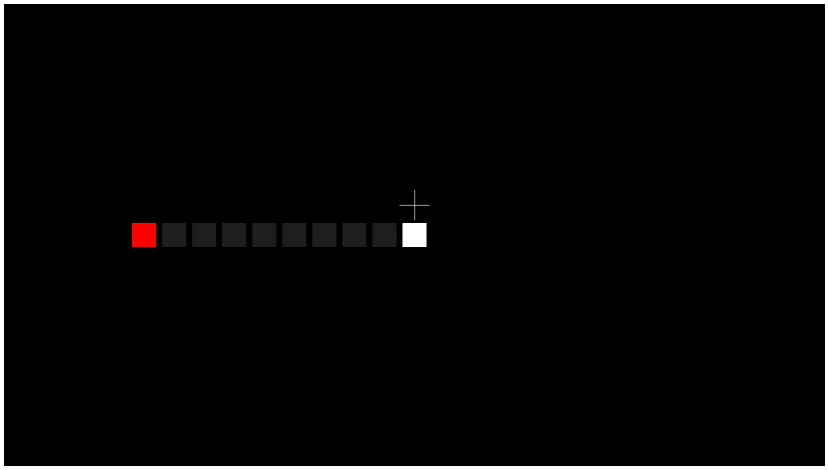

Figure 4.4: The beginning of the experiment when the target is to the left.

4.8 Generation of ErrP Signals using the Moving Box Scenario

The scenario of moving box is a modified version of that found in [6]. In this thesis it is implemented to compare its results with the classification results of ErrP in P300 speller. The experiment consists of a box moving toward a specific target. The subject should imagine to move the box reach its target. As long as the box is moving toward the target, the acquired signals are considered as correct, but if the box moves in the opposite direction an ErrP signal is expected to be generated. This experiment starts showing the moving box, the cross on which the subject should concentrate, and the target on the left or on the right. Fig. 4.4 and Fig. 4.5 show the view of the experiment.

The timing of the experiment was designed in such a way to comfort both the subject and facilitate the process of extracting the signals. First, the target flashes three times where each flash lasts for 150 ms, the aim of this flashing is to attract the subject's attention. Then the view in Fig. 4.4 or Fig. 4.5 stands for 3000 ms. Then the box starts to move, and it stays in each position for 2000 ms. Each session contains ten runs, when the box reaches its target a run ends and a new run starts.

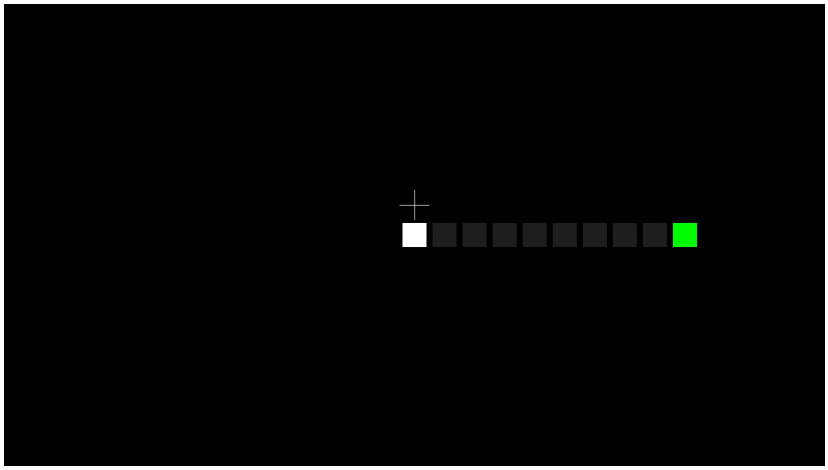

Figure 4.5: The beginning of the experiment when the target is to the right.

4.9 ErrP Shape based on the Moving Box Scenario

Four subjects accepted to join the experiment. The experiment consists of two different runs: one with error probability of 20%, i.e., the probability for the box to move in the opposite direction is 20%, and the other with 40% probability of error.

Figures Fig. 4.6 and Fig. 4.7 show the error-minus-correct difference for acquired signals from both runs. The difference of the signals from the run with 20% error probability is shown in Fig. 4.6, and the one taken from the run with 40% error probability is shown in Fig. 4.7.

The pattern here is not similar to that of signals acquired under the P300 speller, but, for some subjects, it is possible to see that the difference is large right after the trigger. After that the signal seems to die out, this is particularly clear in the first and the fourth subjects' signals.

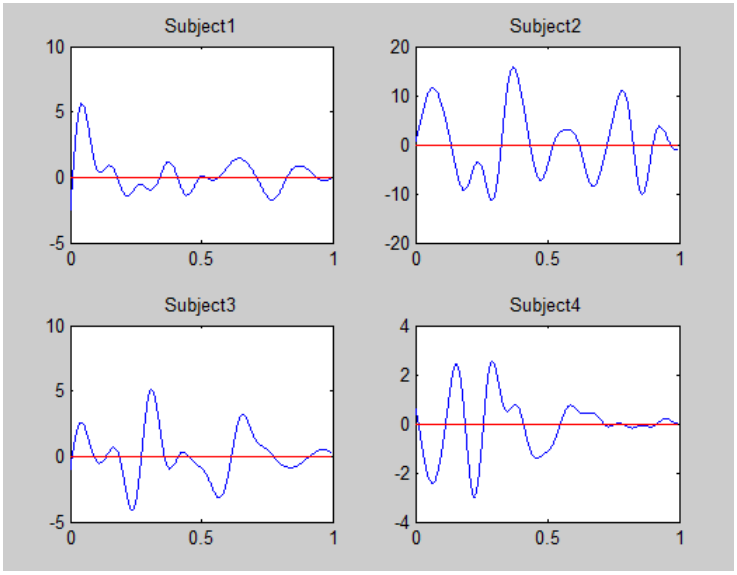

Figure 4.6: Average miss-minus-hit for the four subjects with 20% error probability

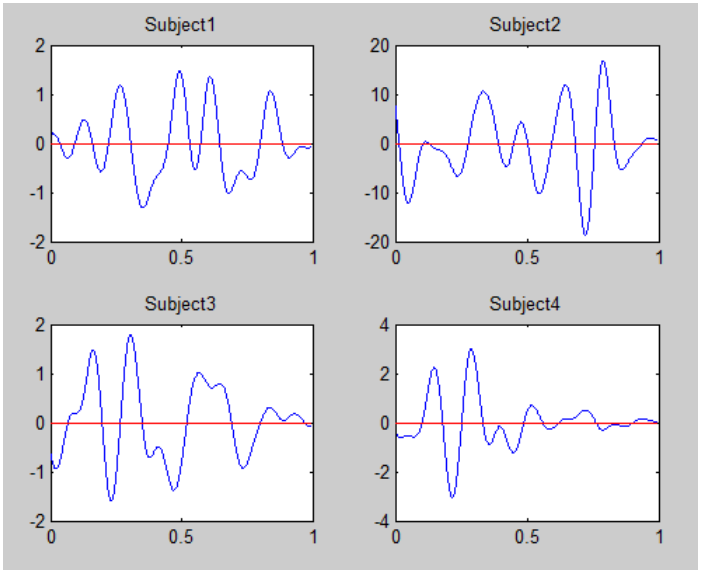

Figure 4.7: Average miss-minus-hit for the four subjects with 40% error probability

4.10 Classification and Results of Moving Box-based ErrP using the Gaussian Classifier

The same techniques that were used to process and classify the ErrP signals under the P300 speller, were also used to process and classify the signals under the moving box scenario. Table 4.3 to Table 4.6 show the confusion matrices of each subject in the first session of the 20% error probability.

