

**Development of a Wearable and Cost Effective Brain-Computer Interface  
Assistive Device**

BY

DAVIDE MARZORATI  
B.Sc., Politecnico di Milano, Milan, Italy, 2015

THESIS

Submitted as partial fulfillment of the requirements  
for the degree of Master of Science in Bioengineering  
in the Graduate College of the  
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:

Hananeh Esmailbeigi, Chair and Advisor

Yang Dai

Luca Mainardi, Politecnico di Milano

## ACKNOWLEDGMENTS

First, I thank my advisor, professor Hananeh Esmailbeigi from the department of Bioengineering at UIC, as she gave me the exciting chance to work on this project. She has always been present to help me whenever I needed, and from time to time she was even able to anticipate my needs.

I'm deeply grateful to professor Matteo Matteucci and professor Luca Mainardi from Politecnico di Milano. Their suggestions, support and help made possible to complete this work. I would also like to thank Dr. Emanuele de Bernardi from Politecnico di Milano for providing me useful hints and valuable help to set up the genetic algorithm.

I want to extend my thanks to Ameen, Ben and Brandon their help, support and active collaboration.

During my work I've used many software tools, and most of them were freely available. Just to name a few, the OpenBCI GUI to test the OpenBCI board, Processing to develop the BCI software, Arduino to develop the robotic arm firmware, EDFBrowser to visualize the EEG signals, and  $\text{\LaTeX}$  to write this manuscript. I'm grateful to all the authors who decided to make them available at no charge.

Many more people contributed indirectly to this thesis work. I'm very thankful to my whole family. My parents have always supported me: thank you mom and dad for every single thing you did for me, and for giving me the chance to live an amazing academic experience in Chicago. As the youngest brother, I thank the best sisters I could have ever dreamed of: thank you Ila,

## ACKNOWLEDGMENTS (continued)

Franci and Mery. Then, thank you granny Pia, because you have always had me on your mind (and, by the way, “*De danée ga n’huu anca mo!*”). I would also like to thank my uncle Aldo and my aunt Brunella for their support.

Many friends kept me company during the nine months I spent in Chicago. Thank you Ale, Andre, Ele, Jack, Luca and Marco. In particular, thank you Ele for all the joyful times we had in Oak Park, and Andre for all the talks and lunches we had together.

During my university years at Politecnico di Milano, I had the chance to make friends with many people. I would particularly like to thank Andre, Eddy, Fra and Lore for their friendship. The talks we had when I was in Chicago always reminded me how important is your friendship.

Many other people have always been close to me. I’m not going to list all my friends here, both because it would too long, and because the day after submitting this manuscript I would probably realize that I forgot someone. Even if I don’t mention all of my friends, I would like to thank each one of them.

Special and important thanks go to Lisa. Thank you for being by my side every day: you believe in me, you support me, and you give me the strength to get through hard times. I don’t know what I would do without you.

DM

## TABLE OF CONTENTS

<b><u>CHAPTER</u></b>		<b><u>PAGE</u></b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	The Cerebrum . . . . .	3
1.1.1	The EEG Signal . . . . .	6
1.2	BCI . . . . .	8
1.2.1	Definition . . . . .	8
1.2.2	Classification . . . . .	9
1.2.3	Components . . . . .	11
1.2.4	Non - Invasive BCI . . . . .	14
1.3	P300 Event-Related Potential . . . . .	15
1.4	Related Works . . . . .	20
1.4.1	P300 Based Wheelchair-Mounted Robotic Arm Control . . . . .	21
1.4.2	P300 Based BCI Control of a Humanoid Robot . . . . .	23
1.4.3	SSVEP Based BCI to control a Hand Orthosis . . . . .	24
<b>2</b>	<b>MATERIALS AND METHODS</b>	<b>26</b>
2.1	Hardware . . . . .	26
2.1.1	EEG Acquisition System . . . . .	26
2.1.2	Robotic Arm . . . . .	33
2.2	Software . . . . .	38
2.2.1	Visual Stimulation . . . . .	43
2.2.2	EEG Data Retrieval . . . . .	46
2.2.3	Synchronization Protocol . . . . .	47
2.3	Classification . . . . .	49
2.3.1	Logistic Classifier . . . . .	51
2.3.2	Genetic Algorithms . . . . .	53
2.3.2.1	Genetic Algorithm for Automatic Feature Extraction . . . . .	60
2.3.3	Online Classification . . . . .	70
2.3.4	Experimental Protocol . . . . .	74
<b>3</b>	<b>RESULTS</b>	<b>76</b>
3.1	Initial Evaluation Phase . . . . .	76
3.2	Classifier Training Phase . . . . .	78
3.3	Classifier Validation Phase . . . . .	82
3.3.1	Accuracy vs Repetitions . . . . .	83
3.3.2	Accuracy vs Training Set Size . . . . .	85

## TABLE OF CONTENTS (continued)

<b><u>CHAPTER</u></b>		<b><u>PAGE</u></b>
	3.3.3 Inter-Subjects Dependence . . . . .	86
3.4	Online Classification . . . . .	88
	3.4.1 Speller . . . . .	90
	3.4.2 Robotic Arm Cartesian Control . . . . .	93
	3.4.3 Robotic Arm High Level Control . . . . .	95
	3.4.4 Average Performance . . . . .	98
	3.4.5 Comparison between Classification Methods. . . . .	102
<b>4</b>	<b>CONCLUSIONS</b>	<b>104</b>
	4.1 Future Improvements . . . . .	109
	<b>APPENDICES</b>	<b>110</b>
	<b>APPENDIX A</b>	<b>111</b>
	<b>APPENDIX B</b>	<b>112</b>
	<b>CITED LITERATURE</b>	<b>115</b>
	<b>VITA</b>	<b>122</b>

## LIST OF TABLES

<u>TABLE</u>	<u>PAGE</u>
I TECHNICAL SPECIFICATIONS OF THE EMOTIV EPOCH EEG DEVICE AND OF THE OPENBCI BOARD . . . . .	27
II DESCRIPTION OF THE DATA PACKET SENT BY OPENBCI. . . . .	47
III SUBJECTS PARTICIPATING IN THE STUDY. . . . .	74
IV DESCRIPTION OF THE TRAINING SET OF THE TWO SUBJECTS. . .	78
V DESCRIPTION OF THE VALIDATION SET OF THE TWO SUBJECTS. .	83
VI INTER-SUBJECT DEPENDENCE: ACCURACY WAS COMPUTED USING THE MAXIMUM NUMBER OF REPETITIONS OF THE VALIDATION SET. . . . .	87
VII NUMBER OF CHOICES N AND MAXIMUM NUMBER OF FLASHES FOR THE THREE POSSIBLE MODES OF USE. . . . .	88
VIII AVERAGE VALUES OF THE ACCURACY AND ITR FOR <i>PROBABILITY AVERAGING</i> METHOD. . . . .	100
IX AVERAGE VALUES OF THE ACCURACY AND ITR FOR <i>EPOCHS AVERAGING</i> METHOD. . . . .	100
X ONLINE CLASSIFICATION RESULTS IN TERMS OF ACCURACY AND INFORMATION TRANSFER RATE. RESULTS OBTAINED USING <i>PROBABILITY AVERAGING</i> METHOD. . . . .	106
XI DETAILED DESCRIPTION OF THE SYMBOL USED FOR THE STIMULATION IN THE CARTESIAN CONTROL OF THE ROBOTIC ARM.. . .	111

## LIST OF FIGURES

<b>FIGURE</b>	<b>PAGE</b>
1 Broodman areas of the brain. Image taken from: <i>OpenStax Anatomy and Physiology</i> , licensed under CC-BY-4.0. . . . .	4
2 Basic structure of a neuron. Licensed under <i>GNU Free Documentation License 1.2</i> . . . . .	5
3 The 10-20 international standard for electrodes positioning. . . . .	7
4 Basic design of a BCI system. In this diagram, the different parts of a BCI are shown: (1) <i>Signal acquisition</i> , (2) <i>Signal processing</i> (3) <i>Output Device</i> . . .	12
5 <i>Left</i> : EEG signal when a desired stimulus is presented. <i>Right</i> : EEG signal when an undesired stimulus is presented. Time $0$ represents the stimulus onset.	17
6 Grid used by <i>Farwell and Donchin</i> for the first P300 - based BCI. . . . .	18
7 <i>Left</i> : mapping between letters and robotic arm movements. <i>Right</i> : the grid presented to the user with flashing letters. . . . .	22
8 The hand orthosis used for the development of the SSVEP-based BCI. . . . .	25
9 The OpenBCI board used as the EEG acquisition system in the proposed device. . . . .	28
10 The 3D printed Mark III EEG headset that was utilized as a support for the electrodes. The chin strap that we included to add more stability to the headset is visible. . . . .	30
11 The available electrodes locations of the 3D printed EEG headset. The utilized locations are: C3,C4,P7,P3,Pz,P4,P8, Oz. . . . .	32
12 <i>Left</i> : picture of the Lynxmotion AL5B. <i>Right</i> : technical specifications of the AL5B Robotic Arm. . . . .	33
13 The HCSR04 ultrasound sensor. . . . .	34
14 The interface of the software developed for robotic arm control. Buttons for discrete movements, sliders for setting increment values, sliders for setting a specific position can be used to tune the parameters in an optimal way. . . .	36
15 Screenshots of the developed BCI software. . . . .	39
16 <i>Left</i> : speller stimulation grid without symbols. <i>Right</i> : speller stimulation grid with symbols. . . . .	43
17 The two stimulation grids that can be used in the robotic arm control mode.	45
18 The default flashing pattern used in both speller and robotic arm mode. . . .	46
19 Signals recorded simultaneously from the flashing square and column. Photoresistor signals are normalized between 0 and 1. . . . .	48
20 Generic structure of a genetic algorithm. . . . .	54
21 Crossover process in genetic algorithms. . . . .	57
22 Mutation process in genetic algorithms. . . . .	58
23 Structure of a chromosome in the genetic algorithm for automatic feature extraction©[2011] IEEE. . . . .	61

## LIST OF FIGURES (continued)

<u>FIGURE</u>		<u>PAGE</u>
24	Weight functions used in the genetic algorithm. . . . .	62
25	The training setup in the speller mode. . . . .	63
26	Process of tournament selection employed in the genetic algorithm. . . . .	65
27	The software component of the assistive device. . . . .	75
28	Average target and non-target epochs for the two subjects during the initial evaluation of the device. In the average target epochs for the two subjects, it is possible to see a positive peak at 300 ms. . . . .	77
29	Average target and non-target epochs of the training set. . . . .	79
30	<i>Left:</i> Fitness value of each of the 100 chromosomes at each generation is shown in blue. Mean fitness value is shown in red. <i>Right:</i> Box plots of fitness values as a function of genetic algorithm generation. [Data obtained with training set of subject S2]. . . . .	80
31	Time required to complete a run of the genetic algorithm as a function of the size of the training set. This set was performed on a 64 bit laptop. . . . .	81
32	Fitness values as a function of number of features and of genetic algorithm generation. [Data were obtained with subject S2 training set]. . . . .	82
33	Average accuracy as a function of the number of repetitions for the validation set. In the first two rows, individual sessions are considered. In the last row, average accuracy is reported. . . . .	84
34	Average accuracy on the validation set as a function of the training set size. The maximum number of repetitions was used to compute the accuracy. . . . .	86
35	Average target and non-target epochs of the validation set for subject S1 and S2. . . . .	87
36	Protocol used to compute $T_{tot}$ . . . . .	89
37	Time to complete a trial, $T_{tot}$ , as a function of the number of repetitions for three BCI modes of use. . . . .	90
38	Online classification accuracy as a function of the number of repetitions for the speller mode. Accuracy on the validation set is shown with dashed lines. . . . .	91
39	Information transfer rate as a function of the number of repetitions for the speller mode. . . . .	92
40	Online classification accuracy as a function of the number of repetitions for the cartesian control of the robotic arm. . . . .	93
41	Information transfer rate as a function of the number of repetitions for the cartesian control of the robotic arm. . . . .	94
42	Online classification accuracy as a function of the number of repetitions for the high level control of the robotic arm. . . . .	95
43	High level control of the robotic arm: information transfer rate as a function of the number of repetitions. . . . .	96
44	Average values of the accuracy and ITR amongst the two subjects as a function of the number of repetitions for the three modes of use. . . . .	101

## LIST OF FIGURES (continued)

<u>FIGURE</u>		<u>PAGE</u>
45	<i>Left)</i> Overall average accuracy among the two subjects for the three modes of use as a function of the number of repetitions. <i>Right)</i> Ratio between classification accuracy of probability averaging and epochs averaging method as a function of the number of repetitions. . . . .	102
46	Comparison between the present study and <i>Thusulasidas et al.</i> , <i>Cecotti et al.</i> , <i>Dal Seno</i> for online classification accuracy as a function of the number of repetitions in the speller mode. . . . .	108

## LIST OF ABBREVIATIONS

BCI	Brain Computer Interface
EEG	Electroencephalography
ERP	Event Related Potential
VEP	Visual Evoked Potentials
DoF	Degrees of Freedom
GUI	Graphical User Interface
GA	Genetic Algorithm
ITR	Information Transfer Rate

## SUMMARY

Annually, across the world, 250,000 to 500,000 individuals suffer an injury at the level of the spinal cord [1]. These injuries could lead to a variety of debilitating conditions, such as tetraplegia, which is the paralysis of all the four limbs. Individuals living with these extreme conditions are no longer able to voluntarily control muscles movement, and are therefore unable to communicate with others, or use external devices. For these individuals, assistive technologies are utilized to allow, or ease, communication and mobility. One example of such assistive technologies are Brain-Computer Interfaces. In this work, we present the development of a wearable and cost-effective BCI assistive device. This device serves as a proof of concept that cost-effective EEG acquisition systems along with robust classification techniques can be used in the field of assistive technologies.

The proposed assistive device functions based on the P300 response of the human brain. The P300 response is an innate response of the brain, therefore no training for the subject is required to use a P300 based BCI. In the proposed device, electroencephalography (EEG) signal is acquired using the OpenBCI Cyton board [2]. The user needs to wear a 3D printed headset which houses the electrodes. Flashing symbols on a grid are employed as stimuli to elicit a P300 response. Online detection of the P300 response is performed using logistic regression, and the genetic algorithm developed by *Dal Seno et al.* [3] is used to perform automatic feature extraction for P300 detection.

## SUMMARY (continued)

The device allows the disabled individual to type words on a computer screen and to control a 4 Degrees of Freedom robotic arm. Two modes of operation could be used to control the robotic arm. One mode (*cartesian* control) consists of controlling the robotic arm with discrete movements. The second control mode (*high level* control) consists of sending high level commands to the robotic arm, that would move autonomously according to the selected action. Therefore, the presented assistive device augments communication capabilities and allows for the control of external devices.

The presented assistive device was validated on two healthy male subjects (20 and 23 years old). Offline and online experiments were performed for all the modes of use. Amongst the two subjects, average maximum information transfer rate for the speller mode is 12.56 *bits/min*, for the cartesian control of the robotic arm is 9.95 *bits/min*, and for the high level control of the robotic arm is 2.93 *bits/min*. Results show that our wearable and cost-effective device is comparable to previously published studies utilizing clinical grade EEG acquisition systems [4; 5; 6].

In this work, a wearable and cost-effective BCI device is presented. Future improvements are necessary to transition from laboratory setting to in-home use, allowing many disabled individuals to improve their quality of life.

## CHAPTER 1

### INTRODUCTION

Annually, across the world 250,000 to 500,000 people suffer an injury at the level of the spinal cord [1]. These injuries can lead to a variety of debilitating conditions, such as tetraplegia, which is a paralysis of all the four limbs. Other than spinal cord injury, there are several disorders which may affect the neural paths normally used by the brain to communicate with the peripheral muscles of the body, and some examples of these disorders are multiple sclerosis, stroke at the level of the brainstem and brain injuries in general [7]. Individuals who suffer from these conditions are affected with various degrees. Those with extreme condition are no longer able to voluntarily control muscles, therefore being unable to communicate with others and express their feeling, or use any external device. For these people, assistive technology is required to allow, or ease, mobility, communication and domestic activities.

In order to restore the original functions affected by these conditions, different options can be exploited:

- the first option would be to increase the functioning of the unaffected pathways. If there are muscles that are still under voluntary control, it is possible to use them in order to substitute the injured muscles. *Damper et al.* presented a system for the non-vocal disabled, allowing them to use their hands to answer questions [8]. *Kubota et al.* presented an ocular movement detector that disabled patients can use to communicate [9];

- the second option would be to detour around the breaks in the damaged neural pathways. Individuals who are suffering from spinal cord injury can use the electromyographic signal generated by muscles, which are controlled by nerves situated above the spinal lesion, to electrically stimulate the injured muscles, which are controlled by nerves below the lesion. *Kobetic et al.* developed an hybrid system able conjugate an exoskeletal bracing together with a device for functional electrical stimulation [10];
- the last option would be to provide to the brain a new channel for communication and control, not based on muscular pathways. Brain monitoring techniques (the electroencephalography signal (EEG), optical imaging techniques, and more traditional imaging techniques such as positron emission tomography (PET) and functional Magnetic Resonance Imaging (fMRI) can serve as a way to analyze the brain activity and detect the user's intent. With these techniques, a Brain-Computer Interface (BCI) can be created. With the created BCI, the user would then be able to transmit messages or other commands to the external world, even if neural pathways are damaged. *Chaudhary et al.* developed a BCI for advanced ALS subjects [26]. In this study, the involved subjects, that live without any available mean of communication, managed to learn how to answer some questions, requiring a "yes" or "no" by means of measurements of frontocentral oxygenation changes. These signals were obtained with functional near-infrared spectroscopy. Among all the possible methods to monitor the brain activity, EEG is the one that has the shortest time constant and that requires simple and inexpensive acquisition devices.