In processing these signals, the interval where the correct and error signals seem to be different were considered. For signals acquired from the scenario with 20% error probability, the interval from 0 to 300 ms was considered. For signals acquired from the scenario with 40% error probability, the interval 300 ms to 600 ms after the trigger was considered. The reason for choosing such intervals is due to the difference between the correct signals and error signals seems to be larger at these intervals. The average of the error signals and the correct signals for both the run with 20% and the run with 40% can be seen clearly in both Fig. 4.8 and Fig. 4.9 respectively. The features that were entered to the classifier were the time amplitudes of the signal at different sample points.

Table 4.3: Confusion Matrix of the first subject

	Correct (real)	Error (real)
Correct (classified)	78	26
Error (classified)	20	4

Table 4.4: Confusion Matrix of the second subject

	Correct (real)	Error (real)
Correct (classified)	88	30
Error (classified)	26	12

As shown in tables from Table 4.3 to Table 4.8, the classifier's results for classifying correct signals are good and trustful, but unfortunately the classifier fails to classify error signals and that could be due to the unbalance

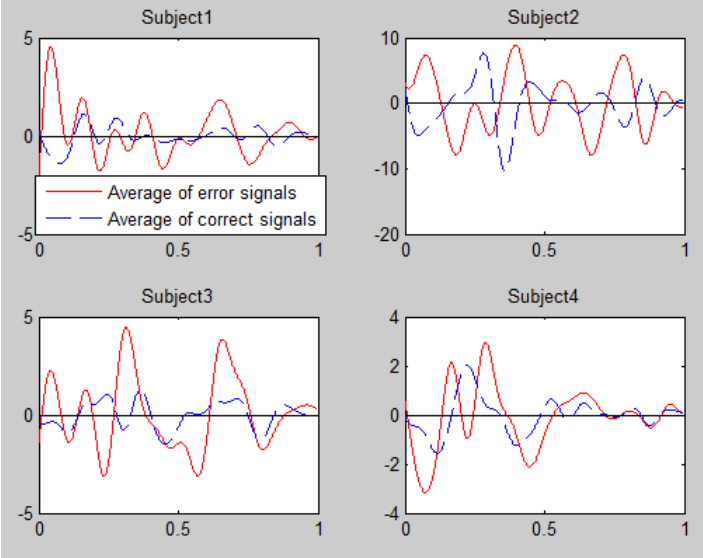

Figure 4.8: Average correct signals and average of error signals for the four subjects with 20% error probability

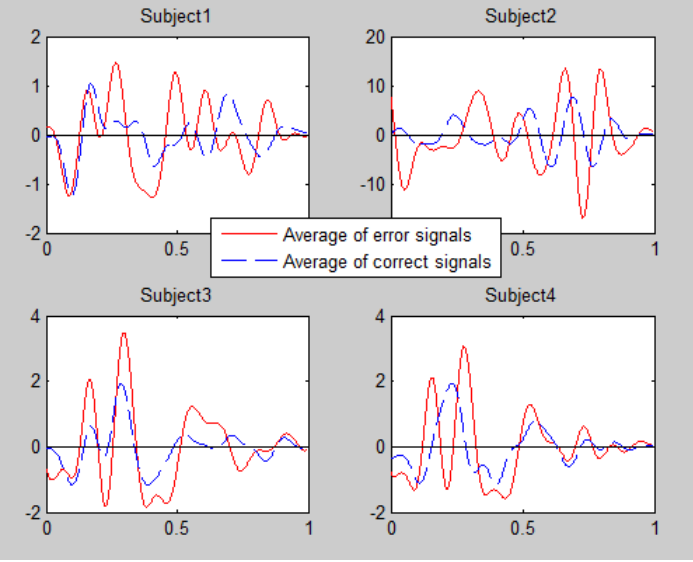

Figure 4.9: Average correct signals and average of error signals for the four subjects with 40% error probability

Table 4.5: Confusion Matrix of the third subject

	Correct (real)	Error (real)
Correct (classified)	81	35
Error (classified)	25	11

Table 4.6: Confusion Matrix of the fourth subject

	Correct (real)	Error (real)
Correct (classified)	96	21
Error (classified)	18	19

nature of the data, i.e., the error signals are much smaller than the correct ones. The best results for error signals can be seen in Subject 4 (slightly amount above the chance), and this may be due to the concentration of this subject more than the others.

4.11 Classification and Results of Moving Box-based ErrP using Mixture of Gaussians

After finding that the Mixture of Gaussians did somehow a good job in classifying the ErrP data acquired by Chavarriaga et al. in their work [6], we tried to test this classifier on our data acquired using the moving box scenario described in Section 4.8. Here it is important to note that we have used the Fuzzy C-means clustering for finding the prototypes.

For classifying our data we used 5 prototypes (except for the data of subject 4_20 which did not have results before decreasing the number of prototypes to 4. Table 4.9 shows the results of classifying our data. It can be shown from the table that the results for the data which were acquired when the probability of error 20% are biased toward classifying correct signals. This sounds logical because the number of error samples are small when the probability of error is small, so the classifier would be biased toward the major portion of the data. While when the probability of generating error is high 40%, the classifier gives somehow balanced results but still the classifier cannot differentiate between the error signals and the correct signals well.

Table 4.7: General results of the first session with 20% probability

Subject	Global Accuracy	Error Accuracy	Correct Accuracy
1	64	16.7	75
2	64	31.6	74.6
3	60	30.6	69.8
4	75	51.4	82.1

Table 4.8: General results of the first session with 40% probability

Subject	Global Accuracy	Error Accuracy	Correct Accuracy
1	54	34	65.6
2	58	5.7	89.6
3	56	46.4	62.3
4	72	58.8	80.1

Table 4.9 shows the results when classifying depending on making each prototype have its own diagonal covariance matrix. Classifying the data based on having a common diagonal covariance matrix for all the prototypes in the cluster gives similar results. We found that the error rate of the correct signals is 0.3937 and for the error signals is 0.6170 when the covariance matrix is common for 1_20.

4.12 Millan's Group Data

The low accuracy of classifying error signals that we acquired using our system urges us to check the classification results on other datasets. We asked Millan's group which is working in the field of BCI for the possibility of using their data in our analysis. A general information about the data that we got can be found in [6] which describes the way the data were acquired, the scenario the used, the preprocessing and classification methods, and finally the results of classifying this data.

Actually, the scenario that we used in generating our error data is similar to the one that Chavarriaga et al. used [6]. The only difference that we noticed are the number of electrodes. They used 64 electrodes while we used

Table 4.9: Results of Classifying the data acquired in our work using the moving box scenario. The Mixture of Gaussians was used as a classifier, with the C-means clustering to find the clusters (5 clusters except 4_20 which has 4 clusters) and their samples. Each prototype has its own covariance matrix

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals
1_20	0.1880	0.7568
2_20	0.19831	0.7500
3_20	0.2881	0.7632
4_20	0.1154	0.7917
1_40	0.3981	0.4656
2_40	0.3182	0.5286
3_40	0.4604	0.4508
4_40	0.4434	0.5957

only 9 electrodes, even they only used 1 or 2 electrodes for classification, the idea of using Common Averaging Reference (CAR) makes all the electrodes affect the classification performance.