So, this type of brain activity monitoring is the only one who offers the possibility, for the locked-in subject, to use the brain signal to communicate and control devices.

Among all the possibilities EEG based BCI is the best choice. They offer to the impaired subject the possibility to communicate with the external world in a fast and non-invasive way. *Luo et al.* developed a smart house system with an EEG based BCI [11] . This device makes it possible to open or close curtains, turn light switches on and off and control the air-conditioning by simply focusing on images flashing on the screen.

The goal of our study was to develop a wearable and cost-effective BCI system, with the idea of allowing the end-user to spell words on a screen and to control a robotic arm. Building an affordable but, at the same time, reliable system means that it could be used by the majority of the disabled people. Today, thanks to the advances in the technology, EEG equipment is not as expensive as it was years ago, so it is possible to have good quality recording without having to rely on high - level and expensive technology. Tetraplegic subjects have no absolute control of voluntary muscles, so there is no residual movement of the upper limbs. For this reason, providing them a way of interaction with the external environment and with people can improve their quality of life.

### **1.1 The Cerebrum**

The encephalon, that is commonly referred to as the brain, is divided in two main parts: the telencephalon and the diencephalon [57]. These parts have different functions: the telencephalon is responsible of high-level functions, as thinking, planning and storing, while the diencephalon

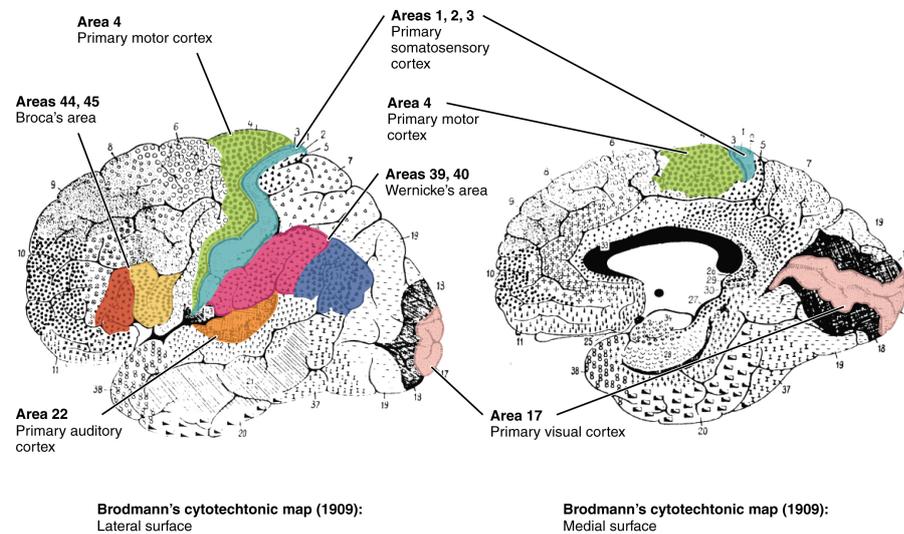


Figure 1: Brodmann areas of the brain. Image taken from: *OpenStax Anatomy and Physiology*, licensed under CC-BY-4.0.

works on low-level functions, as maintaining the homeostasis and controlling the circadian rhythm.

The cerebral cortex represents the outside layer of the telencephalon. In humans, the cerebral cortex is folded, so that it provides a very high amount of surface area in the confined volume of the skull [57]. The cerebral cortex is separated into two cortices by the longitudinal fissure, that divides the brain into the right and left hemispheres. It is possible to identify, in the cerebral cortex, areas in which the neurons share the same function [29]: these are generally referred to as Brodmann areas, and the current number of identified areas is 52. Examples of Brodmann areas, shown in Figure 1, are the motor cortex, which is responsible of all the voluntary movements in the body, the sensory cortex, that receives and processes the signals

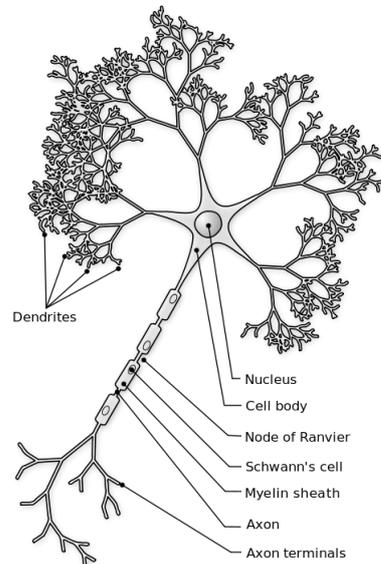


Figure 2: Basic structure of a neuron. Licensed under *GNU Free Documentation License 1.2*.

coming from the sensory neurons of the body, and the visual cortex, which receives the signals of the receptors of the retina.

The cerebral cortex is mainly made up of gray matter, which is constituted by neuron bodies. The underlying white matter consists of the axons of the neurons. The neurons, characterized by the basic structure shown in Figure 2, are connected through long fibers, the axons. The origin of the axon is in the cell body, and they represent the output terminals of the neuron. The dendrites, instead, are those parts of the neurons that receive the input from other neurons. A synapse is a junction present between the end tip of a neuron axon and the dendrites of another neuron. The synapse is used by the neurons to communicate [57]. The neuronal signal coming from the presynaptic cell crosses the junction thanks to the release of chemical substances,

called the neurotransmitters. The neurotransmitters, when bonding with the receptors present on the membrane of the postsynaptic cell, cause a variation of the membrane potential of the postsynaptic cell. If the membrane potential goes above a certain threshold, an action potential is generated, and this potential will be transmitted along the axon and will reach other neurons to transmit the neuronal activity. The threshold for the generation of the action potential depends on the considered type of neuron [57].

Among all the possible methods used to monitor the activity of the cerebrum, two main groups can be identified: methods which measure directly the activity of the brain (for example, the EEG), and methods which estimate the neuronal activity in an indirect way (with a measure of the blood flow, as an example). When using the direct methods, the measurement of the electrical activity of the brain is a nonlinear, spatial and temporal combination of the action potentials of all the present neurons.

### **1.1.1 The EEG Signal**

The electroencephalographic signal (EEG) was discovered in 1929 by Hans Berger [30]. Since then, it has been used for multiple purposes. Today main applications are: diagnoses of diseases (epilepsy, for example); development of BCI; analysis of the patient status during anesthesia.

In order to record the EEG signal in a noninvasive way, it is possible to use electrodes placed outside the brain and directly on the scalp. These electrodes are able to detect the variation of the electric potential, generated by the presence of action potentials and by the basal activity of the neurons.

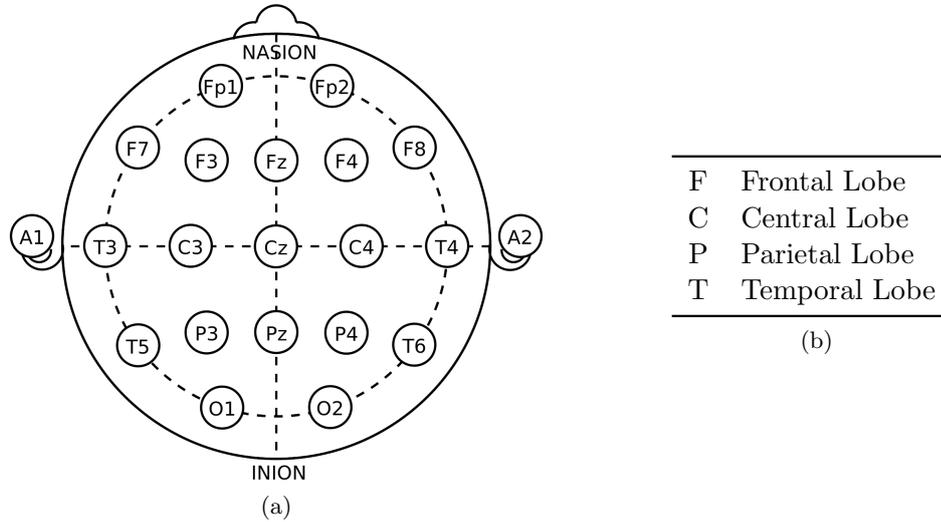


Figure 3: The 10-20 international standard for electrodes positioning [31].

Electrodes are typically made of materials such as gold or silver, so that their impedance is low (ideally, below 10 kOhm). Regarding the positions of the electrodes, a standard was defined: the 10-20 system [31]. The electrodes locations defined in the 10 - 20 system are shown in Figure 3a. In this standard, four fixed points are defined. The first point is the nasion, which is situated between the forehead and the nose. The second point is the inion, which can be located as the lowest point of the skull from the back of the head, and situated along the antero-posterior plane as the nasion. The last two points are the preauricular points, situated in the anterior part of each ear. The number '10' and '20' were chosen because the distance between each electrode is equal to the 10 or 20% of the nasion-inion or right preaurical point - left preaurical point distances on the skull.

Each electrode position is identified with a letter and a number. The meaning of each letter is shown in Table 3b (note that the letter C, which stands for central lobe, is used only for identification purposes, since there is no central lobe in the brain). To identify electrodes of the right hemisphere, even numbers are used. Left hemisphere electrodes are marked with odd numbers.

With this standard, it is possible to identify a total number of 21 electrodes positions, as shown in Figure 3a. Additional positions can be added by using the 10 - 10 standard, in which the distance between each electrode is always equal to the 10% of the total back-front or left-right distances of the skull.

The aim of EEG is to record the variation of the electric potential on the scalp. In order to do this, it is necessary to have a reference level for these measurements. There are several possible ways to define a reference level. One way is to choose an electrode that acts as a fixed reference for all the other electrodes, and that is typically positioned at the level of the mastoid or on the earlobe: this technique is called *common reference*. Another way is to compute the average value of all the signal recorded from each channel and then, subtract this value from the signal recorded at each channel (*average reference*). The reference can also be defined for each couple of electrodes, so that each channel has its own reference (*bipolar configuration*).

## 1.2 BCI

### 1.2.1 Definition

*Wolpaw et al.* define BCI as: “A BCI is a communication system in which messages or commands that an individual sends to the external world do not pass through the brain’s

normal output pathways of peripheral nerves and muscles” [7]. An individual who makes use of a BCI is able to maintain an interaction with the external world even if his/her neural pathways are severely injured.

### 1.2.2 Classification

According to *Wolpaw et al.* [7] BCI can be classified as follows:

- **Dependent and independent BCI:** a dependent BCI doesn't use the normal output pathways of the brain to carry the command or the message. At the same time, to generate the required brain activity it is necessary to have residual activity in these pathways. To understand what a dependent BCI is, consider a BCI based on Visual Evoked Potentials (VEP). To elicit a response from the subject, he is presented with a matrix of flashing letters. Then, he has to gaze at a specific letter in order to have a VEP that is greater when that letter flashes in comparison to the others that are recorded when other letters flash. So, the output channel of the brain is the electric signal recorded with EEG, but its generation still depends on the gaze direction, which is dependent on the eye muscles. For the mentioned reasons, a dependent BCI is another method to detect messages that are present in output paths of the brain (in the described example, the direction of the gaze isn't monitored by observing the position of the eye, but rather by recording EEG signal). Instead, an independent BCI is completely independent, as the name suggests, on the normal output pathways of the cerebrum, and for this reason no activity in those paths is necessary for the generation of the brain signal of interest. To better understand how an independent BCI works, consider a BCI that is based on the P300 response.

The P300 response is an event related potential, that is elicited when stimuli, which are significant from the subject's point of view, are presented to him/her. and that presents the user with flashing letters. When the letter the user is focusing on flashes, a P300 will be elicited. No P300 response will be present when other letters flash. The EEG signal that is measured is not dependent on the gaze direction of the eyes, but rather on the user's intent. As it is possible to understand from the provided examples, independent BCIs are of much more interest, because they provide completely new channels to the brain.

- **BCI employing invasive or non-invasive techniques for brain monitoring:** An example of a non-invasive technique is the EEG recording, that can be recorded from the scalp without requiring surgery or invasive procedures. Invasive techniques require the placement of matrices of electrodes directly on the brain, and for this reason surgeries are necessary.

There are both advantages and disadvantages for the two methods. The most important advantage of non-invasive BCI is that the end-user is not required to undergo surgeries.

The disadvantage of non-invasive BCIs, which represents an advantage of invasive BCIs, is the quality of the electric signal recorded. *Ball et al.* compared invasive and non-invasive EEG measurements [12]. The main goal of the study was to determine if blink related artifacts, which are always present in non-invasive recordings, were also present in invasive recordings. As expected, eye blinks caused artifacts in non-invasive recordings and, unexpectedly, these artifacts were also present in the invasive recordings, particularly

in the prefrontal region. After this analysis, by computing the ratio between the amplitude of the artifacts and the amplitude of the background brain activity, it was possible to determine that the quality of the invasive EEG signal was from 20 to 100 times better than the non-invasive EEG recorded simultaneously.

Reduction in noise and eye-blink artifacts would allow to improve the quality of the EEG signal and to obtain a better implementation of a BCI system, since classification of EEG signal can be performed easily.

Despite of the advantages in using invasive techniques, the majority of researchers consider the non-invasive techniques as more appropriate than the invasive ones, since they have an advantage that overcomes all the potential advantages of the invasive techniques: they do not require a complex and dangerous surgery for the subject. [7].

### **1.2.3 Components**

In Figure 4 several components of a generic BCI system are shown. A BCI can be described as a generic control system. It has an input and components that process and transform the input signal into the output signal. In the represented system, the output signal consists of commands sent to a robotic arm or by letters shown on a screen.

#### **Signal acquisition**

In Section 1.2.2, we discussed the difference between invasive and non-invasive recordings of the brain activity. What we would like to highlight here is that it is of utmost importance that the quality of the recording is the best possible. The introduction of noise and artifacts could

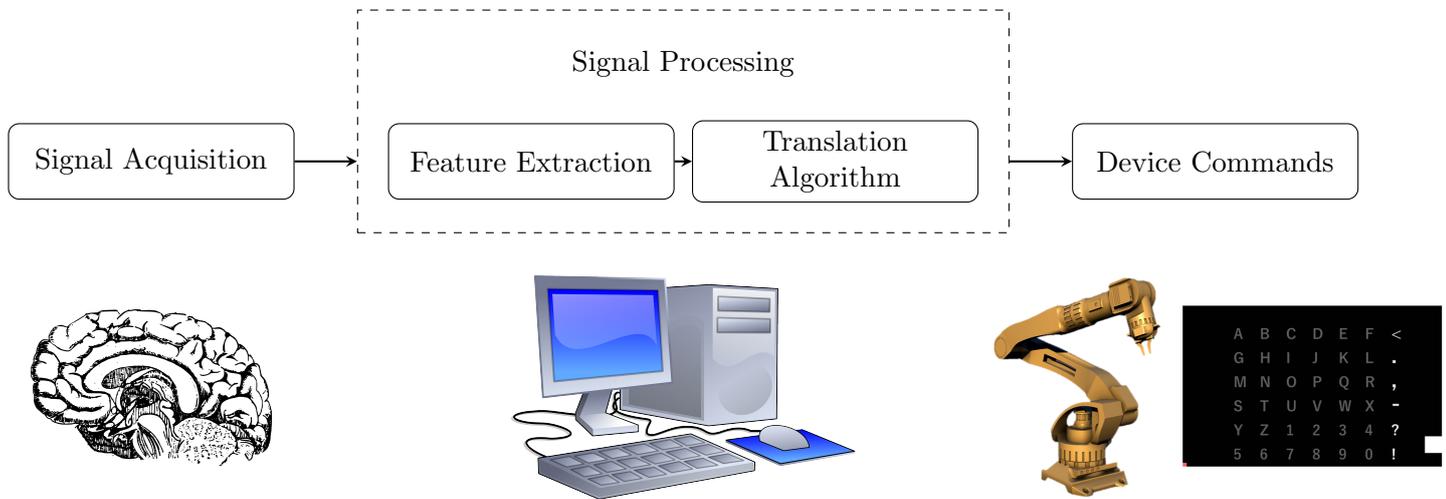


Figure 4: Basic design of a BCI system. In this diagram, the different parts of a BCI are shown: (1) *Signal acquisition*, (2) *Signal processing* (3) *Output Device*.

affect and greatly reduce the overall performance of the BCI. For non-invasive EEG recording, there are some entities that need to be defined: the number of channels to be used, the position of the electrodes on the scalp, the EEG acquisition device and so on. Not only these entities affect the ability of the BCI to extract features from the EEG signal, but they also have an effect on the portability of the system. If a bulky EEG system is chosen for the recording, then it would be difficult for the subject to use the BCI outside of a hospital/laboratory.

### Signal Processing

As visible in Figure 4, the signal processing block is composed of two methods: feature extraction and translation algorithm. In the feature extraction method, the signals are subjected to a feature extraction procedure. Example of feature extraction procedures are filtering or

spectral analysis. Hopefully, this analysis determines the features representing what the user wants to communicate. Features that are commonly extracted are related to specific brain waves or rhythms that mirror events currently happening in the brain.

The translation algorithm method converts the previously extracted features into device commands. Effective translation algorithms must adapt to the user's signals characteristics [7]. Three different levels of adaptation can be identified. At the first BCI access of each user, the algorithm has to adapt to him/her by analyzing the features of the signal. EEG signals typically display variability on both short and long term. A translation algorithm that only has the first level of adaption would be completely ineffective on the short and long term usage of the BCI. An additional level of adaptation, with periodic modifications aimed at reducing the impact of these variations, is required. The third and last level of adaptation is the most challenging one. When a signal feature that has always been a reflection of the brain function starts to be an output signal encoding the user's intent, it is then subjected to the adaptive capabilities of the brain. So, the outcome of the BCI will also affect the input signal of the BCI, and an effective translation algorithm should take this into consideration.

### **The Output Device**

For the majority of BCI systems, the output device is a screen and the real output is the result of the selection of symbols presented on the screen. These symbols can be letters, icons, or any other type of stimuli. According to the choice that is selected, a letter is displayed on the screen, or the action of an external device (wheelchair, ...), correspondent to the selected icon, is performed.

#### 1.2.4 Non - Invasive BCI

Any of the following techniques to monitor the cerebral activity can serve to build a non-invasive BCI:

1. **Magneto - Encephalography (MEG)**: MEG is a technique that makes use of magnetic fields to monitor the brain activity. Intracellular currents occur in a natural way in the brain. Magnetic fields are produced by these currents. In order to detect these magnetic fields, sensitive magnetometers have to be used [32]. With this technique, areas that are active during cerebral processes can be identified. Problems related to MEG are sensitivity to external sources of noise (other magnetic fields) and expensive, cumbersome and unportable equipment;
2. **Functional Magnetic Resonance Imaging (fMRI)**: fMRI is an imaging technique that makes use of the magnetic resonance to evaluate the status of the brain. This imaging technique is complementary to the morphologic imaging, which is focused on analyzing the morphology of the organ. The signal that is measured with fMRI-based BCI is the BOLD signal (Blood Oxygenation Level Dependent *signal*) [33]. This measure is an indirect measure of the activity level of a cerebral area. A greater amount of oxygen consumption corresponds to a higher level cerebral activity. The problems associated with fMRI-based BCI are the same as the one described for the MEG. In addition to these, a delay of  $\sim 3-6$  seconds, between the cerebral activity and the recorded signal, could be present;
3. **Slow Cortical Potentials (SCP)**: SCPs are low-frequency (DC-2Hz) voltage changes recorded at the level of the scalp. These voltage changes are associated to cognitive and

sensorimotor events. Negative SCPs are related to functions that cause an activation at the level of the cortex. Positive SCPs are associated with a reduction in the cortical activation [7]. SCPs - based BCI depend on the ability of people to learn how to control these voltage changes. *Birbaumer et al.* developed a BCI in which subjects could use SCPs for movement control (upwards, downwards) of a cursor displayed on a screen [34].

4. **Mu Rhythms:** in awake people, motor cortex displays an activity in the frequency range of 8-12 Hz. This activity is called *mu rhythm*, and is comprised of a variety of different rhythms. Each rhythm can be distinguished according to the location, frequency and relationship to a contemporaneous motor output. The prior preparation to the movement and the movement itself correspond to a decrease in the mu rhythm in a contralateral way (event-related desynchronization). An increase of the rhythm occurs after movement and during relaxation. For the mentioned reasons, a pattern visible in the mu rhythms can be associated to a specific movement. The identification of this pattern can then be used to control external devices. No real movement is necessary, since mu rhythms are also present with motor-imagery [15].

### **1.3 P300 Event-Related Potential**

Event Related Potentials (ERPs) are stereotyped electrophysiological responses that happen after a sensory stimulation. ERPs are visible in the EEG signal and can be distinguished from the background electric activity of the brain [57]. Evoked potentials can be classified in two main categories. Those dependent on the nature of the stimulus are defined as exogenous

potentials. Those dependent on the meaning of the stimulus, rather than on its nature, are defined as endogenous potentials.

One of the most known ERP is the P300 response. A P300 response is elicited when an auditory, visual, or somatosensory stimulus, which is significant for the subject, is presented to him/her infrequently and interlarded with insignificant stimuli. The P300 response is visible in the EEG trace as a positive peak at about 300 ms after the stimulus onset [16]. The reason why this ERP is called P300 is that it is a positive deflection (**P**) that happens **300ms** after the stimulus onset. The P300 response is commonly referred to as the "oddball" response. As shown in Figure 5, it can be noticed that P300 response is predominant in responses elicited by the stimulus representing the user's intent (so, when the stimulus is the *target* one). Instead, no P300 response is present when the provided stimulus does not reflect the user's intent (so, when the stimulus is not the *target* one).

*Duncan et al.* showed that the amplitude of the peak of the P300 response is inversely proportional to the frequency of the target stimulus [35]. Therefore, the higher the frequency of the presentation of the target stimulus, the lower is the amplitude of the wave. This makes it clear that the frequency of the stimulation affects the resulting generated response.

*Ruchkin et al.* showed that the latency of the P300 response is related to the time necessary for the subject to fully recognize and understand the presented stimulus [36]. So, the latency depends on the complexity of the stimulus and on the amount of information carried by the stimulus.

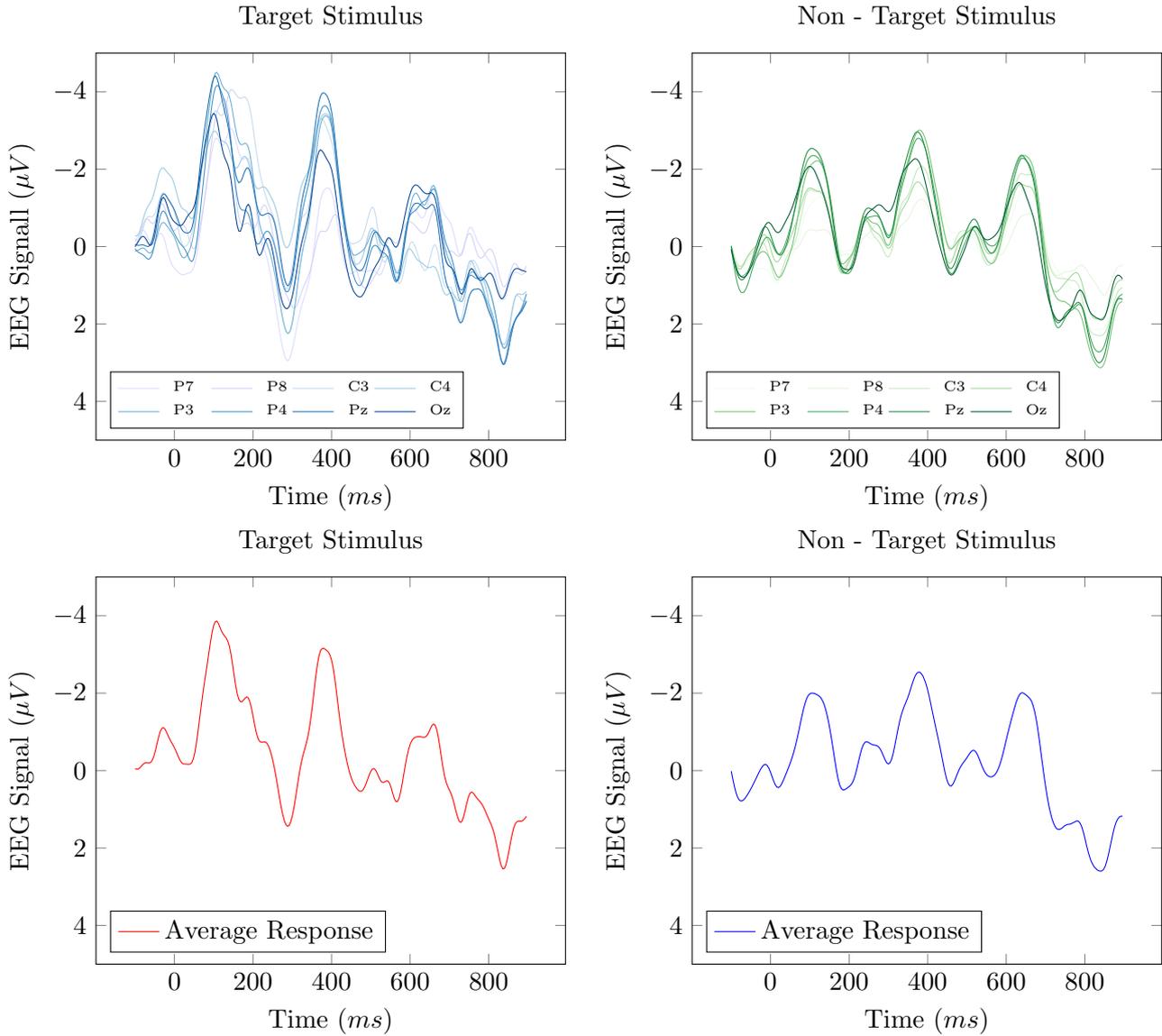


Figure 5: *Left*: EEG signal when a desired stimulus is presented. *Right*: EEG signal when an undesired stimulus is presented. Time 0 represents the stimulus onset.

Choose one letter or command

A	G	M	S	Y	*
B	H	N	T	Z	*
C	I	O	U	*	TALK
D	J	P	V	FLN	SPAC
E	K	Q	W	*	BKSP
F	L	R	X	SPL	QUIT

Figure 6: Grid used by *Farwell and Donchin* for the first P300 - based BCI [17].

The P300 response has a great potential in the BCI field. *Farwell and Donchin* [17] described the development of the first P300 based BCI. In this BCI, the user is presented with a 6 by 6 grid, with letters and one-word commands. A 100 ms flash of a row or a column happens every 125 ms, and in each repetition, composed of 12 flashes, each letter/number/command flashes twice. For the user, it is possible to make a selection by focusing on the desired input. Counting how many times the symbol flashes helps the user to stay focused on the task. In order to identify the presence of a P300 response in the EEG trace, the authors used stepwise discriminant analysis as a classification technique. This procedure yields to a score measuring the distance between each epoch, composed of 1200 ms of EEG signal extracted in a symmetric way around the time at which the stimulus is provided, and the average of a group of epochs known to include a P300. With the value of this measure, it is possible to determine if that epoch contains a P300, and so if that stimulus is the one on which the user was focusing his/her intent.

The greatest advantage of employing the P300 response in BCI field is that subject training is not required. This is due to the fact that the P300 response is an innate response of the human brain. Slow cortical potentials and mu rhythms instead, require training of the subject to be successfully used in a BCI.

The P300 response, as other event related potentials, changes over time. Periodic adaptation of the translation algorithm to the current characteristic of the wave is required in order to have a constant good performance. When employing the P300 response in BCI, the communication rate is low. In fact, the response of the subject is time-locked to the presentation of the stimulus: for this reason, it is necessary to define a value of inter-stimulus interval (ISI) that allows to increase the transfer rate of information but, at the same time, allows to obtain a good performance in the identification of the P300 response [37].

One of the goals of our wearable BCI device is to control a robotic arm. In order to perform this control through a BCI, the choice of the P300 response as the electrophysiological signal to be used to control the arm was the best one. Using a stimulation grid similar to the one employed by *Farwell and Donchin* in [17] and presenting a predetermined set of possible movements the robotic arm can perform, the user would be able to select the movement he would like the robotic arm to perform. We also wanted to develop a system that doesn't require any training of the subject. The P300 response is an innate response of the brain, and so no training of the subject is required.

Our goal was to develop a wearable and cost effective BCI assistive device. To do this, we analyzed possible choices of cost-effective EEG acquisition devices, to reduce the overall cost

of the device. We designed a BCI software, which records and processes the EEG signal while presenting a stimulation grid to the user. Other than controlling the robotic arm, with our software it is possible for the user to type characters and words, using a grid similar to the one used by *Farwell and Donchin* in [17].

Our aim is that this work will be useful in helping the definitive transition of BCI from hospital or laboratories to patients' houses. Using wearable and affordable EEG equipment, a small robotic arm and a computer screen, the disabled could make use of a reliable method for communicating and interacting with other people.

#### 1.4 Related Works

Several BCI based assistive devices were developed. As we mentioned earlier, *Farwell and Donchin* were the first group who presented a P300 based BCI to type words [17]. Other examples of BCI spellers can be found in *Donchin et al.* and *Krusienski et al.* [61; 62].

Assistive devices for the control of electric wheelchairs were developed. *Iturrate et al* and *Rebsamen et al.* both presented P300 based BCI systems that could be used to control electric wheelchairs [18; 19].

*Congedo et al.* presented a prototype of a P300-based video game working, using the OpenVIBE platform [20; 63]. *Finke et al.* were able to develop a P300 video-game obtaining a classification accuracy of 65% on single trials [21].

In this section, we review three studies related to BCI control of external devices. In two of them, P300 is employed. In one of them, steady state visual evoked potentials are used.

#### 1.4.1 P300 Based Wheelchair-Mounted Robotic Arm Control

*Palankar et al.* developed a system they developed to control a 9 DoF wheelchair mounted robotic arm using a P300 based BCI [22]. The robotic arm used in this system was designed to fulfill the specifications necessary to perform daily tasks. They used a relatively expensive and powerful robotic arm. The most widely used robotic arms, similar to this one, such as the *Jaco Kinova* [40], typically cost around \$ 30000 to \$ 40000 [23] . Using a more cost effective robotic arm would allow more people to use such a system at home too.

In this study, the control of the robotic arm was achieved with a P300 based BCI. The authors used the BCI2000 software [24] to stimulate the user, record and process the EEG signal. In this case, the application presented to the user a grid, very similar to the one used by *Farwell and Donchin* [17], showing letters from A to 0. The stimuli grid presented to the user is shown in Figure 7. The cells are arranged in a 5 x 3 matrix. A 75 ms intensification of a column or a row happens every 125 ms. So, each sequence of flashes, equal to 8 intensifications, lasted for a total time of 1 second. Each letter is mapped to a movement of the robotic arm. However, the subject is not presented with movements of the robotic arm directly, but with letters that are associated to these movements. The mapping between letters and movements is shown in Figure 7.

Other than the head cap used for EEG recording, no information about the EEG acquisition system is provided in the paper.

The authors report an important issue related to safety. Since the complete process of scanning the matrix and detecting the selected stimulus takes about 15 seconds, a delay in the

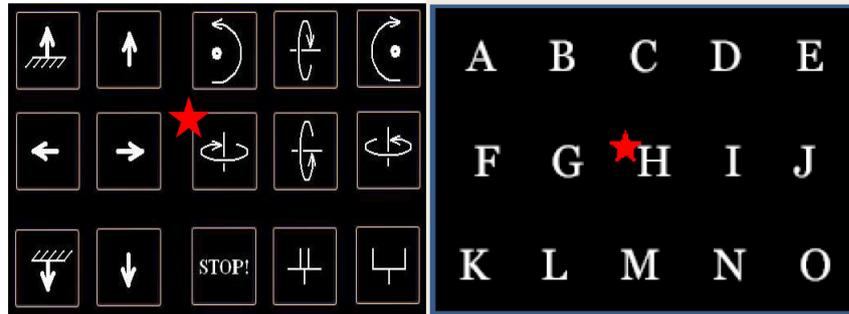


Figure 7: *Left*: mapping between letters and robotic arm movements. *Right*: The grid presented to the user with flashing letters [22].

©[2011] IEEE.

response of the robotic arm is introduced. Should a dangerous situation arise, the robotic arm wouldn't be able to respond in time, and a possible crash could take place.

After summarizing this study, it is possible to identify three main improvements that need to be done in order to build a cost effective P300 - based BCI system for the control of a robotic arm. It is necessary to present to the user stimuli visually representing movements of the robotic arm; then, a cost effective robotic arm should be used; last, safety issues have to be solved. It would be relatively difficult for locked-in individuals to use such system in the daily life, because the user would have to always remember the mapping between letters and movements of the robotic arm.

#### 1.4.2 P300 Based BCI Control of a Humanoid Robot

*Bell et al.* developed a P300 BCI to control a small humanoid robot [25]. This robot was programmed in a way so that it could be controlled with high-level commands. No low level commands had to be used to control the robot.

To develop the system, the authors used a dynamic image-based BCI. The user is presented with flashing images, recorded from cameras mounted on the robot, and the P300 response is utilized to determine the image that the user is focusing on. Thus, enabling the robot to pick up the correspondent object.

EEG signal was recorded using a BiosemiActiveTwo system, which is an acquisition system designed for electro physiology research [67]. This acquisition system is both expensive (\$16000) for the 16 channels version): and unportable too.