Having the data of Chavarriaga et al. encouraged us to find a good classifier for our data, and to check if better classification accuracy on Millan's data could be obtained by changing the classifier, adding a feature extraction method, or by changing the parameters in the preprocessing step.

Chavarriaga et al. have used the Mixture of Gaussians to classify their signals. They did not use any feature extraction method. They only downsampled the signal from 512 Hz to 64 Hz, used a band-pass filter of a range [1,10] Hz, and used the CAR filter.

The results that Chavarriaga et al. got in their research are shown in Table 4.10. Note that accuracy of classifying error signals is lower in general than classifying correct signals; this is because the number of error samples is less than the number of correct samples, so it seems that the classifier cannot capture the pattern of the error signals.

Table 4.10: Chavarriaga et al. results. The first column shows the subject number and the probability of generating an error stimulus in the interface (20% or 40%)

Subject_error probability	Error-rate of correct signals	Error-rate of error signals
S1_20	0.1416	0.2393
S2_20	0.2629	0.3411
S3_20	0.1782	0.3034
S4_20	0.2927	0.4167
S5_20	0.2569	0.4180
S6_20	0.3194	0.4891
S3_40	0.2415	0.3216
S4_40	0.4222	0.4933
S5_40	0.3732	0.3846
S6_40	0.4662	0.4261
Average	0.2955	0.3833

In our method to classify the signals that we got from Millan's group, we used the same parameters that were considered in Chavarriaga et al. work. These parameters are shown in Table 4.11 for each subject. The only difference in our classifier is the technique that was used to find the centre of the clusters (or prototypes) for the Mixture of Gaussians classifier. In their work, Chavarriaga et al. have depended on the Self-Organization Maps (SOM) to find the clusters and the data that belongs to these clusters. SOM is considered to be slow in comparison to other algorithms for clustering. In our work we have used K-means clustering for finding the prototypes and their data points.

4.13 Classification of the Millan's Data

As mentioned in the previous section, we performed the techniques that Chavarriaga et al. used in preprocessing the data. The only difference in

Table 4.11: The parameters that Chavarriaga et al. used for each subject. For all the subjects the frequency range is [1,10] Hz. At time 0 the feedback occurs.

Subject_error probability	Electrodes	Time Interval (seconds)
S1_20	FCz,Cz	[0.2,0.45]
S2_20	Cz	[0.15,0.6]
S3_20	FCz,Cz	[0.2,0.45]
S4_20	FCz	[0,0.6]
S5_20	FCz,Cz	[0.15,0.6]
S6_20	FCz,Cz	[0.15,0.6]
S3_40	FCz,Cz	[0.2,0.45]
S4_40	FCz,Cz	[0,0.6]
S5_40	FCz,Cz	[0.15,0.6]
S6_40	FCz	[0.15,0.6]

our work is changing the clustering method used in the classification. At the beginning we used K-means clustering to find the clusters and their samples. It is important to note also that we used the number of prototypes that Chavarriaga et al. have considered which is 6 prototypes. As Millan's group did, we entered the signal amplitudes as the features to the classifier after down-sampling.

Table 4.12 shows the results that we had after the classification. Notice that for those subjects who have results the classification accuracy is similar to those shown in Table 4.10. But the problem of having no results when we try to classify the data of some subjects makes our method worse than the one of Chavarriaga et al. . Actually, having no results is due to the existence of empty clusters, i.e., we assumed that we have 6 prototypes and each sample should be in one cluster, but we found that there are some clusters that do not have any sample.

Due to the problem of having no results, we try to reduce the number of clusters so that each cluster would have at least one sample. By doing so only for those subjects' data that did not have results in Table 4.12, we have obtained the results shown in Table 4.13.

Table 4.12: The results classifying Error signals and Correct signals. Mixture of Gaussians was used in addition to K-means clustering for finding the centre of each prototype. Here each prototype has its own covariance matrix. Each class has 6 prototypes and the used parameters for each subject are shown in Table 4.11

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals
S1_20	0.1093	0.3125
S2_20	No results	No results
S3_20	0.1641	0.2617
S4_20	No results	No results
S5_20	0.2197	0.4286
S6_20	No results	No results
S3_40	No results	No results
S4_40	0.3540	0.4966
S5_40	0.3752	0.3131
S6_40	0.4524	0.5127
Average	0.2791	0.3875

As it is shown in Table 4.13, reducing the number of prototypes generates results but the results are not good as compared to those in Table 4.10 except for S6_20 which gives better results when the number of clusters is equal to 3. Here it is important to notice that S4_20 and S3_40 did not give results until we set the number of clusters to 1, in this case the classifier becomes a Gaussian Classifier, in general, the results are reasonable except for subject S4_20 and S3_40

One of the major problems is the estimation of the covariance matrix in classifiers such as the Mixture of Gaussians. This is because the number of samples is small compared to the dimension of the data. So instead of giving each cluster its own covariance matrix, we can assign a common covariance matrix for all the prototypes in one class as shown in Eq. 4.1. This is the idea that was used in [13] to classify correct signals and error signals but they have used different scenario than the one which was used in [6].

Table 4.13: The results of classification after reducing the number of prototypes for those datasets that did not have results in Table 4.12.

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals	# of mixtures
S2_20	0.2202	0.4580	3
S2_20	0.1978	0.4580	2
S4_20	0.9956	0	1
S6_20	0.2244	0.5664	5
S6_20	0.2756	0.5398	4
S6_20	0.2756	0.4336	3
S6_20	0.4231	0.3363	2
S3_40	0.9982	0	1

$$\Lambda = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu^*)^T(x_i - \mu^*) \tag{4.1}$$

Where μ^* is the closest centre to the sample x, and n is the number of sample in one class.

Using Eq. 4.1 to find the common covariance matrix for all the clusters belonging to one class, we got the results shown in Table 4.14. In table 4.14 we have used a common covariance matrix to generate the results. As it is clear from Table 4.12 and Table 4.14, the results of Table 4.14 are better than the results shown in Table 4.12. Also the results for those data which have results, in general, outperform the results of Chavarriaga et al. shown in Table 4.10. However still the data which did not give results in Table 4.12, they do not have results in Table 4.14. Again this encouraged us to check if reducing the number of prototypes per each class will give us better results.

Table 4.15 shows the results the we got from the analysis performed in Table 4.14 but with less the number of clusters There are two things to note in Table 4.15, first: reducing the number of prototypes did not generate good results, and second: reducing the number of prototypes in case of each prototype has its own covariance matrix produced better results than reducing

Table 4.14: The results classifying Error signals and Correct signals. Mixture of Gaussians was used in addition to K-mean clustering for finding the centre of each prototype. Here each prototype has its own covariance matrix. The parameters of each subject are given in Table 4.11.

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals
S1_20	0.1458	0.2734
S2_20	No results	No results
S3_20	0.1752	0.1776
S4_20	No results	No results
S5_20	0.2511	0.3643
S6_20	No results	No results
S3_40	No results	No results
S4_40	0.2938	0.4899
S5_40	0.3371	0.3034
S6_40	0.3238	0.3131
Average	0.2545	0.3203

the number of prototypes while having a common covariance matrix for all the prototypes in each class.

By decreasing the dimensions it may be possible to decrease the number of prototypes and at the same time have good accuracy.One way to solve the problem of the high dimensionality of the data depends on feature extraction methods. Depending on the Principle Component Analysis (PCA) dimensions are chosen according to the value of the eigenvalues, that is, the dimensions which has the large eigenvalue is chosen while the other are neglected.

Table 4.16 shows the eigenvalues of the covariance matrix of the first session of the second subject's dataset, we can say that the first 17 dimensions carry important information about the signals. So the dimension could be decreased from 48 to 17 using the PCA. To check if the feature extraction does a good job, we took from the second subject in the second session 80%

Table 4.15: The results of classification after reducing the number of prototypes for those subjects that did not have results in Table 4.14

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals	# of mixtures
S2_20	0.9461	0.0534	3
S4_20	0.9956	0	1
S6_20	0.8141	0.0796	4
S3_40	0.9982	0	1

of the data for 10-fold cross validation to find the best parameters (the new number of dimension and the number of prototypes in each class), after running this test over 3, 4, 5 and 6 prototypes and the dimensions were chosen to be around 17. By looking at the results of the 10-fold cross validation classification, we found that 3 prototypes with 16 dimension gives the best results which is (0.3346 for correct signals 0.2972 for error signals). Here the reader should note that the classification does not always give results sometimes the problems of having empty cluster happens. In our test most of tests stop in the middle due to empty clusters. Even when we tried to get the best result by setting the dimensions to 16 and the number of prototypes to 3, the classification stopped in the middle.

After deciding on the parameters, 16 dimensions and 3 prototypes in each class. We train the classifier on 80% of the first session from the second subject data that used previously for the cross-fold validation, then test the classifier on the remaining 20% of the data to make our classification unbiased. The error rates were 0.4545 for correct signals and 0.2414 for error signals. This results cannot be compared with Millan's results because the train and test were performed on a different data coming from the same session. However it is important to note that the error rate of the error signals is smaller than the error rate of classifying the correct signals which contradicts with most of the results shown previously.

After getting the previous results and all of them better than the chance level, this pushed us to use this technique for all the data to compare it with

Millan's results and our previous results. Again using PCA and for reducing the dimensions to 16 dimensions for all the subjects, and using the Mixture of Gaussians for classifying the features, we found that the generated results are not as good as those found in Table 4.14. The results of classifying the reduced dimensions by PCA for all the data are summarized in Table 4.17.

It can be seen that the results, in average, outperforms the results that Millan's got, and they are pretty well for subjects $S1\_20$, $S3\_20$, $S6\_20$ and $S5\_40$. Although the number of dimensions were reduced and the number of prototypes were reduced form 6 to 3, still we could not have results for $S4\_20$ and $S3\_40$. Even though the reduction of dimension was based on $S2\_20$, it is useful to notice that in Table 4.16 $S2\_20$ and $S6\_20$ have results and the results of $S2\_60$ are better than those found in Chavarriaga et al. work as shown in Table 4.10.

4.14 Changing the parameters of Millan's data

Trying to get better results compared to what we got in Table 4.14, we tried to change the parameters of the preprocessing stage of the data that we have instead of using the same parameters that Chavarriaga et al. used. Here we have concentrated on changing the frequency range and the channels. We tried many values for each dataset by performing the 3-fold cross validation. Notice that the three fold cross-validation was performed only on the train data (80% of the first session) to chose the parameters for each subject.

According to the 3-fold cross-validation test for each subject, Table 4.18 shows the parameters that should be chosen for each subject. Notice that we focus on two electrodes, FCz and Cz while trying to change the other parameters, this is because in the literature they have found that the ErrP is detected from this region [25], [6]. The features that we have used are the amplitudes of the signal at different sample points.

In order to observe the statistical variability of our results, we run our approach 10 times, each time on a different random 70% portion of the test data. We display the average classification accuracy in each of these 10 runs for the 20% and for the 40% error scenarios in Figs. 4.10 and 4.11, respectively. The overall average performance for these two scenarios can be seen

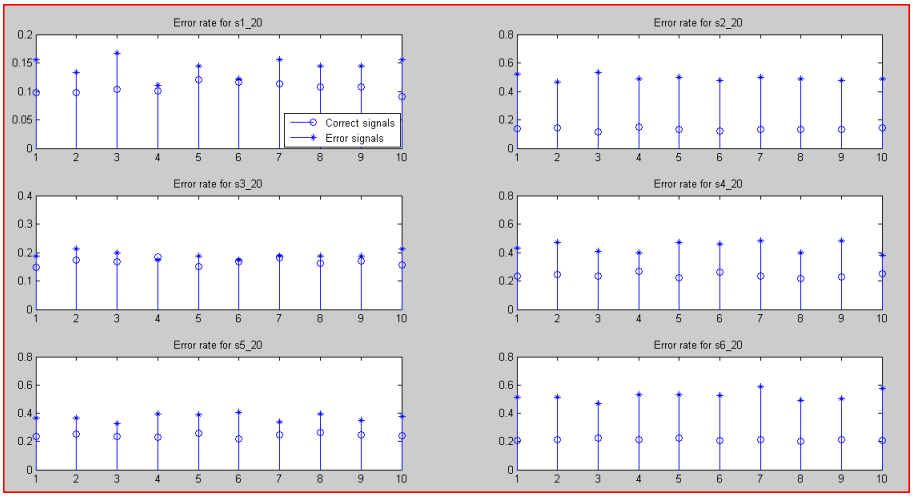

Figure 4.10: The results of classifying the random samples of the test session, each sample contains 0.7 of the test data. The probability of error in this set is 20%.

in Table 4.19. Based on Figs. 4.10 and 4.11, we observe that the variability of our results as a function of test data is at a reasonable level.

Although the results of some subject are worse than the results that Millan's group found, when we compare Table 4.10 and Table 4.19, still on the average we are outperforming their results. Also it can be noted that we have succeeded in getting results for all subjects.

4.15 Conclusion

We found that classifying our data did not give good results compared to the results that Chavarriaga got in his work, and that is due to two reasons, the first is the small number of samples, so the classifier could not capture the shape of the signals. The second could be the small number of electrodes used in acquiring these signals.