During a generic selection, the border of each image is intensified in a random sequence; the flashing happens every 250 ms, and each intensifications lasts for 125 ms. The subject has to focus attention on the image of interest. After 10 flashes per image, a classification on EEG is performed. This classification is used to determine, according to the selected image, where the humanoid robot should direct and which object it should pick up. Average classification accuracy was 98.4% amongst subjects.

The main outcome of this study is that non-invasive BCI can be used to control complex devices, such as humanoid robots. The authors state that this system could be potentially used in the field of helper robots for disabled patients. These robots could move in a home environment, performing actions that the paralyzed individuals are no longer able to do.

The authors also suggest that other types of EEG responses, such as SSVEPs and mu rhythms, could be used for command selection and for the control of devices with better results than P300 response. However, these approaches require more training data and show wider variation across subjects, which are not true for the P300 response.

### 1.4.3 SSVEP Based BCI to control a Hand Orthosis

*Ortner et al.* developed a system to control an orthosis using Steady State Visual Evoked Potentials [27]. These potentials are natural responses to visual stimulations occurring at a fixed frequency  $f_0$ . The eye, when excited with a visual stimulus at frequency  $f_0$  generates an electrical activity in the brain at the same frequency or at multiples of the stimulation frequency.

SSVEP signals can be used in the BCI field for many tasks, ranging from games to the control of hand orthosis [28]. In general, the use of SSVEP in the BCI field requires a lower amount of training than other types of potentials. In fact, the subject is required to only gaze at a light source, without performing any complex task. The drawback of SSVEP control is that the user is required to maintain and focus attention on the blinking lights, and this may cause fatigue.

The hand orthosis used in this study shown in Figure 8. For the development of this system, two LEDs flashing at 8 and 13 Hz were positioned on the orthosis. If the subject gazed at the LED mounted on the left part of the orthosis, the hand would open gradually. Gazing at the LED mounted on the right part of the orthosis caused the hand to close gradually.

To record the EEG signal, the authors used the g.bBSamp EEG amplifier [41] with only one bipolar channel placed in a position close to O1 and the ground electrode placed on Fz. EEG

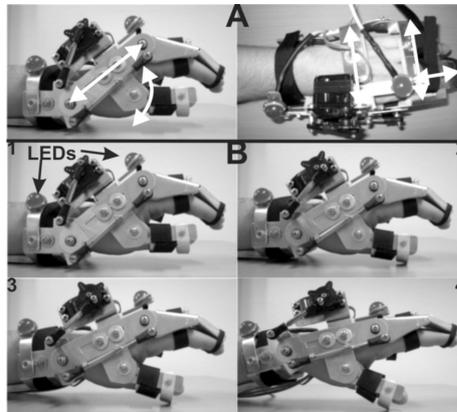


Figure 8: The hand orthosis used for the development of the SSVEP-based BCI [27].  
 ©[2011] IEEE.

signal was sampled at 256 Hz. This acquisition system is very powerful and used in hospital and medical clinics, but it is very expensive.

The results that the authors obtained in these studies were good, with a positive predicted value of  $79\% \pm 21\%$ . The advantage of this type of control is that the end-users can send information whenever they want to, without waiting for a specific stimulus to be presented, as it happens with synchronous BCIs. One of the problem that were highlighted in this study is that the number of false positive detection (i.e. the number of times an SSVEP was identified even if the subject was not focusing on any LED) was high. According to the authors, this could be due to the fact that, even without focusing on the light sources, the subjects still had the light sources in their visual fields.

## CHAPTER 2

### MATERIALS AND METHODS

Three main goals of the proposed wearable and cost-effective BCI assistive device can be identified:

1. stimulate the user with an oddball paradigm, presenting to him/her a grid with letters and numbers (BCI speller), movements or actions of the robotic arm (BCI control of a robotic arm);
2. classify the recorded EEG signal and determine the presence of P300 responses, necessary to detect the stimulus on which the subject was focusing
3. display the detected letter/number on the screen, or make the robotic arm perform the selected movement or action.

In this chapter, we will describe in detail the full development of the BCI device. We will discuss the hardware components, the software and the classification method employed to determine the presence of a P300 response in the EEG signal.

#### **2.1 Hardware**

The hardware consists of two modules: the EEG acquisition system and the robotic arm.

##### **2.1.1 EEG Acquisition System**

The goal of our team was to build a wearable and cost-effective BCI assistive device. For this reason, we compared two cost effective EEG recording devices available on the market.

TABLE I: TECHNICAL SPECIFICATIONS OF THE EMOTIV EPOCH EEG DEVICE [39] AND OF THE OPENBCI BOARD [2].

(a) Emotiv Epoch		(b) OpenBCI	
Property	Value	Property	Value
Channels	Up to 14	Channels	Up to 16
Sampling rate	256 Hz	Sampling rate	250 Hz
Resolution(bit)	14	Resolution(bit)	24
Filtering	Built - in [0.2 - 43Hz]	Filtering	None

The two devices we compared are: Emotiv Epoch [39] and OpenBCI [2]. Emotiv Epoch is a 14 channels wireless (Bluetooth communication) EEG acquisition device. The electrodes are saline based wet sensors. The technical specifications of Emotiv Epoch are described in Table Ia. The market price of the device is of 800\$. OpenBCI is an open-source electronic board with a variable number of channels (up to 16). OpenBCI is compatible with any electrodes. The technical specifications of OpenBCI are described in Table Ib. The price of the 8 channel board is of 499\$. Recently, a new version of the OpenBCI board was released: the Ganglion Board. The Ganglion board has 4 channels, and the market price is of 199\$. The Ganglion board features Bluetooth 4.0 communication protocol.

The Emotiv Epoch is a commercial product that is sold together with a proprietary software. With this proprietary software, there is no easy way to access the raw EEG data. Electrodes in Emotiv Epoch are embedded in a pre-assembled headset. So, using this device means being completely locked to the electrode locations of the pre-assembled headset. These locations are, according to the 10-20 system notation, the following ones: AF3, F7, F3, FC5, T7, P7, O1,

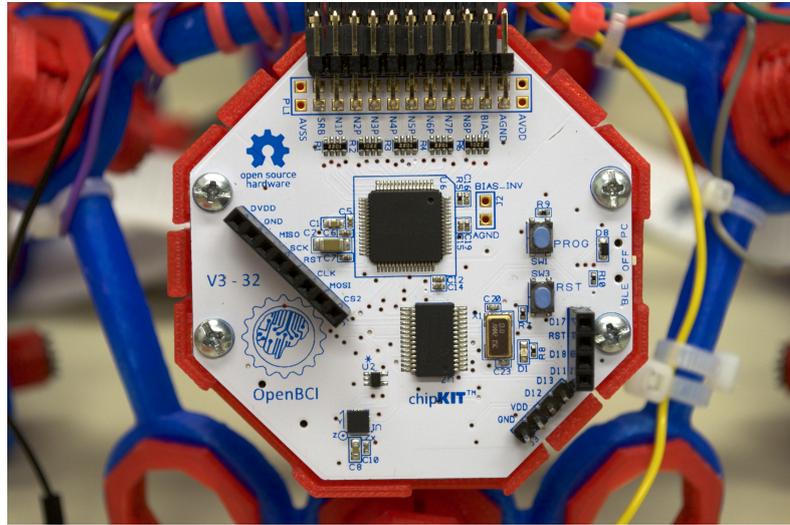


Figure 9: The OpenBCI board used as the EEG acquisition system in the proposed device [2].

O2, P8, T8, FC6, F4, F8, AF42. Since one of the most important components in detecting the P300 response is the electrodes locations, Emotiv Epoch would allow to choose other electrodes locations to be used.

For the above mentioned reasons, after this preliminary analysis we decided to use the OpenBCI board. The company which produces OpenBCI board bases its philosophy on open-source technology. So, complete specifications of the board, including the firmware and the protocol used for Bluetooth communication are available. The access to this information allowed us to read EEG raw data in an easy way. Then, it was also possible to develop a custom firmware for the board. Last but not least, we could decide which set of electrodes locations to use.

The OpenBCI board is based on the MicroChip PIC microcontroller model PIC32MX250F128B [45]. This microcontroller uses the ADS1299 [46] as the circuitry for analog to digital conversion.

OpenBCI board transmits the data using Bluetooth communication. A USB dongle plugged in into a computer is necessary in order to retrieve the data sent by the board.

The electronic board that we used is the Cyton board version 3, 32 - bit. The board is shown in Figure 9. One channel is used as reference for all the others, and another channel is used as the bias channel, that is similar to the ground channel on medical EEG devices. This channel makes use of destructive interference waveform techniques to eliminate the common mode noise of all the active recording channels.

One of the questions that we wanted to answer was if the OpenBCI signal quality is comparable to the one of a medical device. The quality of the signal is a fundamental component of a BCI. *Frey* [38] presented a study in order to fully address this question. In the study, the OpenBCI board was compared to a medical grade EEG system, the g.tec g.USBamp [41]. To perform the comparison, the EEG signals were contemporaneously recorded from the two devices in the same locations of the 10-20 systems. The main outcome of this study was that the OpenBCI board represents a good alternative to the most traditional EEG devices. Correlation computed both in time and frequency domain showed that the signals acquired by the g.USBamp and by the OpenBCI board were very closely related. The Pearson R score was higher than 0.99 for all the channels tested.

It is important to underline here that the OpenBCI board is not a medical device, and is not intended for that use: in fact, it has no certification.

Another important aspect is that the board uses a battery as a power source. Using a battery, hazards caused by the power supply can be avoided. Establishing a connection be-

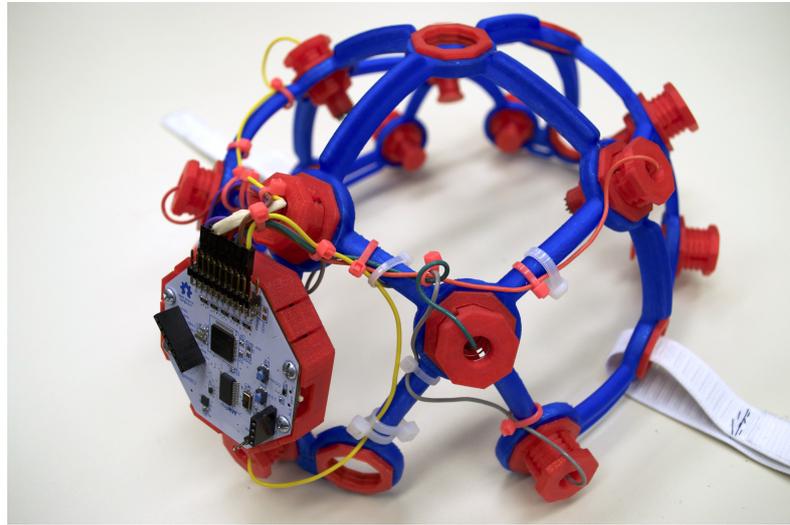


Figure 10: The 3D printed Mark III EEG headset that was utilized as a support for the electrodes. The chin strap that we included to add more stability to the headset is visible [49].

tween the human body and the power lines requires protection and isolation circuits, fulfilling international standards for safety. The battery that we used to power the OpenBCI board is a 500mAh lithium battery.

### **EEG Headset**

In order to record the EEG signal with the OpenBCI board, electrodes need to be precisely positioned on the scalp with an harness that limits motion artifacts. OpenBCI is an electronic board designed to record any type of biological signal (EEG, EKG, EMG,...) and therefore no EEG electrodes harness is supplied with the board.

For this reason, we had to determine which type of support to use. In order to do this, we identified three possible options:

- **Gold cup electrodes with harness:** OpenBCI comes with gold-cup electrodes. These electrodes typically have a low impedance, below  $20\text{ k}\Omega$ , and allow to obtain high quality EEG recordings. Conductive paste is required to be used with these electrodes. This paste has to be placed on the scalp, on the positions correspondent to the chosen electrodes locations. From the end-user point of view this is not a good solution, because it would require the constant presence of a care-giver who takes charge of the application of the paste and of the subsequent positioning of the electrodes. Furthermore, electrodes have to be fixed on the user's scalp using elastic bands. These elastic bands are not, in our opinion, suitable for long term recording. They may cause involuntary movements of the electrodes, decreasing the quality of the EEG signal;
- **Electro-caps:** these caps are head caps specifically designed for EEG recordings. They are used in hospital/laboratory recording of EEG, and allow to obtain high quality signals over long-term recordings. There are three main problems with these caps when designing a cost effective device: first of all, the market price associated to them ranges from 250 \$ to 400 \$, according to the chosen model; they also require an adapter, which costs around 100%, in order to be used with the OpenBCI board; lastly, they also require the use of a conductive gel to be applied between the cap and the scalp;
- **3D printed headset:** the last option was to use a 3D printed headset, acting as a support for the electrodes. Files for 3D printing are released by the OpenBCI company [2]. This headset can be easily 3D printed with adjustable sizes according to the user's head circumference. This headset makes use of dry electrodes, that don't require any

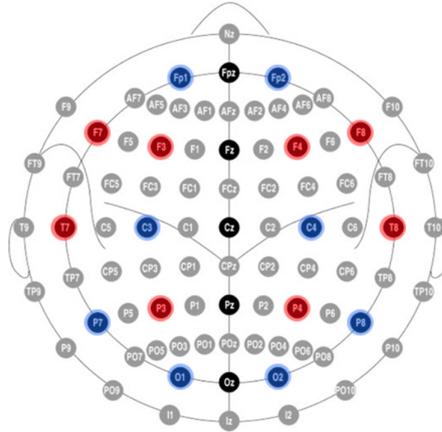
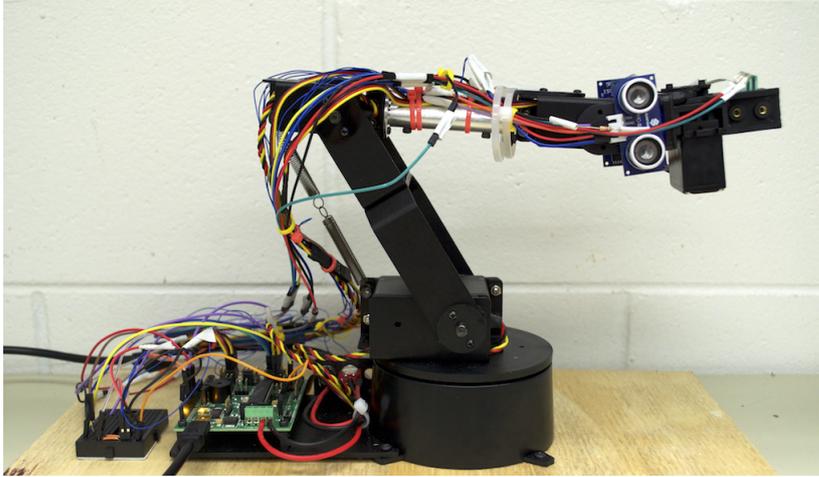


Figure 11: The available electrodes locations of the 3D printed EEG headset. The utilized locations are: C3,C4,P7,P3,Pz,P4,P8, Oz. Image from: <http://docs.openbci.com/Headware/02-Ultracortex-Mark-III-Nova-Revised>.

specific paste or gel to be applied on the scalp. Furthermore, printing and purchasing the material to build this headset costs around 100\$, which is far less than the cost necessary to buy an electro-cap. Therefore, we used the 3D printed headset model Mark III [49] for our wearable and cost effective device.

An image of the headset that we built and used to test our device is shown in Figure 10. With this headset, it is possible to place the electrodes in the following locations of the 10-20 system: Fp1, Fp2, F7, F3, Fz, F4, F8, T7, C3, Cz, C4, T8, P7, P3, Pz, P4, P8, O1, Oz, O2. These locations are shown in Figure 11.

The P300 response is mostly predominant on the parietal region and along the central line of the scalp [16]. For this reason, we choose the following electrodes locations: C3, C4 on the



Property	Value
Shoulder to Elbow	4.75"
Elbow to Wrist	5"
Wrist to Tip	3.375"
Height (reaching up)	15.75 "
Weight	23 oz

Figure 12: *Left*: picture of the Lynxmotion AL5B. *Right*: technical specifications of the AL5B Robotic Arm [42].

“central lobe; P7, P3, Pz, P4, P8 on the parietal lobe; Oz on the occipital lobe. As reference and ground, we used two ear clip electrodes placed on the right and left earlobes.

### 2.1.2 Robotic Arm

One application of BCI our device is to control a robotic arm. We choose to use a robotic arm produced by Lynxmotion: the AL5B arm [42]. The technical specifications of the robotic arm are shown in Figure 12b. A picture of the complete setup for the robotic arm is shown in Figure 12a. The robotic arm has 4 degrees of freedom. One DoF is due to the rotation of the base, and the other three are due for three joints that act as shoulder, elbow and wrist. An additional DoF can be added with a servo motor for the rotation of the wrist.



Figure 13: The HCSR04 ultrasound sensor.

The movement of each joint is obtained by setting the angle of the correspondent servo motor. In order to achieve this, the servo motors have to be controlled with an electronic board. The electronic board that we used to control the servos is the BotBoarduino microcontroller [43]. This board is based on the Arduino Duemilanove [44], and allows to control a set of servo motors in an easy way.

We powered the board directly from the USB port of a laptop, and we used a 6V power supply to power the 5 servo motors of the robotic arm. This distinction in the power supply was necessary because the servo motors could take away the current necessary for the electronic board to function properly.

In order to enhance the functionality of the robotic arm, we added ultrasonic and force sensors. Two ultrasonic sensors are used to avoid crashes of the robotic arm with close objects, while two force sensors are used to detect the width of the object being grabbed.

The ultrasonic sensors are shown in Figure 13. Each ultrasonic sensor is placed on one side of the robotic arm, in a position close to the gripper. These sensors are used to detect the presence of any object in proximity of the robotic arm. The chosen proximity threshold is 4”.

The principles behind ultrasonic sensors is very simple. An ultrasonic sensor sends high frequency sound pulses at a regular time interval. If the sound wave strikes an object, the pulse is reflected back. By determining the time difference between the sent and received signal, one can compute the distance of the object, according to this equation:

$$d = \frac{t \cdot v}{2} \quad (2.1)$$

where  $d$  is our unknown variable (distance),  $t$  is the measured time and  $v$  is sound speed in the air, which is approximately 767 mph. The 2 in the denominator is necessary because the sound wave has to travel back and forth in the air in order to be detected.

On the gripper of the robotic arm, we added two force sensing resistors. These sensors was necessary to detect the width of the object being grabbed, in order to avoid crashing the object. This technique prevents the servo motor to overheat, that could happen when the servo motor keeps moving while an object has already been grasped by the gripper.

A software was developed by our team in Processing [50] to control the robotic arm. This software communicates via a serial port with the BotBoarduino board controlling the servo motors of the robotic arm. A screenshot of the software is visible in Figure 14 This software was designed with the following idea: each icon corresponds to a discrete movement of the

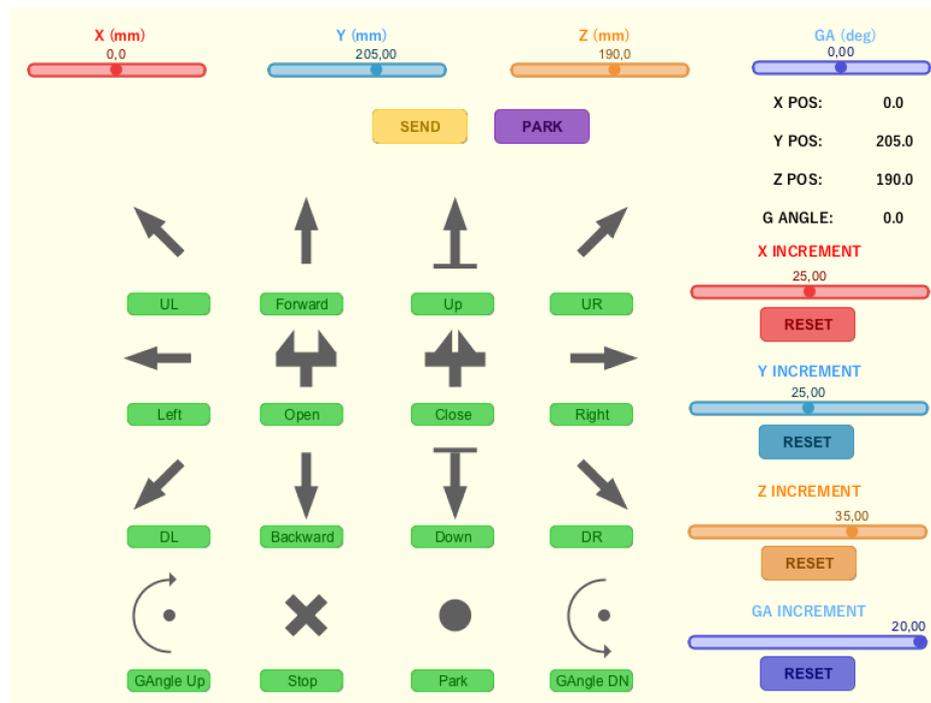


Figure 14: The interface of the software developed for robotic arm control. Buttons for discrete movements, sliders for setting increment values, sliders for setting a specific position can be used to tune the parameters in an optimal way.

robotic arm, such as *up*, *left*, . . . . A complete description of the discrete movements that can be performed by the robotic arm is reported in Appendix (Table XI).

When a button is clicked, the updated  $X$ ,  $Y$ ,  $Z$  coordinates and the value of the wrist angle are sent to the robotic arm, and the arm would move accordingly. The current position of the robotic arm is always displayed and updated with the current position of the robotic arm. The software also allows to control the incremental value for the discrete movement along the three axis and for the angle of the wrist. So, it was possible for us to identify the best set of values to be used in the BCI control of the robotic arm. In fact, a small increment value would have resulted in the possibility of performing finer movements, but at the expense of a longer time. A large increment, instead, would resulted in a smaller amount of time necessary to cover the same distance, but at the expense of movement resolution. Experiments were performed in order to identify the set of increment values leading to the best trade off between time and movement resolution.

Some locations in the three-dimensional space may not be accessible to the robotic arm, since reaching that location may be obstructed by close objects. Therefore, it is important to read back the correct values of  $X, Y, Z$  coordinates and the wrist angle, which are sent by the BotBoarduino board.

Furthermore, the software allows to directly specify the  $X, Y, Z$  coordinates and the wrist angle the robotic arm should reach. This resulted useful to us to determine the extreme locations that can be reached by the robotic arm.

## 2.2 Software

After setting up the EEG acquisition system and the robotic arm, we implemented the BCI software. The main goals of the software are the following ones:

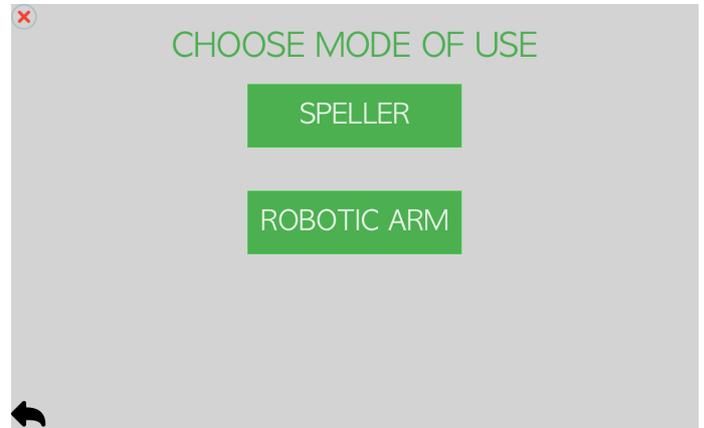
- present to the user visual stimuli using a flashing grid of options. When the stimulus representing the user's intent flashes, a P300 is elicited;
- acquire EEG signal streamed by the OpenBCI board;
- process the EEG signal. In particular, classification is aimed at the identification of the P300 response;
- communicate with the robotic arm, sending commands to it and receive information back from it.

We analyzed several programming languages and software solutions that could be employed to develop our BCI application. After this analysis, we decided to use Processing [50], the same software that we used for the robotic arm control application. According to Processing website, "Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts" [50]. Processing is mainly used to develop graphic-based application. With Processing, it is also possible to use tools to communicate with external hardware through serial communication. The combination of these factor leads us into choosing Processing as the software to develop our BCI application.

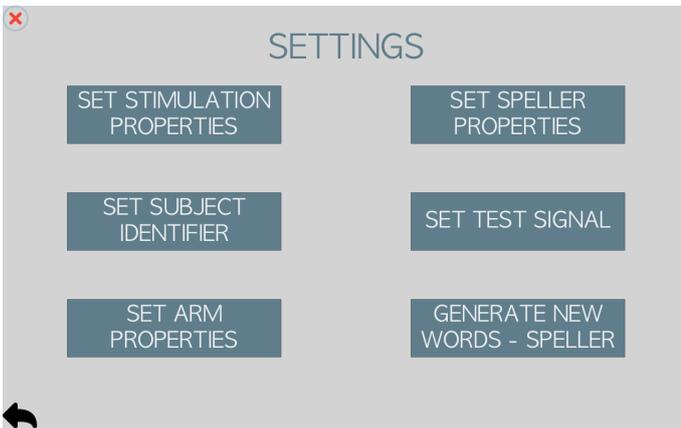
Some screenshots of the developed BCI software are presented in Figure 15. These windows are as follows:



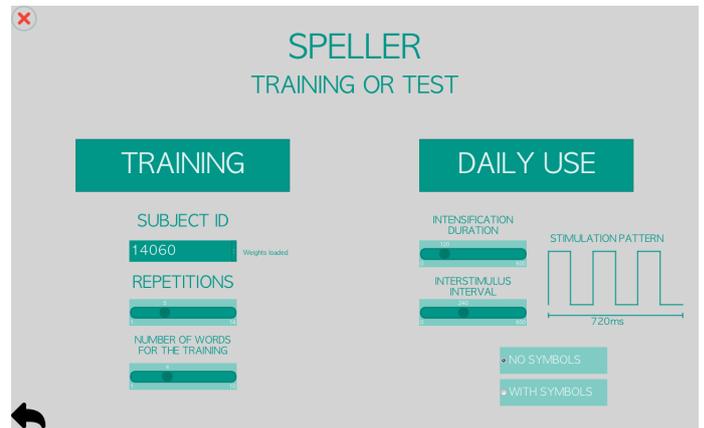
(a) Main menu of the BCI.



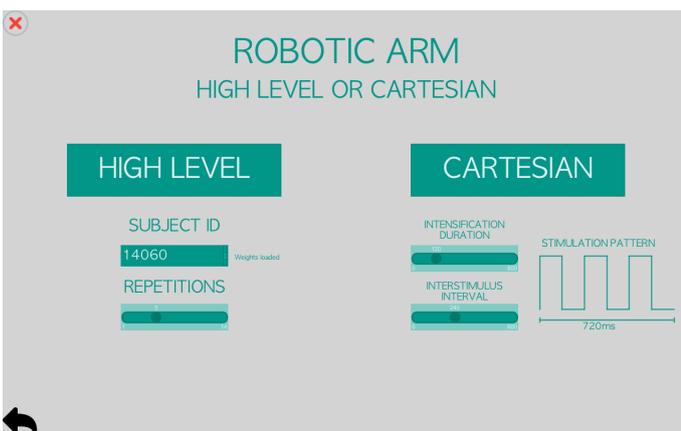
(b) Possible modes of BCI use.



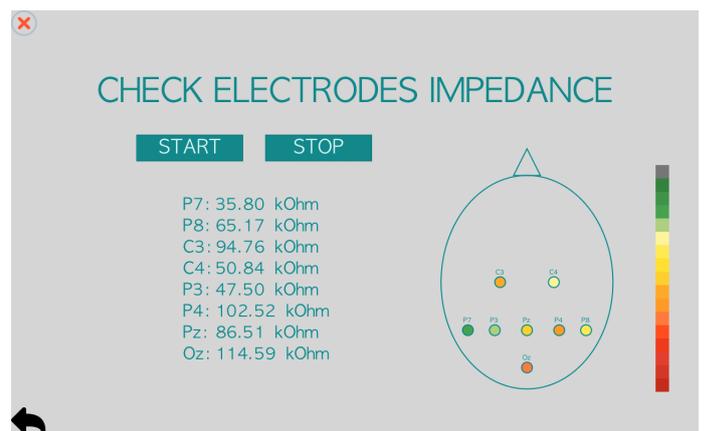
(c) Settings.



(d) Settings for speller mode.



(e) Settings for robotic arm mode.



(f) Impedance Check.

Figure 15: Screenshots of the developed BCI software.

### A. Main Menu Window

The main menu is shown in Figure 15a. This is the first screen presented to the user. The user can choose to start the system or to enter the menu to set some settings related to the parameters for the stimulation, to the robotic arm control, . . . . Furthermore in the bottom left corner it is possible to check if the USB Dongle, the OpenBCI board and the robotic are connected or not.

### B. Start Menu Window

If the start button is chosen in the main menu, the screen visible in Figure 15b appears. This window presents to the user two modes of operation: *Speller* and *Robotic Arm Control*. If the *Speller mode* is chosen, the user can use a P300 speller similar to the one developed by *Farwell and Donchin* in [17]. The speller mode is also used for the training phase of the EEG classifier. This will be discussed in Section 2.3.2.1. The second possible option is the *Robotic Arm Control*. When chosen, the user gains access to a P300 based control of the robotic arm. So, according to the user's needs, it is possible to choose between the control of the robotic arm and the use of the speller.

### C. Settings Window

When entering the settings page, shown in Figure 15c, it is possible set some options. These options are related to: stimulation parameters, such as inter stimulus interval, intensification duration, number of repetitions; properties for the training of the classifier done with the speller, such as the number of words to be used for the training; subject ID, which is necessary to save the data and to load the parameters for classification; test signal on the board, which is useful

to verify that the communication with the board is working; properties for the control of the robotic arm, such as the incremental values for the discrete movements.

#### **D. Speller Mode Window**

Speller mode window is shown in Figure 15d. When entering speller mode, the user can set stimulation parameters, among these parameters there is the number of repetitions. When dealing with evoked potentials, it is important to know that it is very difficult to identify the presence of an evoked potential after only one presentation of the stimuli. In order to identify the presence of evoked potential with a great accuracy, multiple repetitions of the same stimuli need to be presented to the user, since detection of an evoked potential with just one presentation of the stimulus is extremely difficult. In our case, the term *repetition* refers to a complete set of flashes: this means that, if the stimulation grid has 6 rows and 6 columns, a repetition would be made of the total 12 flashes (i.e., stimuli). In a single repetition, each column and each row flashes, so it is not possible for a column or a row to flash multiple times during the same repetition. By increasing the number of repetitions used, and a result increasing the number of stimuli presented to the user, a P300 response can be identified in a more accurate manner, but this comes at the expense of more time required. The possible choices for the number of repetitions to be used range from 1 to 14.

For the speller mode, it is possible to choose between the training and the test phase. During the training phase, the user is asked to spell predetermined letters. This is used in order to perform the training of the classifier, and will be explained in detail later in the chapter. While during the test phase, the user can spell any desired word.

### **E. Robotic Arm Mode Window**

The robotic arm mode window is shown in Figure 15e. When entering the robotic arm mode, the user is presented with the options to set some parameters, such as number of repetitions and subject ID. Then, the user is also provided with an option to choose between high level control and cartesian control of the robotic arm. The difference between the two modes will be explained later in the chapter.

### **F. Impedance Control Window**

The last screenshot, visible in Figure 15f, shows the electrodes impedance check. It is very important to have good quality signals obtain good accuracy in P300 detection. One way to check if the electrodes are placed correctly on the scalp is to measure the electrode-to-skin impedance. In order to measure the impedance, a small, and known, current is injected in the electrode, and the resulting voltage is measured. Since both voltage and current are known, it is then possible to compute the electrode-to-skin impedance as:

$$R = \frac{V}{I} \quad (2.2)$$

By using a color-based legend, the user can determine which electrodes have to be adjusted before starting to use the OpenBCI for recording.

Now we'll discuss in detail how we implemented the visual stimulation, both for speller and robotic arm mode.

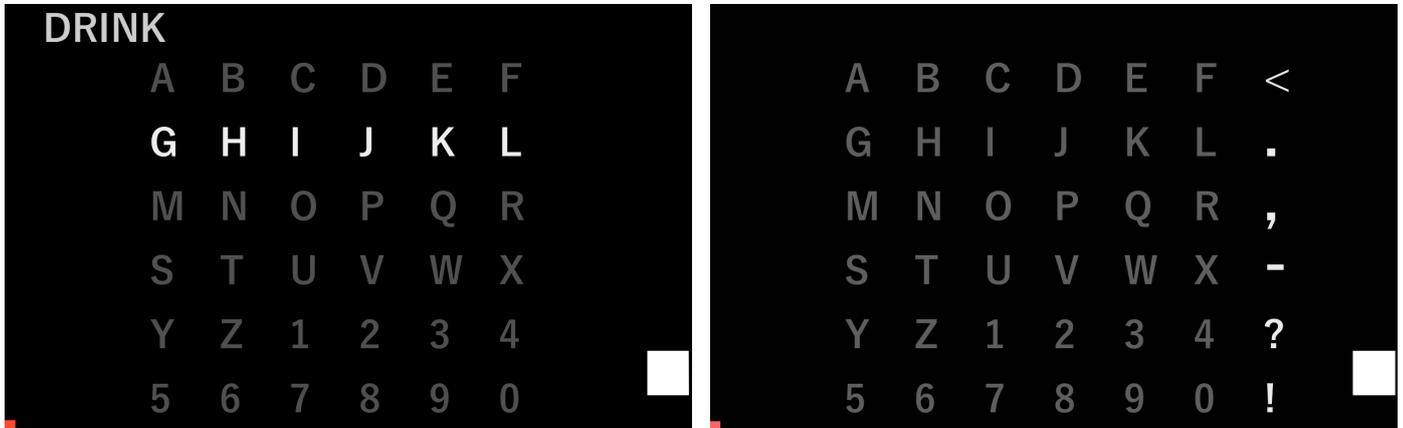


Figure 16: *Left*: speller stimulation grid without symbols. *Right*: speller stimulation grid with symbols.