In classifying Chavarriaga's data, we got faster results, and that is due to the fast clustering technique that we used. On average our method outper-

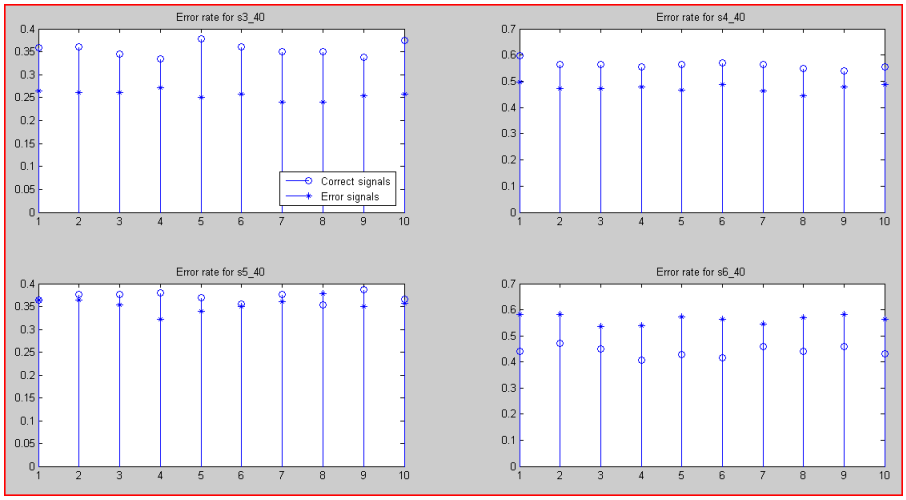

Figure 4.11: The results of classifying the random samples of the test session, each sample contains 0.7 of the test data. The probability of error in this set is 40%.

forms the one that Millan's used in his paper. And we look at Table 4.14, it is clear that our results for most of the subjects, on which we have results, outperform the results that Chavarriaga's got.

It is clear from changing the parameters of the classifier, changing the feature extraction methods and changing the parameters in the preprocessing step gives us different results: for some data they are good and for others they are not. So it is better to have specific parameters one subject instead of trying to generalize the classification and the preprocessing to all subjects.

Table 4.16: The eigenvalues of the covariance matrix of the second subject in the first and the second session, the probability of generating error in a session is 20%. The values are are multiplied by 10^3

First session	Second session
4.8672	0.1047
3.1804	0.0826
0.0987	0.0603
0.0797	0.0451
0.0622	0.0407
0.0428	0.0385
0.0412	0.0341
0.0303	0.0300
0.0268	0.0241
0.0217	0.0198
0.0161	0.0133
0.0114	0.0093
0.0084	0.0064
0.0071	0.0053
0.0057	0.0042
0.0030	0.0030
0.0011	0.0012
0.0008	0.0007
0.0004	0.0003
0.0003	0.0002
0.0001	0.0001
0.0001	0.0001
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
.	.
.	.
.	.

Table 4.17: The results that we got from classification of Millan's data using PCA as feature extraction and Mixture of Gaussians as classification. Here the number of dimensions was reduced to 16 and the number of prototypes in each class was 3. The covariance matrices of the prototypes belonging to one class are common. The parameters of each subject are shown in Table 4.11.

Subject_Error Probability	Error-rate of Correct signals	Error-rate of Error signals
S1_20	0.0296	0.3906
S2_20	0.5079	0.1450
S3_20	0.0887	0.3271
S4_20	No results	No results
S5_20	0.1614	0.5357
S6_20	0.3803	0.3009
S3_40	No results	No results
S4_40	0.2071	0.6577
S5_40	0.2343	0.4053
S6_40	0.8234	0.1303
Average	0.3041	0.3616

Table 4.18: The chosen parameters for each dataset. Notice that the time interval starts from the beginning of the trigger.

Subject_error probability	Electrodes	Frequency	Time Interval (sec.)
S1_20	FCz,Cz	[1,10]	[0,1]
S2_20	FCz,Cz	[1,20]	[0,1]
S3_20	FCz,Cz	[1,10]	[0.2,0.45]
S4_20	FCz,Cz	[1,20]	[0,1]
S5_20	Cz	[1,20]	[0,1]
S6_20	FCz,Cz	[1,20]	[0,1]
S3_40	FCz,Cz	[1,20]	[0,1]
S4_40	Cz	[1,20]	[0,1]
S5_40	FCz,Cz	[1,20]	[0,1]
S6_40	FCz,Cz	[1,20]	[0,0.1]

Table 4.19: Classifying results using Mixture of Gaussians and K-means clustering. Each class has 6 prototypes, each prototype has its own covariance matrix.

Subject_error probability	Error-rate of correct signals	Error-rate of error signals
S1_20	0.1055	0.1433
S2_20	0.1888	0.4261
S3_20	0.1661	0.1907
S4_20	0.2388	0.4370
S5_20	0.2435	0.3703
S6_20	0.2120	0.5233
S3_40	0.3554	0.2552
S4_40	0.5621	0.4751
S5_40	0.3706	0.3533
S6_40	0.4402	0.5633
Average	0.2883	0.3738

Chapter 5

P300 and Mechanical Devices

P300 paradigms can be used in controlling mechanical devices as well as in typing letters. This is usually carried out by changing the items in the interface, i.e., by placing any suitable items instead of usual characters.

In this chapter, some of the work that has been performed to use the P300-paradigms to control mechanical devices, will be discussed. Then the new interface that has been designed in this thesis, is described with its advantages.

5.1 Review of previous work

After the design that Farwell and Donchin implemented using the idea of P300 [8], many people working in this field tried to modify the paradigm in order to get better results and also to use this paradigm in different applications.

Requiring high accuracy and faster paradigm urges researchers to modify the paradigm in order to get larger P300 components. In [26], Aloise et al. implement a new paradigm based on the P300 idea. Instead of having the characters in the row and columns, they put the characters in vertices of a regular geometric figure that has a cross in the middle. Each figure contains 6 characters and there are 12 figures. By having this implementation, they argued that by having the cross on which the subject concentrates, eye contamination will be reduced and the signal-to-noise ratio of the P300 components will be larger.

Jin et al. have tried to mix between the motion stimulus and changing color in their P300 speller [27]. In the motion stimulus the flashing technique is performed by changing the position of the column/row for a specific period of time. It can be imagined like a vibration of the flashing column/row. In their research, this motion is performed accompanied with changing of the color of the moving column/row. They have found that their combined interface outperforms both the interface that depends on motion and the interface that depends on changing color.

In addition to the enhancement of the P300 paradigms, some people have used the P300 paradigm to control robots and mechanical devices, and that could be performed simply by using control commands instead of characters in the interface.

Blasco et al. have tried many interfaces in their research [10]. In one of them they specified the interface for controlling a robot. Using their interface the robot can move in four different directions and can have different orientations. In addition to modifying the interface, they added a confirmation page that appears after choosing the target, the aim of this confirmation page is to avoid errors. For example if the subject chooses the left direction the system will show the chosen direction, and then will ask the subject to confirm again using P300 by choosing YES or NO.

Long et al. used the idea of P300 in their wheelchair application, but they did not depend totally on their P300 paradigm to control the wheelchair, instead they have used the motor imagery with the P300 to control the speed. In their research if the foot movement is idle and a P300 is detected, the speed will decrease, while it will increase if a foot movement is detected while there is no P300. To generate a high signal-to-noise ratio P300 signal they used a flashing screen consisting of eight items [11].

Our work in this thesis, shows how it is possible to depend only on P300 to control a robot, and have a high accuracy. The following sections show the implementation and some results of this P300 paradigm.