### 2.2.1 Visual Stimulation

To elicit a P300 response an oddball paradigm is frequently used. With this paradigm, various stimuli are presented to the user. From the user’s point of view, one stimulus represents the *target*, i.e. the one the user is focused on. All the other stimuli are non-target. Considering all the stimuli presented to the user, the target stimulus occurs infrequently. When presented with this stimulus, a P300 response is elicited [16].

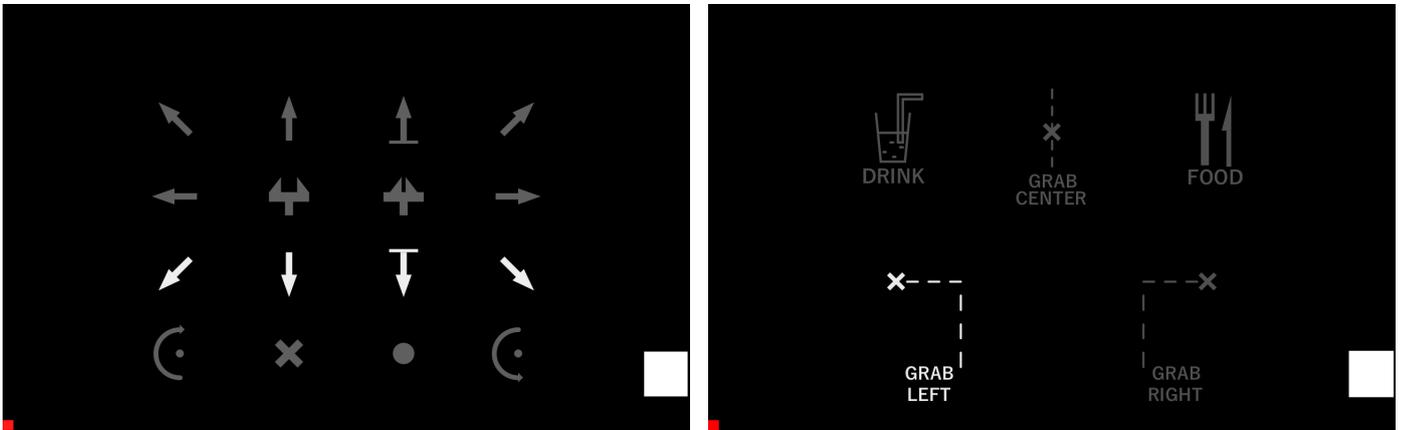
For our wearable BCI device, we employed visual stimulation. Stimuli are arranged in a  $n \times m$  grid. For speller mode and cartesian control of the robotic arm, we decided to flash columns and rows, instead of single cells, in order to reduce the time required to perform a complete stimulation. With this paradigm, the generation of a P300 response would be associated to the flashing of a row or of a column, rather than to the specific cell.

In the speller mode, it is possible to use two different types of grids: one grid is a  $6 \times 6$  grid containing only letters and numbers; the other one is a  $6 \times 7$ , grid which, other than letters and numbers, also has 4 symbols ( . , - ? - ! ), a symbol to delete the last letter ( < ), and the space character (-). These two grids are visible in Figure 16.

For the robotic arm mode, two different control modes can be used, *cartesian control* and *high level control*. For the cartesian control, the grid is a  $4 \times 4$  grid, in which 16 movements are represented. The 16 movements are the same as shown in the robotic arm control software, and allow to perform discrete movements with the arm. A screenshot of the flashing grid is shown in Figure 17. Also in this case, columns and rows flash, and a single repetition is composed of 8 flashes.

In the high level control, instead, the subject is not asked to perform a series of discrete movements to complete the task. The idea behind the implementation of this mode is that the subject can directly perform a complete, and predefined, action with a single command. These commands are represented in Figure 17, and are as follows:

- Grab a glass of water: the robotic arm will move to a location, already defined in the firmware, in which a glass of water is expected to be present. After grabbing the glass, it will move towards the subject mouth;
- Grab food: as above, the robotic arm will perform a complete movement to a location where food is expected to be present. After picking up the food, the robotic arm will move towards the subject;



(a) The grid for cartesian robotic arm control.

(b) The grid for high level robotic arm control. Single cell flashing is used.

Figure 17: The two stimulation grids that can be used in the robotic arm control mode.

- Grab an object on the left: the idea behind this high level command is that if a person wants to give to the user an object, the robotic arm will move to a defined location on the left, and wait until an object is grabbed;
- Grab an object in central position: this movement is the same as above, but the robotic arm will move to a central location;
- Grab an object on the right: as above, but on the right.

Regarding the stimulation in high level robotic arm control, single cell flash one at the time, instead of rows and columns. So, a single repetition is composed of a single flash of each cell. While the robotic arm is performing one of these high level actions, the stimulation is stopped until the movement is complete.

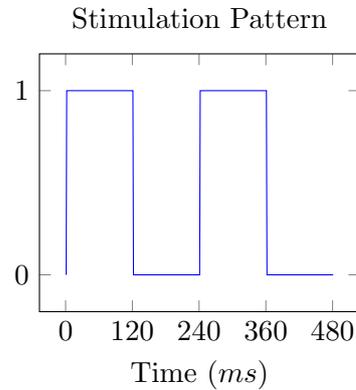


Figure 18: The default flashing pattern used in both speller and robotic arm mode.

With the current implementation of the robotic arm firmware, it is required that the objects are in a position already defined in the firmware: otherwise, the robotic arm won't know where it has to move to find the glass of water, the food, . . . .

A visual representation of the flashing is shown in Figure 18. As a default, stimuli are presented with this flashing pattern: a random column, row or cell intensifies with a white color for 120 ms. This intensification happens every 240 ms. These two parameters (intensification duration and inter stimulus interval) can be adjusted in the settings page . A study on the effect of the stimulus rate on P300 response was done by *McFarland et al.*. They showed that the optimal inter stimulus interval is between 62.5 and 250 ms.

### 2.2.2 EEG Data Retrieval

The 32 bit OpenBCI board that we used for our wearable BCI device makes use of bluetooth connection to communicate EEG data. A USB dongle plugged in the laptop to retrieve the

TABLE II: DESCRIPTION OF THE DATA PACKET SENT BY THE OPENBCI BOARD [2].

Byte(s)	Value	Byte(s)	Value
1	0xA0	18 - 20	Channel 6
2	1 - 255	21 - 23	Channel 7
3 - 5	Channel 1	24 - 26	Channel 8
6 - 8	Channel 2	27 - 28	Auxiliary
9 - 11	Channel 3	29 - 30	Auxiliary
12 - 14	Channel 4	31 - 32	Auxiliary
15 - 17	Channel 5	33	0xC0

data. Then, in our software we implemented a serial communication with the USB dongle with a baud rate of 115200. Each data packet sent by the board is composed of 33 bytes. Table II provides the complete description of the data packet.

In order to parse the data contained in the data packet, the start and stop bytes, respectively 0xA0 and 0xC0, are used. Throughout all of our tests, we've never experienced any data loss or incorrect retrieval.

### 2.2.3 Synchronization Protocol

Synchronization between the recorded EEG and the stimulation time is a fundamental component for analyzing evoked potentials. If this condition is not met, then it would be impossible to correctly detect the evoked potential in the EEG signal, because time information would be misleading. As an example, with P300 responses we expect a positive deflection in the EEG signal at around 300 ms. It is easy to understand that, if time information about

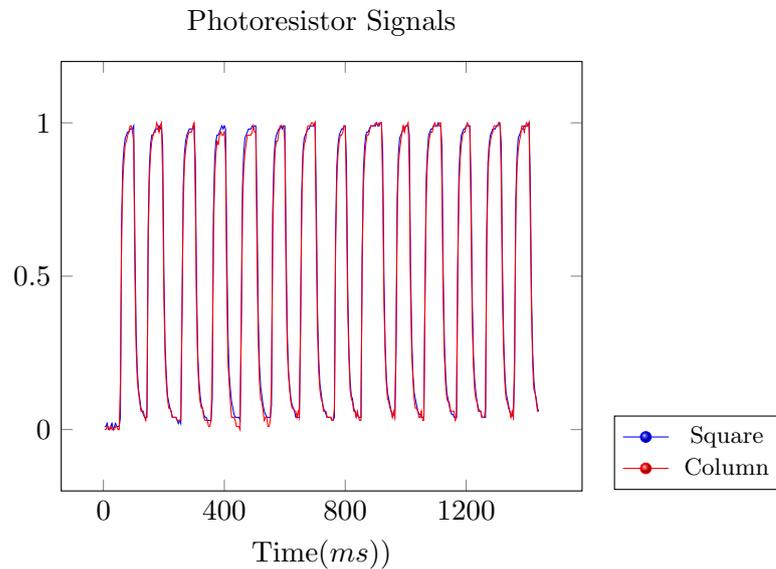


Figure 19: Signals recorded simultaneously from the flashing square and column. Photoresistor signals are normalized between 0 and 1.

stimulation is incorrect, then it would be impossible to correctly identify the presence of a P300 response in the EEG trace.

A synchronization protocol had to be set up. In our application, the synchronization protocol uses one of the available analog channels on the board to read the light activity generated by the flashing stimulation. In order to implement this, we added a small square in the bottom right corner of the grid flashing at the same frequency of the columns and rows flash. This square is visible both in Figure 16 and Figure 17. Using a simple photoresistor and an analog pin on the board, the photoresistor signal can be read directly with the board together with the EEG signal, and in this way synchronization between stimulation and EEG is established.

In order to assure that the square flashed at the same frequency of the columns and rows, we performed the following test: we simultaneously recorded, using two photoresistors, the light signal of the square and of a flashing column of the matrix. The result is visible in Figure 19. The two signals overlap, meaning that the square and column are flashing exactly with the same frequency. So, this method represents a reliable technique to synchronize the EEG signal with the visual stimulation provided to the user.

### 2.3 Classification

Detection of P300 event-related potentials with high accuracy is a fundamental component of any BCI system. In order to do so, classification of the EEG signal is required to detect the P300 responses. A classifier is a function that maps generic input data to a category. In our case, the input data are EEG signals, and the categories are presence or absence of P300 response. A *training* set is a set of data previously classified into categories. Using the *training* set, the classification models generate predictive rules that would be used to determine the category of future unclassified data [51]. This process is referred to as training of the classifier. After the training phase, the classifier is tested on a *validation* data set. The *validation* data set is composed of classified examples that were not used during the training of the classifier.

In mathematical terms, for a classification problem the training set is made up of  $m$  previously classified examples,  $(\mathbf{x}_i, y_i), i \in M$ . The elements of the vector  $\mathbf{x}_i \in \mathfrak{R}^n$  represent the values of the  $n$  predictive attributes for the  $i$ th example;  $y_i \in H$  defines the category correspondent to the vector  $\mathbf{x}_i$ . Let us define the *hypotheses* function  $F$  as the class of functions  $f(\mathbf{x}) : \mathfrak{R}^n \mapsto H$ . This class of functions  $F$  represents possible relationships between  $y_i$  and  $\mathbf{x}_i$ .

In a generic classification problem, the aim is to define an appropriate hypothesis space  $F$  and an algorithm  $A_f$ , that ,together, would allow to determine a function  $f^* \in F$  that specifies the relationship between the value of the  $n$  predictive attributes and the corresponding categories [51].

If we define  $\alpha$  as a parameter, or a vector of parameters, the function  $f$  is dependent on, the optimal classifier is the function  $f^*(\mathbf{x}, \alpha_0)$ , where  $\alpha_0$  is the parameter or the vector of parameters that minimizes the loss function:

$$\begin{aligned} R(\alpha) &= E[|y - f(\mathbf{x}, \alpha)|] = \int |y - f(\mathbf{x}, \alpha)| dp(\mathbf{x}, y) \\ &= \int \int |y - f(\mathbf{x}, \alpha)| p(\mathbf{x}, y) d\mathbf{x} dy \end{aligned} \quad (2.3)$$

This loss function depends on the parameter or the vector of parameters  $\alpha$ , but also on the distribution  $p(x, y)$ , which is a prior unknown. Therefore, the loss function is approximated with the expected mean value of the error on the training set. Equation 2.3 then becomes:

$$R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i, \alpha)| \quad (2.4)$$

We can see from Equation 2.3 and Equation 2.4 that  $R_{emp}(\alpha) \xrightarrow{N \rightarrow \infty} R(\alpha)$ , so it could be a good choice to approximate  $R(\alpha)$  with  $R_{emp}(\alpha)$ . However, for a finite number  $N$ , the difference between the terms  $R_{emp}(\alpha)$  and  $R(\alpha)$  can be very high. This could lead to a phenomenon

referred to as *overfitting*, that occurs when the chosen classifier has a perfect performance on the training set, but it is not able to generalize and correctly classify future examples of  $\mathbf{x}$ .

To avoid the problem of *overfitting*,  $k$ -fold cross validation is typically used. With  $k$ -fold cross validation, it is guaranteed that each observation  $\mathbf{x}_i$  is used the same number of times in the training set and only one time in the test set. The complete dataset is divided into  $k$  disjoint subsets, and at the  $j$ th iteration of the procedure, one of the  $k$ th subsets is used as the test set, while the union of the other subsets is used as the training set. At the end of the cross-validation procedure, the accuracy of the classifier is estimated with the average value of the accuracy on the  $k$  test sets [52].

### 2.3.1 Logistic Classifier

In our application, we use logistic regression as the classification technique to detect P300 responses in the EEG signal. In logistic regression, the binary response  $y$  to a vector  $\mathbf{x}$  (of  $n$  values) is explained using a Bernoulli distribution with  $E(y) = P(y = 1) = p$ . Considering the  $i$ th example,  $E(y_i|X = \mathbf{x}_i) = P(y_i = 1|X = \mathbf{x}_i) = p_i$  is a number between 0 and 1. Therefore, considering a cumulative distribution function (CDF),  $F$ , it is possible to model  $p_i$  as  $F(\beta_0 + \beta\mathbf{x}_i)$ . Any  $F$  can establish a relationship between  $p$  and  $\mathbf{x}_i$ . In the case of logistic regression, the used CDF is the logistic distribution. For logistic regression, the operator  $\log\frac{p}{1-p}$  is called the *logit* operator.

Therefore, for logistic regression the followings are valid:

$$y_i \sim Ber(p_i) \quad (2.5)$$

$$\text{logit}(p_i) = \log \frac{p_i}{1 - p_i} = \beta_0 + \beta \mathbf{x}_i \quad (2.6)$$

Upon estimating the parameters  $\beta_0$  and  $\beta$ , the probability  $p_i$  can be computed as:

$$p_i = P(y = 1 | X = \mathbf{x}_i) = \frac{1}{1 + e^{\beta_0 + \beta \mathbf{x}_i}} \quad (2.7)$$

$$\begin{aligned} 1 - p_i &= P(y = 0 | X = \mathbf{x}_i) = 1 - P(y = 1 | X = \mathbf{x}_i) \\ &= \frac{e^{\beta_0 + \beta \mathbf{x}_i}}{1 + e^{\beta_0 + \beta \mathbf{x}_i}} \end{aligned} \quad (2.8)$$

Therefore, it is possible to assign the vector  $\mathbf{x}_i$  to category 0 or 1 according to the value of the computed probabilities.

The vector  $\beta$  can be found by maximizing its log likelihood:

$$L(\beta) = \sum_{i=1}^N \log P(y_i | \mathbf{x}_i, \beta) \quad (2.9)$$

In practice, Equation 2.9 is rarely used, and a penalty term is added to avoid high values of  $\beta$ . Equation 2.9 then becomes:

$$L(\beta) = \sum_{i=1}^N \log P(y_i | \mathbf{x}_i, \beta) - \lambda \|\beta\|^2 \quad (2.10)$$

### 2.3.2 Genetic Algorithms

The first step in applying classification is feature selection. Performing feature selection by hand can be sub-optimal, and cumbersome. For this reason, we used the genetic algorithm presented by *Dal Seno et al.* in [3] to perform automatic feature selection for P300 response detection.

In this section, we'll introduce genetic algorithms in a general terms, explaining the required terminology, and in the following section we'll describe the specific genetic algorithm that we have used for automatic feature extraction. In this description, we will refer to the generic structure of genetic algorithms, represented in Figure 20.

In nature, survival of the fittest happens as a result of selection, crossover, and mutation among the individuals of the population. Genetic algorithms, which belong to the family of evolutionary algorithms, mimic this natural process in order to identify an optimal solution to a presented problem starting from a population of candidate solutions.

In a genetic algorithm, a solution is represented as a fixed-length string of values, describing an individual in a population. Each value of the string is related to a specific characteristic of the individual, and the overall combination of the characteristics determines the solution to a problem. The key elements in a genetic algorithm are *individuals* and *populations*, that represent a collection of individuals.