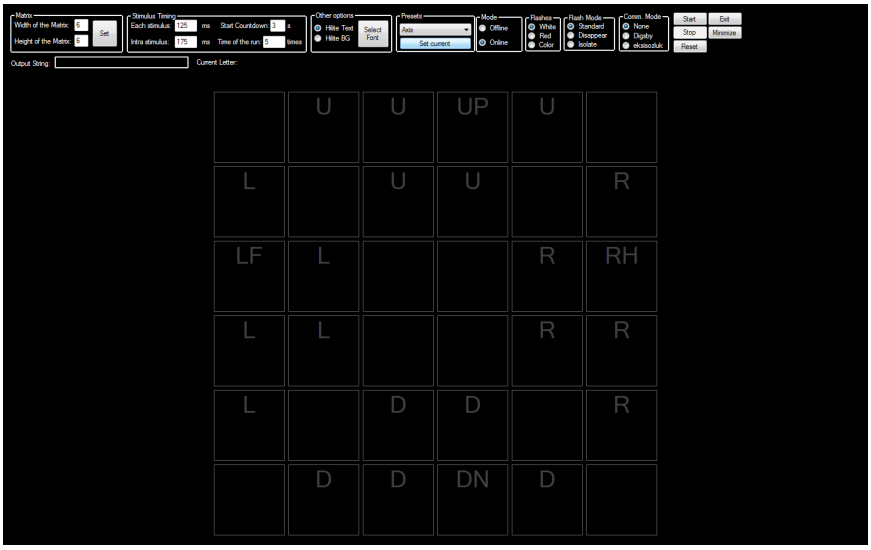

Figure 5.1: The modified P300 paradigm that was proposed to move a robot.

5.2 A new P300 paradigm for robot control

As mentioned previously, the P300 paradigm is used also for moving mechanical devices. Here in our research, we assume that it is required to control a robotic arm which has four directions; up, down, right and left. Normally, it is hard to have good P300 signals just by having only four directions in a paradigm. To have a good P300 signals the probability of the target stimulus should be low.

In this thesis, to our knowledge, a new idea was implemented in which the P300 paradigm could have both many items with only four choices. We mixed them in a way to have both, good P300 components and high accuracy. The paradigm is shown in Fig. 5.1.

As shown in the figure, there are only four choices, up "UP", down "DN", right "RH" and left "LF". The other items are "U"'s, "D"'s, "R"'s, "L"'s and free spaces stands for "No decision", i.e., no movement will take place. Notice that the paradigm has 6 rows and 6 columns, therefore the probability of the target row is 1/6 and the probability of the target column is also 1/6 as in the P300 speller.

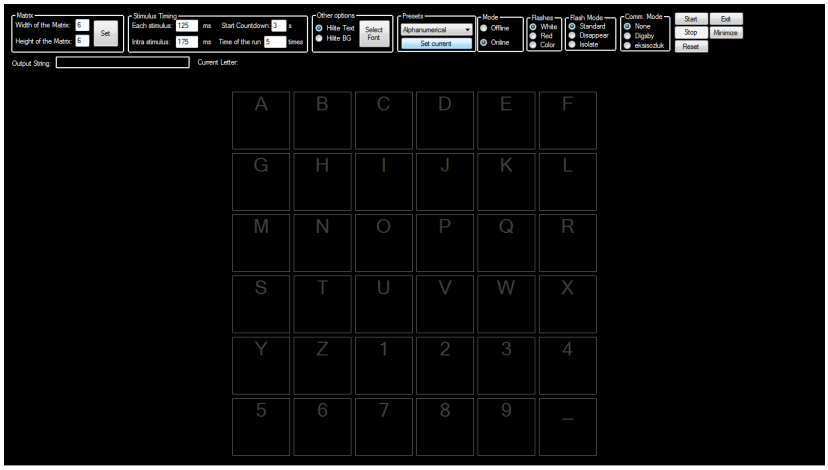

Figure 5.2: The P300 speller that Amcalar et al. have designed. The P300 paradigm shown in Fig. 5.1 is a result of some modifications in the former interface.

In this experiment we asked the subjects only to consider the four choices "UP", "DN", "RH" and "LF" for the four different directions. In this case, the robot will consider the "U" and "UP" as up, "D" and "DN" as down "R" and "RH" as right and "L" and "LF" as a left. And free space as no choice.

By knowing that the errors in the P300 speller are normally around the target letter [28]. It is easy to see that by having this paradigm, we can reduce the number of errors the P300 paradigms make, and a robust P300 paradigm can be implemented.

5.3 Experiments and Results

The implemented P300 paradigm for moving a robot is a modification of the P300 speller that Amcalar et al. designed in their work [29]. Fig. 5.2 shows the P300 speller interface that Amcaler et al. have implemented.

One subject joined the experiment. The experiment consists of online training, in which the subject is asked to type eight characters chosen by

the P300 speller that Amcalar et al. designed. Then an online test is performed on the same paradigm used for training. In the online test the subject was asked to print 19 characters from his mind. After printing the subject was asked whether the outputs were similar to those which he wanted to type.

After that a test on the modified paradigm that is shown in Fig. 5.1 was performed. The parameters of the classifier of the modified paradigm were also on the training that the subject did. In the online test the subject was choosing a direction and we were checking the result.

Between the test on the P300 speller and the test on the modified P300 paradigm no re-training was performed, i.e., the parameters of the system did not change, no pause was given, and the electrodes were not removed from their places; by doing so, we ensured a good comparison between the modified P300 interface and the P300 speller used in Amcalar et al.'s research.

As shown in Table 5.1 and Table 5.2, the modified paradigm was able to reduce the number of errors. Furthermore according to the subject it is better to have fewer number of errors even if the system offers fewer number of choices. It is important to note here that the "No decision" is not as costly as the error decision.

Table 5.1: The results of the test performed on the P300 speller after training

Number of chosen character	Correct decision	Error decision
19	6	13

Table 5.2: The results of the test performed on the P300 moving-paradigm after training

Number of chosen characters	Correct decision	Error decision	No decision
27	15	5	7

From an engineering perspective one way to compare the accuracy of the two systems could be using Cohen's kappa coefficient which is given in Eq. 5.1:

$$\kappa = \frac{C \cdot P_{cc} - 1}{C - 1} \tag{5.1}$$

Where, C is the number of classes, P_{cc} is the probability of the correct class.

From the Table 5.1, it is clear that $P_{cc} = 6/19$ for the P300 speller. However, from Table 5.2 P_{cc} is equal to $22/27$ if we consider the "No decision" as a correct decision, while it will be $15/27$ if the "No decision" is considered as an error.

Substituting the values of P_{cc} and $N = 5$ in the Eq. 5.1, it can be seen that for the P300 speller $\kappa = 0.2962$, and for the modified paradigm it is, in its best case i.e., considering "No decision" as correct, 0.7685, and in its worst case it is 0.4444 which is still better than the accuracy of the P300 speller. Note that if we are going to neglect the effect of "No decision" in the accuracy (assuming that it just makes the system slower), then we substitute $N = 4$ and $P_{cc} = 15/20$. In this case the result is $\kappa = 0.6667$.

To make the comparison fair, the Bit-rate should be also considered because systems with many choices used to have many errors but their Bit-rates are high. The Bit-rate equation is given in Eq. 5.2.

$$B = log_2 N + P log_2 P + (1 - P) log_2 \frac{(1 - P)}{(N - 1)} \tag{5.2}$$

Where P is the average of the correct classified symbols, and N is the number of choices.

Again by substituting the values in Eq 5.2, we can see that for the P300 speller $B = 0.7607$ bit-per-trial group, and for the modified paradigm it is, considering "No decision" as correct, 1.2603, and when "No decision" is considered as an error it is 0.4420. It is clear that the modified system, with "No decision" as correct outperforms the usual P300 speller.

It is important to notice that, assuming the "No decision" as a correct is too optimistic assumption and assuming it as a wrong is a pessimistic limit. According to Fig. 5.3 we can see that the "No decision" should be considered as something between the wrong decision and the correct decision if we are talking about bit rate, assuming that this "No decision" located in the

middle between the right and the wrong decision. From Fig. 5.3 it can be seen that the "No decision" choice makes the system slower.

To estimate the effect of this "No decision" correctly we should find the bit-rate considering the effect of "No decision" once, while neglecting it in the other. To find the effect of the "No decision" we should measure the time of the experiment then we divide the number of spelled symbols on the time to get the symbol/min (in our experiment we should divide 27 on the time of the experiment). While neglecting the "No decision" could be achieved by dividing the number of all except "No decision" on the time of the experiment (in our case we divide 20 on the time of the experiment). Then to have the bit/min for each case, we use the output Eq. 5.2 considering N = 4, P = (15/20), and multiply it with the symbol/min of the two cases mentioned above. In this case The estimated numbers can be used to indicate the effect of "No decision" in the bit-rate. Note that because we did not estimate the time of the experiment we could not do the comparison that we mention above.

However, if we are going to consider the cost of wrong decision and no decision for a person who wants to cross a street using a wheel chair, it is for sure that having a no decision as a result of the classification is much much better than choosing forward instead of stop.

Last thing to say about this topic is that this work could not be compared to other similar works regarding the accuracy and the bit-rate; this is due to the fact that for BCI experiments to be compared in a fair way all the conditions under which one experiment is held should be applied to the other experiment as well. But it is useful to note that other people proposed many ideas for dealing with the shape of the speller, in addition to those that were mentioned in Section 4.1, we mention here that [10] made a confirmation window after the window of the items to ensure that the subject has chosen the right item. Also [30] wanted to use only two items in their paradigm but because the idea of P300 is based on having a low probability of the target item they was forced to use 4 items. They also used auditory stimulus to support the visual one so that using four items could be feasible.

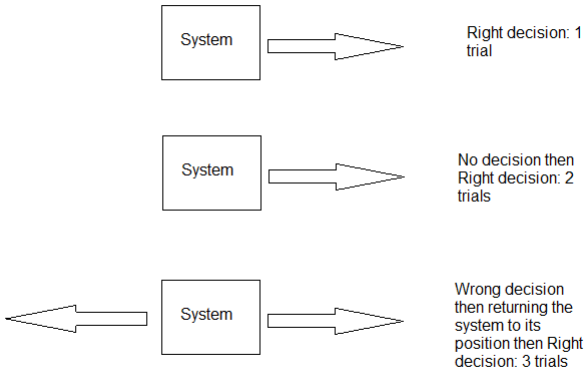

Figure 5.3: The figure shows that the "No decision" could not be considered as the correct decision which performs the command in one trial, and it could not be considered as the wrong decision which should be modified by and additional one trial. "No decision" is something between the correct and the wrong decision.

5.4 Accuracy, Bit-rate, and Comparison

Different BCI systems are compared by considering the accuracy and/or speed for specific applications. But this comparison would not be valid always because there could be different parameters in these different systems [31]. In the literature, it can be found that people consider both measures (accuracy and speed) for comparison, but they indicate that the transfer-rate (the speed) should not overshadow the importance of accuracy, and that the speed could be sometimes misleading [30].

It is hard to make a comparison between our system and other systems, and that is due to the nature of the experiment that had been done. The experiment had been done in an online fashion in which the decision was taken after a variable number of trials, i.e., normally when talking about bit-rate it is important to know the time that the system needs to generate a symbol, but in our case the number of trials were variable for each character, because it depends on the system certainty that the subject wants to spell a specific letter. Also, we have not recorded the time of the experiment in the

preliminary experiment we have presented here. By having both the number of symbols and the experiment time the average time that the system needs to generate a symbol could have been estimated.

What we did actually is fixing all the parameters (electrodes, subject, session time, and so.) of two different interfaces to check the capability of our new interface to avoid error and the results that we have give us an idea that the new interface is really a good design in P300 field to avoid error while having a reasonable bit-rate compared to the normal speller.

5.5 Conclusion

In general most of the errors that occur in the P300 interfaces are due to a miss in deciding the row or the column that generates the P300 components. In these cases the chosen character is usually near the target character. Such errors make it hard to connect the P300 paradigms to mechanical systems because wrong decisions could cause problems to the patients.

Because most of the errors are due to choosing the neighbour characters of the target, we decided to surround each target with characters that do the same job as the target and also place "No decision" items between the different control commands. As it is shown in the previous section, the modified P300 paradigm has the potential to offer a number of benefits in terms of accuracy and bit-rate as compared to the conventional P300 speller interface.

Chapter 6

Conclusion and Future Work

This chapter summarizes the work performed in this thesis, discusses the results of the experiments, and also proposes potential research directions that may enhance this work if they are considered in the future.

6.1 Summary and Conclusion

In this thesis, we have implemented two different scenarios for generating ErrP. One of these scenarios is similar to the P300 speller. And the other consists of a box that moves toward a specific target. We found out that the first scenario is slow and it is hard to get a lot of ErrP from it. However the second scenario is faster and it is easy to get more ErrP signals from it in less amount of time.

We tried to compare the ErrP signals that were generated from different subjects in each scenario. We found that the ErrP signals generated using P300 speller have some similar patterns and that they are similar to the signals generated from P300 speller in previous researches. Even though, the ErrP signals that were generated from the P300 speller are too few to be used for building a good classifier, we could compare our signals with other's, and we were able to check the feasibility of generating ErrP using this scenario.

On the other hand, we have generated a good number of ErrP signals using the moving-box scenario, but it was hard for us to find a similarity in the shape of the signal among the different subjects. Furthermore, the classifier could classify the correct signals with a good accuracy, but its classification

to error signals was near the chance level.

Also, we have processed and classified ErrPs data acquired by Millan's group. In our work we have used different methods to process and classify the data and shown that our methods, in general, outperform Millan's methods in both accuracy and speed of analysis.

Moreover, we have implemented a new interface based on the P300 theory that has a higher accuracy than the normal P300 speller. We assume that our new interface is for controlling a four-direction mechanical devices, and instead of having each item in the interface independent of the other, we make each group of neighbour items have the same job. By having this new interface we have shown that we can obtain a higher accuracy than the conventional P300 speller.

6.2 Future Work

The major motivating idea driving this thesis was the vision of an adaptive BCI system depending on the idea of ErrP. This adaptivity was to be performed in the P300 speller. To implement this adaptive P300 speller a good classifier of ErrP potentials should be first found. However to make a good classifier capable of classifying ErrP potentials in the P300 speller, a sufficient number of ErrP signals should be acquired from an experiment with P300 speller.