#### **Individuals**

In nature, an individual (i.e., an organism) is characterized by a set of rules that describe completely the individual. These rules are encoded in the *genes* of the organism, and these

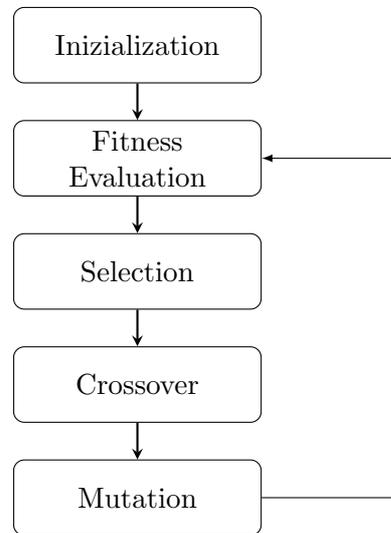


Figure 20: Generic structure of a genetic algorithm.

genes are connected together to form the *chromosomes* of the organism. The genes are typically referred to as the *genotype* of the organism, while the physical expression of the genotype, which represents the organism, is referred to as the *phenotype* [54].

In a genetic algorithm, candidate solutions to the presented problem are represented by individuals. Each individual is characterized by two representations of the solution:

1. chromosome, that describe the solution in terms of bit or values;
2. phenotype, that represent the solution in the correct terms as defined by the problem.

The phenotype space is linked to the solution space by a morphogenesis function, that associates the genotype to the phenotype.

## Genes

The *genes* represent the basic building block in the development of a genetic algorithm. A gene describes a possible solution to the problem as an arbitrary length string of values. The structure of this string defined in terms of parameters that are necessary to map the genotype to the phenotype.

## Fitness

In genetic algorithms, *fitness* measure is used to evaluate each chromosome (i.e., each solution). To do so, a chromosome must be firstly decoded, and then an objective function has to be defined to compute the fitness value.

## Population

A population is made up of individuals, that are tested in order to find the optimal solution to the presented problem. Two important characteristics of the population have to be taken into consideration: the initial population and the size of the population. The population size depends on the complexity of the problem. The larger the population is, the easier it is for the genetic algorithm to explore the solution space. But, having a very large population, even though it helps in reaching global optimum, results in more computational cost, memory and time. The typical value of population size is around 100 individuals [54]. Regarding the initial population, that is often randomly initialized at the beginning of the genetic algorithm as visible in Figure 20, it should present a large gene pool so that it is possible to explore the whole solution space. For this reason, the population is typically chosen using random techniques. If an heuristic method is used to seed the initial population, the resulting average fitness of the

population is already high, and the genetic algorithm will converge to an optimal solution in a faster way.

## Breeding

The breeding process represents the fundamental component of the genetic algorithm. During this process, new individuals, ideally with a better fitness, are created. The main steps for the breeding process are shown in Figure 20:

- Selection of parents;
- Crossover of parents to generate the children individuals and mutation of the new individuals;
- Replacement of the old individuals with the newly generated ones

We describe here in detail these steps:

1. **Selection:** it is the process of picking two parents from the old population in order to recombine them. The idea at the basis of selection is to choose individuals with a high fitness, with the hope that the generated children will have, in turn, an higher fitness than their parents. Typically, the higher the fitness of an individual, the higher the probability of that individual to be chosen. We can define as *selection pressure* the degree to which the fitter individuals are favored in the selection process. The selection pressure drives the convergence rate of the genetic algorithm: higher selection pressure results in higher convergence rate. A problem, that could result when choosing a high selection pressure, is that it is more probable that the genetic algorithm will converge to a local optimum

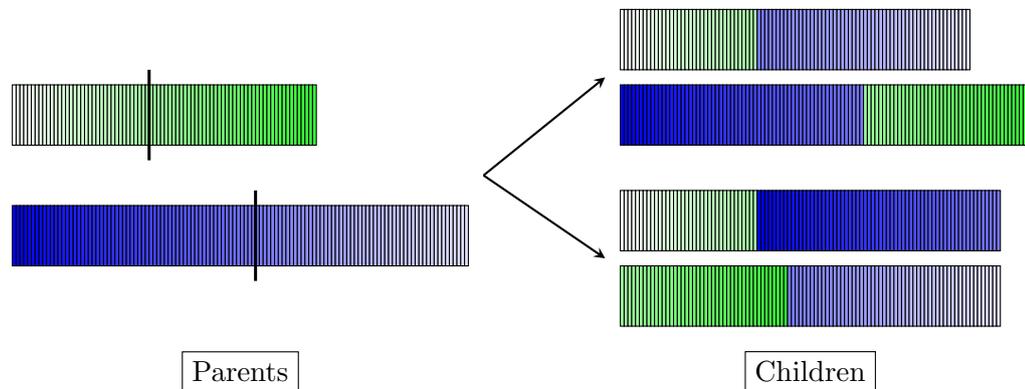


Figure 21: Crossover process in genetic algorithms.

[54]. Instead, if a low selection pressure is chosen, then the resulting convergence rate would be very low. *Elitism* is a process that takes part in the selection. It represents the practice of keeping the best chromosomes from the old generation, and transfer them to the new generation. In fact, if these individuals are not selected, or if crossover or mutation destroy them, they would get lost.

2. **Crossover:** after selection is performed, crossover takes place. Selection makes clones of the fitter individuals, but doesn't create new ones. With crossover, recombination of the individuals is performed in order to generate children with higher fitness. The simplest way to perform crossover is with single point crossover, which is shown in Figure 21. Three steps can be identified in this crossover operation: firstly, two individuals are chosen at random from the mating pool; secondly, two random points in the genes' strings are chosen; thirdly, the resulting parts are recombined together to form two children, by exchanging the sections after the cut point. An important parameter in the crossover

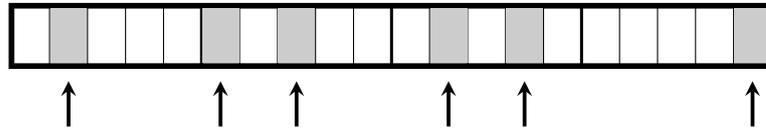


Figure 22: Mutation process in genetic algorithms.

operation is the probability that an individual would be selected to undergo crossover: if this value is 0%, then the new population would result in a copy of the old generation; if it is equal to 100%, then all the old individuals undergo crossover. An intermediate value helps in having a good diversity in the population, by creating new individuals as well as keeping some of the old ones.

3. **Mutation:** after crossover is performed, mutation, shown in Figure 22, occurs. The purpose of mutation is to help the genetic algorithm to explore the solution space, by recovering lost genetic material and, also, to disturb the genetic information. Due to the process of mutation, the genetic algorithm is prevented from being stuck in a local minimum. Mutation works by changing the value of each gene with a predetermined, and typically small, probability.
4. **Replacement:** the last stage of the breeding process is the replacement. Replacement is necessary because, since the population size is constant, it's not possible that both parents and children take part in the new population. Therefore, a criterion to determine which individuals will proceed to the new generation has to be identified. The two main techniques that can be used in the replacement process are generational updates and

steady state updates. In the case of generational updates,  $N$  children are produced from a population of size  $N$ , and so the new population completely replaces the old one. There are also derived forms of generational updates, that may consider the fittest individuals from both the children and the parents. In the case of steady state updates new individuals enter the new population as soon as they are created. This is different from the generational update, since in that case an entire new generation is produced at each generational step. In order to perform a steady state update, it is necessary to determine which individual has to be deleted. The individual to be deleted can be chosen as the worst or the oldest one, but these methods are typically drastic. For this reason, a tournament method is typically set up. A tournament between a fixed number of individuals takes place, and the individual with the highest fitness is chosen as the one who will proceed to the next generation.

### **Stop Criterion**

An important parameter that has to be defined for genetic algorithms is the stop criterion. There are several possibilities that can be employed to determine when the algorithm will stop:

- maximum generations: the evolution of the algorithm will stop when the defined maximum number of generations is reached;
- no change in fitness: if no change in the population's fitness values is observed, the algorithm will stop;

- maximum time: the algorithm will finish its search for the best solution if the defined maximum time is reached.

### 2.3.2.1 Genetic Algorithm for Automatic Feature Extraction

As already underlined at the beginning of the previous section, we used the genetic algorithm described by *Dal Seno et al.* in [3] to perform automatic feature extraction for P300 identification. In this approach, a genetic algorithm operates on simple feature extractors, and through the evolution process it allows to find the optimal set of feature extractors to be used in logistic regression classification. The only preprocessing that is done on the signal is bandpass filtering with cutoff frequencies of 0.5 and 30 Hz. In their work, the authors took inspiration from previous studies, where genetic algorithms were used to find the best combination between features and classifier for motor-imagery task [55] or to find suitable features for P300 detection [56].

#### **Feature encoding**

In this genetic algorithm, the structure of a chromosome is shown in Figure 23. A chromosome represents a possible solution to the defined problem. In this case, the problem is finding the best set of features to be extracted from the signal to perform classification. A chromosome encodes, through its genes, a set of features. A chromosome is made up of a certain number of genes, and each gene has the same identical structure with five elements. As shown in Figure 23, the first three elements of a gene allow to define a function that acts as a feature extractor. The first integer in the gene (*Func.*) defines one function for feature extraction, while the following two arguments (*Arg.1* and *Arg.2*) are two parameters for that function, that lie in the range

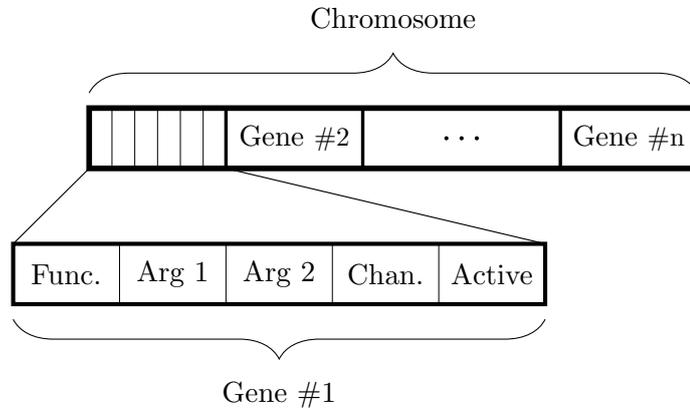


Figure 23: Structure of a chromosome in the genetic algorithm for automatic feature extraction [3] . ©[2011] IEEE

[0, 1). The fourth element of a gene determines the EEG channel from which that feature has to be extracted from. The last element of the gene determines if that gene is active or inactive. Inactive genes are not considered for fitness computation. The role of inactive genes is of genetic reserve, since they can be turned on later during the evolution. The feature extractors all share the same basic structure. For each feature extractor, the input signal is cross-correlated with a weight function. The operation performed by the feature extractor is the following:

$$x = \sum_{t=1}^T u(t)s(t) \quad (2.11)$$

where  $x$  is the resulting feature,  $u(t)$  is the weight function and  $s(t)$  is the EEG signal at the input. So, the resulting feature can be seen as the correlation between the input signal  $s(t)$  and the weight function  $u(t)$ . For our implementation of the genetic algorithm, an epoch is

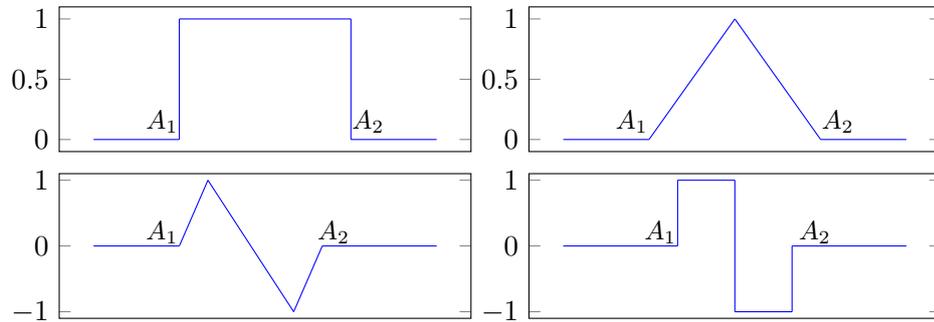


Figure 24: Weight functions used in the genetic algorithm.

considered 1 second long. Since the sampling frequency of the OpenBCI board is equal to 250 Hz, an epoch is made up of 250 samples. Every epoch is extracted in the time interval around the stimulus onset. In particular, an epoch goes from 200 ms prior to the stimulation to 800ms after the stimulation.

The four weights function that we used are represented in Figure 24. The top left weight function computes the average value of the input signal on the interval determined by parameters  $A_1$  and  $A_2$ , while the top right weight function computes the average of the signal by considering the values in the central portion of the interval with a greater weight. The bottom functions compute the differences in the signal of two adjacent intervals.

### Fitness

The fitness measure of each chromosome is chosen as the performance of a logistic classifier using the set of features encoded by the genes of the chromosome. To consider a better measure

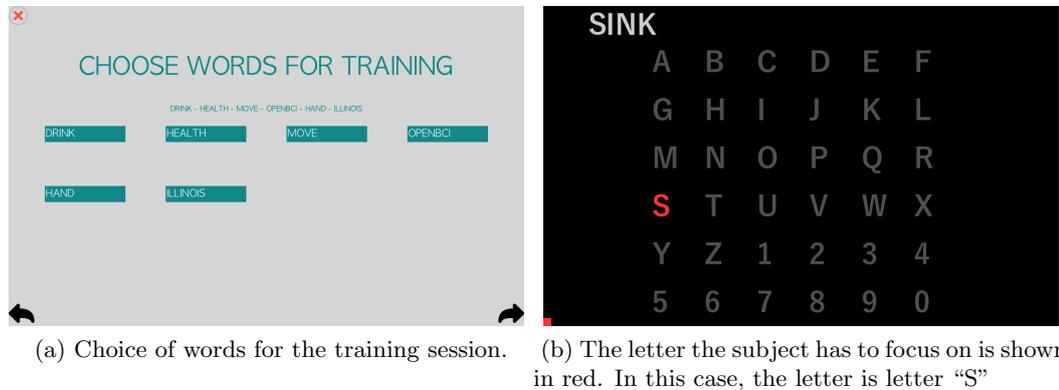


Figure 25: The training setup in the speller mode.

of the performance, a cross validation scheme using a number of folds equal to 4 is employed. The resulting performance on the 4 test sets is used to compute the fitness measure. In order to perform the training of the logistic classifier, a training set has to be built. For our BCI system, the training was done using the speller mode. The subject, during the training phase, is provided with a series of words to spell. Before the stimulation starts, the letter or number the subject has to focus on is highlighted in red. This process is shown in Figure 25. During the training phase, together with an EDF file [58] containing the EEG data, a text file with information about stimulation is saved. This file is necessary to determine, during the training of the classifier, which epochs have to be labeled as target and which ones have to be labeled as non target.

The performance measure is not considered on the single epoch classification, but rather on the number of letters correctly predicted, with a bonus for those letters that can be predicted

with a number of repetitions that is less than the maximum. If we call  $l$  the number of correctly predicted letters out of the total number of letters  $n$ ,  $I$  the set of correctly predicted letters.  $R$  the maximum number of repetitions for the session, and  $r_i, i = 1, \dots, n$ , the number of repetitions needed to correctly predict letter  $i$ , the fitness measure is defined as:

$$f = \frac{1}{n} \left( l + \frac{1}{l} \sum_{i \in I} \frac{R - r_i}{R + 1} \right) \quad (2.12)$$

The second term of the parentheses represents the bonus term. Considering only the set of correctly predicted letters, it computes an index that is inversely proportional to  $r_i$ . This index is always strictly less than 1, so its contribution to the measure of fitness is lower than that of the one obtained with an additional letter correctly predicted. For this reason, having a higher number of correctly predicted letters is always better than having a lower number of repetitions necessary to perform a correct prediction.

$r_i$  is considered as the number of repetitions such that, if a letter is correctly predicted after  $r_i$  repetitions, then it has to be predicted in a correct way also for the following repetitions, i.e. for  $r_i + 1, \dots, R$ . repetitions. For example, if a letter is correctly predicted after only 2 repetitions, wrongly after 5, correctly when using 6 or more repetitions, then the value of  $r_i$  would be 6 and not 2.

### **Selection**

Once the fitness of each chromosome is known, it is used to determine which chromosomes should proceed to the next generation. The employed selection mechanism is the tournament selection with elitism. In tournament selection, represented in Figure 26, each new chromo-

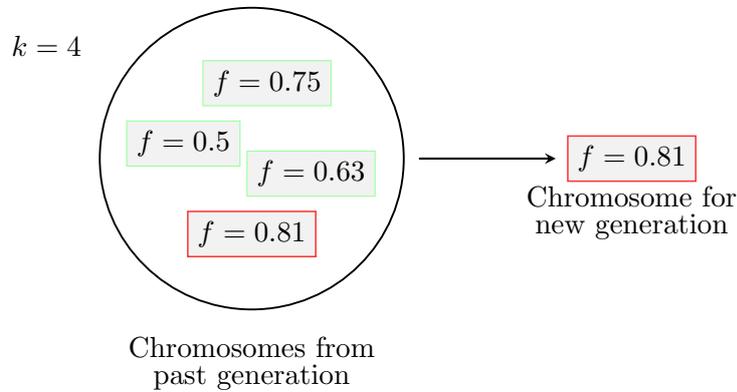


Figure 26: Process of tournament selection employed in the genetic algorithm.

some is chosen by setting up a competition between chromosomes of the old generation: in this competition, the winner is the chromosome with the highest fitness. The number of individuals considered for the tournament is typically small, in order not to favor the fittest individuals. In this implementation, we choose 4 as the number of individuals participating in the tournament. Elitism is another common practice in genetic algorithms. When employed, the fittest chromosome or chromosomes are kept in the new generation even if selection discarded them, or if crossover or mutation modified them. In our case, the number of chromosomes considered for elitism is equal to 1.