The problem of the P300 experiments is the long time it requires, and it is hard to keep the concentration of the subject high because the letters are chosen by the system not by the subject. In addition, if he tries to choose another character than that asked by the system, he will discover the idea of having a system which produce results independently from the subject decision. In this case ErrP signals cannot be considered.

So, rather than using a system in which the system generates errors randomly, it may be a good idea to evaluate ErrP detection on a real P300 system operating based on the EEG signals generated by the subject. In this scenario to make sure that wrong decisions made by the system are labeled

correctly, while leaving the subject free to choose what he/she wants to type. One can ask the subjects to press one of the keyboards' buttons when they see an erroneous feedback, but he would need to make sure that the keyboard command is well separated in time from the display of the system decision on the screen recording of the EEG data to be used in error classification.

Moreover, in this thesis, we have succeeded in implementing a new P300 paradigm with high accuracy for moving a mechanical device. This new interface could be connected and tested on a real mechanical device in the future.

Bibliography

[1] Bernhard Graimann, Brendan Z. Allison, and Gert Pfurtscheller. *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. Springer, 2011.

[2] Guangquan Liu, Gan Huang, Jianjun Meng, Dingguo Zhang, and Xiangyang Zhu. Unsupervised Adaptation Based on Fuzzy C-Means for Brain-Computer Interface. *First International Conference on Information Science and Engineering*, pages 4122–4125, 2009.

[3] Aldemar Torres Valderrama, Pavel Paclik, Mariska J Vansteensel, Erik J Aarnoutse, and Nick F Ramsey. Error probability of intracranial brain computer interfaces under non-task elicited brain states. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 123:2392–401, 2012.

[4] Fabrizio Beverina, Giorgio Palmas, Stefano Silvoni, Francesco Piccione, and Silvio Giove. User adaptive BCIs : SSVEP and P300 based interfaces. *PsychNology Journal*, 1:331–354, 2003.

[5] A. Combaz, N. Chumerin, N.V. Manyakov, A. Robben, J.a.K. Suykens, and M.M. Van Hulle. Towards the detection of error-related potentials and its integration in the context of a P300 speller brain-computer interface. *Neurocomputing*, 80:73–82, 2012.

[6] Ricardo Chavarriaga and José R Millán. Learning from EEG Error-Related Potentials in Noninvasive Brain-Computer Interfaces. *IEEE transactions on neural systems and rehabilitation engineering*, 18:381–388, 2010.

[7] G Visconti, B Dal Seno, M Matteucci, and L Mainardi. Automatic Recognition of Error Potentials in a P300-Based Brain-Computer inter-

face. In *Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course*, 2008.

[8] Farwell LA and Donchin E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.

[9] A. C. Lopes, G. Pires, L. Vaz, and U. Nunes. Wheelchair Navigation Assisted by Human-Machine Shared-control and a P300-based Brain Computer interface. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2438–2444, 2011.

[10] J.L. Sirvent Blasco, E. Iáñez, A. Úbeda, and J.M. Azorín. Visual evoked potential-based brain-machine interface applications to assist disabled people. *Expert Systems with Applications*, 39:7908–7918, 2012.

[11] Jinyi Long, Yuanqing Li, Hongtao Wang, Tianyou Yu, Jiahui Pan, and Feng Li. A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 20:720–9, 2012.

[12] Steven J. Luck. *An Introduction To The Event-Related Potential Technique*. The MIT Press, 2005.

[13] Pierre W Ferrez and José R Millán. You Are Wrong! Automatic Detection of Interaction Errors from Brain Waves. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005.

[14] Dean J Krusienski, Eric W Sellers, Franois Cabestaing, Sabri Bayoudh, Dennis J McFarland, Theresa M Vaughan, and Jonathan R Wolpaw. A comparison of classification techniques for the P300 speller. *Journal of neural engineering*, 3:299–305, 2006.

[15] Hye Lee and Seungjin Choi. PCA + HMM + SVM FOR EEG PATTERN CLASSIFICATION. In *Signal Processing and Its Applications*, 2003.

[16] Nico M Schmidt, Benjamin Blankertz, and Matthias S Treder. Online detection of error-related potentials boosts the performance of mental typewriters. *BMC neuroscience*, 13:19, 2012.

[17] A Llera, MAJ Van Gerven, V Gmez, O Jensen, and HJ Kappen. On the use of interaction error potentials for adaptive brain computer interfaces. *Neural networks : the official journal of the International Neural Network Society*, 24:19, 2011.

[18] Jose Valente de Oliveira Witold Pedrycz, editor. *Advances in Fuzzy Clustering and its Applications*. Wiley, June 5, 2007.

[19] M Murugappan, M Rizon, R Nagarajan, and S Yaacob. FCM clustering of Human Emotions using Wavelet based Features from EEG. In *4Th Kuala Lumpur International Conference on Biomedical*, 2009.

[20] Pierre W Ferrez and José R Millàn. EEG-Based Brain-Computer Interaction: Improved Accuracy by Automatic Single-Trial Error Detection. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 441–448. MIT Press, Cambridge, MA, 2008.

[21] Gert Pfurtscheller, Brendan Z Allison, Clemens Brunner, Gunther Bauernfeind, Teodoro Solis-Escalante, Reinhold Scherer, Thorsten O Zander, Gernot Mueller-Putz, Christa Neuper, and Niels Birbaumer. The hybrid BCI. *Frontiers in neuroscience*, 4:1–11, 2010.

[22] R Scherer, A Mohapp, P Grieshofer, G Pfurtscheller, and C Neuper. Sensorimotor EEG patterns during motor imagery in hemiparetic stroke paitients. *Bioelectromagn*, 9:155–162, 2007.

[23] J D R Millán, R Rupp, G R Müller-Putz, R Murray-Smith, C Giugliemma, M Tangermann, C Vidaurre, F Cincotti, A Kübler, R Leeb, C Neuper, K-R Müller, and D Mattia. Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and challenges. *Frontiers in neuroscience*, 4:1–15, 2010.

[24] Ethem Alpaydın. *Introduction to Machine Learning*. The MIT Press, 2004.

[25] Brendan Z Allison and Jaime A Pineda. ERPs evoked by different matrix sizes: implications for a brain computer interface (BCI) system. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 11:110–113, 2003.

[26] Fabio Aloise, Pietro Aricó, Francesca Schettini, Angela Riccio, Serenella Salinari, Donatella Mattia, Fabio Babiloni, and Febo Cincotti. A covert attention P300-based brain-computer interface: Geospell. *Ergonomics*, 55:538–51, 2012.

[27] Jing Jin, Brendan Z Allison, Xingyu Wang, and Christa Neuper. A combined brain-computer interface based on P300 potentials and motion-onset visual evoked potentials. *Journal of neuroscience methods*, 205:265–76, 2012.

[28] Çağdaş Ulaş and Müjdat Çetin. Deriving Error Characteristics and Developing a Statistical Model for P300-Based Brain Computer Interfaces, 2011.

[29] Armağan Amcalar and Müjdat Çetin. Design, Implementation and uppercaseEvaluation of a Real-Time P300-based Brain-Computer interface system. In *2010 20th International Conference on Pattern Recognition*, 117-120.

[30] Eric W Sellers and Emanuel Donchin. A P300-based brain-computer interface: initial tests by ALS patients. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 117:538–48, 2006.

[31] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 113:19, 2002.