### Crossover

Once selection is performed, the selected individuals undergo the crossover operation. This operator allows to create new individuals starting from a pair of parents chromosomes. For each chromosome, crossover is applied with a probability of 0.7. Two chromosomes are divided

in a random manner at a gene boundary, and then the resulting four sections are recombined together, as shown in Figure 21. As shown in the figure, one section of a chromosome can be coupled to one of the two sections of the other chromosome, and the way to do this is decided randomly each time. Since the crossover operation can be applied to chromosomes which share a common ancestor, in the children chromosomes duplicate genes may be present. These duplicate genes are not considered for fitness measure.

### **Mutation**

After selection and crossover, mutation operator, shown in Figure 22, is applied on the chromosomes. As already explained, mutation operator works on gene elements, rather than on chromosomes. For each element of each gene, mutation is applied with a probability of 0.005. For discrete elements in the gene (function for feature extraction, channel and boolean flag), another admissible value is chosen. For continuous elements (the parameters for the feature extraction function), a perturbation is added with a Gaussian distribution. Since it may happen that the new value is outside of the admissible range of  $[0, 1)$ , the new value is wrapped around.

### **Population Size**

The number of chromosomes constituting the population is constant throughout the evolution of the genetic algorithm. For our implementation of the genetic algorithm, the population is composed of 100 chromosomes. The initial population is randomly initialized. In particular, a geometric distribution ( $\mu = 20$ ) is used to determine the number of features (i.e., genes) for each chromosome of the population. The values of each gene element are chosen from uniform distribution over the whole range of admissible values for that element.

### Stop Criterion

The criterion used to stop the evolution of the algorithm is the maximum number of generations. We used a number of generations equal to 15, decreasing it to 12 for the runs that were very high time-consuming. A check on the growth of the fitness value is done after each run, so that the evolution can be stopped if no growth of the fitness is present.

### Feature Interpretation

Due to the fitness measure, which depends on the performance of the logistic classifier, feature extraction and training of the logistic classifier are tightly linked. Therefore, a deeper interpretation of the selected features is possible.

A single feature  $x_j$  is extracted as:

$$x_j = \sum_{t=1}^T u_j(t)s(t) \quad (2.13)$$

where  $j$  stands for the feature encoded in the  $j - th$  gene of the chromosome.  $T$  is the number of samples of the epoch.  $u_j$  is the specific weight function encoded by the parameters of the  $j - th$  gene.  $s(\cdot)$  is the EEG epoch extracted from a single channel.

A logistic classifier estimates the probability of  $s(\cdot)$  belonging to the target class with the following formula:

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + e^{w_0 + \sum_{j=1}^n w_j x_j}} \quad (2.14)$$

The exponent in the denominator of Equation 2.14 can be rearranged as:

$$\begin{aligned}
w_0 + \sum_{j=1}^n w_j x_j &= w_0 + \sum_{j=1}^n w_j \left( \sum_{t=1}^T u_j(t) s(t) \right) \\
&= w_0 + \sum_{t=1}^T \left( \sum_{j=1}^n w_j u_j(t) \right) s(t)
\end{aligned} \tag{2.15}$$

Vector  $v(t)$  is defined as:

$$v(t) = \sum_{j=1}^n w_j u_j(t) \tag{2.16}$$

As shown in Equation 2.16 The vector  $v(t)$  depends only on the features set and on the classifier.

Since  $\mathbf{u}(\cdot)$  and  $\mathbf{w}$  don't depend on the value of the signal itself, vector  $v(t)$  is the same for all

the considered epochs. Equation 2.14 can be rewritten as:

$$\begin{aligned}
P(y = +1 | s(\cdot)) &= \frac{1}{1 + e^{w_0 + \sum_{t=1}^T v(t) s(t)}} \\
&= \frac{1}{1 + e^{w_0 + \langle \mathbf{v}, \mathbf{s} \rangle}}
\end{aligned} \tag{2.17}$$

If all channels are considered, the precedent equations can be readjusted as follows:

$$x_j = \sum_{t=1}^T u_j(t) s_{c(j)}(t) \tag{2.18}$$

where  $C$  is the number of channels,  $s_c(\cdot)$  is the EEG signal recorded on channel  $c$ ,  $c_{(j)}$  is the channel from which the  $j$ -th has to be extracted from. Equation 2.15 becomes:

$$\begin{aligned} w_0 + \sum_{j=1}^n w_j x_j &= w_0 + \sum_{j=1}^n w_j \left( \sum_{t=1}^T u_j(t) s_{c_{(j)}}(t) \right) \\ &= w_0 + \sum_{t=1}^T \left( \sum_{j=1}^n w_j u_j(t) s_{c_{(j)}}(t) \right) \end{aligned} \quad (2.19)$$

Then, we group together features related to the same channels, and Equation 2.19 becomes:

$$\begin{aligned} w_0 + \sum_{t=1}^T \left( \sum_{j=1}^n w_j u_j(t) s_{c_{(j)}}(t) \right) \\ &= w_0 + \sum_{t=1}^T \left( \sum_{c=1}^C \sum_{j:c(j)=c} w_j u_j(t) s_c(t) \right) \\ &= w_0 + \sum_{c=1}^C \sum_{t=1}^T \left( \sum_{j:c(j)=c} w_j u_j(t) \right) s_c(t) \end{aligned} \quad (2.20)$$

The term

$$v_c(t) = \sum_{j:c(j)=c} w_j u_j(t) \quad (2.21)$$

is the same for all the epochs, and depends only on the considered channel.

Considering all the above equations, the final formula for computing the probability of the signal  $s(\cdot)$  pertaining to class  $y = 1$  is:

$$P(y = 1 | \mathbf{s}_1, \dots, \mathbf{s}_c) = \frac{1}{1 + e^{w_0 + \sum_{c=1}^C \langle \mathbf{v}_c, \mathbf{s}_c \rangle}} \quad (2.22)$$

Equation 2.22 allows to compute the probability for the target class by using directly the signals  $\mathbf{s}_1, \dots, \mathbf{s}_c$ . The dot product results in a correlation between vector  $\mathbf{v}_c$  and the signal. Therefore this classification is based on the similarity between the epoch signals and channel templates, expressed by the vectors  $\mathbf{v}_c$ .

### 2.3.3 Online Classification

Now that we have explained the topics of classification with logistic regression and of automatic feature extraction performed with a genetic algorithm, in this section we cover the topic of online classification. Online classification works in the same way for our three possible BCI applications (speller mode, cartesian control of the robotic arm, high level control of the robotic arm), with the only difference that in the high level control of the robotic arm single cells flash, rather than columns and rows,

In the software interface, the number of repetitions and stimulation parameters (presence of symbols in the case of the speller matrix, duration of column/row intensification, inter stimulus interval, ...) are set and impedance of the electrodes is checked prior to the beginning of visual stimulation.

At the conclusion of the visual stimulation for each trial, the classification process begins. Using the photoresistor signal, the EEG signal is segmented into epochs.

After the epochs are extracted, they are filtered with a sixth-order Butterworth bandpass filter with cut off frequencies of 0.5 and 30 Hz, and they're ready to be classified. We implemented two different classification modalities, to compare the results obtained with them, and we will refer to them as *probability averaging* and *epochs averaging*.

## Probability Averaging Method

In the *probability averaging* method, data are averaged in the classification score space. For each extracted epoch, the probability of the presence of a P300 is computed with the Equation 2.22. Since information about which stimuli are presented is saved during the stimulation process, it is possible to determine which row and column elicited a P300 response with the highest probability.

If we consider row  $i$  of the matrix, for each repetition of the stimulation, row  $i$  flashes one time. For each repetition, the probability of the presence of the P300 in the epoch extracted around the time at which row  $i$  flashed is calculated. The average probability is computed as:

$$\bar{P}_{i,P300} = \frac{\sum_{j=1}^R P_{i,j}}{R} \quad (2.23)$$

where  $P_{i,j}$  is the probability of the P300 response being present in the epoch associated to row  $i$  at the  $j$ -th repetition, and  $R$  IS THE .

This process is done for all the rows and for all the columns. So, at the conclusion of the probability computation, we will have the following vectors:

$$\bar{P}_{rows} = \left[ \bar{P}_{r,1} \quad \bar{P}_{r,2} \quad \bar{P}_{r,3} \quad \bar{P}_{r,\dots} \quad \bar{P}_{r,rows_{MAX}} \right]$$

$$\bar{P}_{cols} = \left[ \bar{P}_{c,1} \quad \bar{P}_{c,2} \quad \bar{P}_{c,3} \quad \bar{P}_{c,\dots} \quad \bar{P}_{c,cols_{MAX}} \right]$$

The target symbol is found by determining the intersection between the row and the column for which the average value of probability is maximized:

$$\begin{aligned}\text{row}_{P300} &= \arg \max_i \overline{P_{r,i}} \\ \text{column}_{P300} &= \arg \max_i \overline{P_{c,i}}\end{aligned}$$

This classification process is the same for speller and cartesian robotic arm control. The only difference between these two modalities is the dimension of the stimulation matrix. For the high level control of the robotic arm the cell with the highest probability is directly determined.

### **Epochs averaging method**

In the *epochs averaging* method for the classification, EEG epochs are averaged in the data space. The basic difference in respect to the *probability averaging* method is that, instead of computing the average probability, we compute the average epoch associated to the flashing of each row and column. Then, we compute the probability of this average signal.

As an example, the average EEG epoch associated to row  $i$  of the matrix is:

$$\overline{s}_i = \frac{\sum_{j=1}^R s_{i,j}}{R} \quad (2.24)$$

where  $s_{i,j}$  is the epoch associated to the flashing of row  $i$  at repetition  $j$ -th, and  $R$  repetitions is the maximum number of repetitions. After computing the averages for all rows and columns, the following vectors are defined:

$$\overline{s}_{rows} = \left[ \overline{s}_{r,1} \quad \overline{s}_{r,2} \quad \overline{s}_{r,\dots} \quad \overline{s}_{r,rows_{MAX}} \right] \quad \overline{s}_{cols} = \left[ \overline{s}_{c,1} \quad \overline{s}_{c,2} \quad \overline{s}_{c,\dots} \quad \overline{s}_{c,cols_{MAX}} \right]$$

Then, the probability associated to each of these average signals is computed, and the following vectors are obtained:

$$P_{rows} = \begin{bmatrix} P_{r,1} & P_{r,2} & P_{r,3} & P_{r,\dots} & P_{r,rows_{MAX}} \end{bmatrix}$$

$$P_{cols} = \begin{bmatrix} P_{c,1} & P_{c,2} & P_{c,3} & P_{c,\dots} & P_{c,cols_{MAX}} \end{bmatrix}$$

Once the probability values are known for each average epoch, the target symbol is determined by computing the row and column for which probability is the highest. Similar the *probability averaging* method, we have that:

$$\text{row}_{P300} = \arg \max_i P_{r,i}$$

$$\text{column}_{P300} = \arg \max_i P_{c,i}$$

Similar to *epochs averaging* method, in the high level control of the robotic arm, an individual cell, instead of a column and row, is identified as the one which generated the P300 response.

### Software Summary

In Figure 27, we show a graphic summary of the software part of the system, including the classification process. In this block diagram, all the different phases and modalities are shown.

The process of filtering, segmentation and classification during live use requires 100 - 300 ms to be performed, depending on the number of repetitions used.

TABLE III: SUBJECTS PARTICIPATING IN THE STUDY.

Subject ID	Age	BCI Experience
S1	23	No
S2	20	No

### 2.3.4 Experimental Protocol

Two subjects (Table III) participated in evaluating the device. None of the subjects had prior experience with EEG recording or BCI. Each subjects was required to participate to three EEG recording sessions. In session one P300 elicitation was valuated and data to train the classifier were recorded. In the second session experimental data for classifier validation was collected. In the last session, tests on online use of the BCI system in speller and robotic arm mode were carried out.

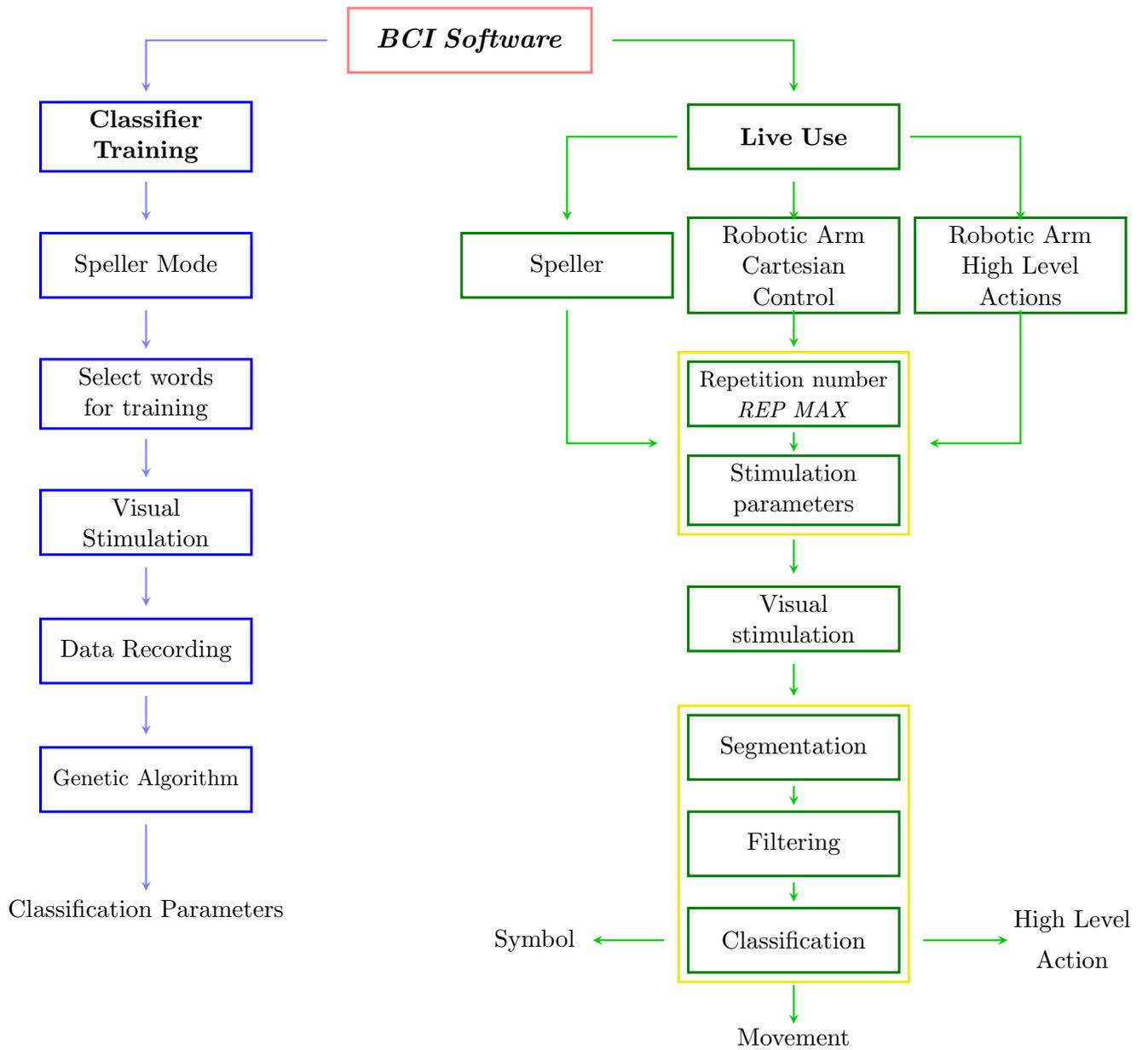


Figure 27: The software component of the assistive device.

## CHAPTER 3

### RESULTS

In this chapter, we will discuss the results obtained from two subjects participating in evaluating the device. Firstly, initial experiments on P300 elicitation will be described. Then, data related to the genetic algorithm and to the classifier validation will be provided. Lastly, online performance of the device will be evaluated.

#### 3.1 Initial Evaluation Phase

In the first stage of the study, our goal was to evaluate the presence of P300 response for each individual subject using the standard settings of the device.

The subjects were asked to focus on a specific letter of the speller grid shown on the monitor. EEG data were recorded, synchronized offline with the visual stimulation, and epochs were extracted. The epochs were then filtered with a using a sixth order bandpass Butterworth filter with cutoff frequencies of 0.5 and 30 Hz, and labeled as target or non-target according to the associated stimulus. If the associated stimulus was the column or the row containing the letter the subjects had to focus on, the epoch would have been labeled as target, otherwise it would have been labeled as non-target. The averages of the target and non-target epochs for the two subjects were computed considering all the 8 EEG channels. The results are shown in Figure 28.

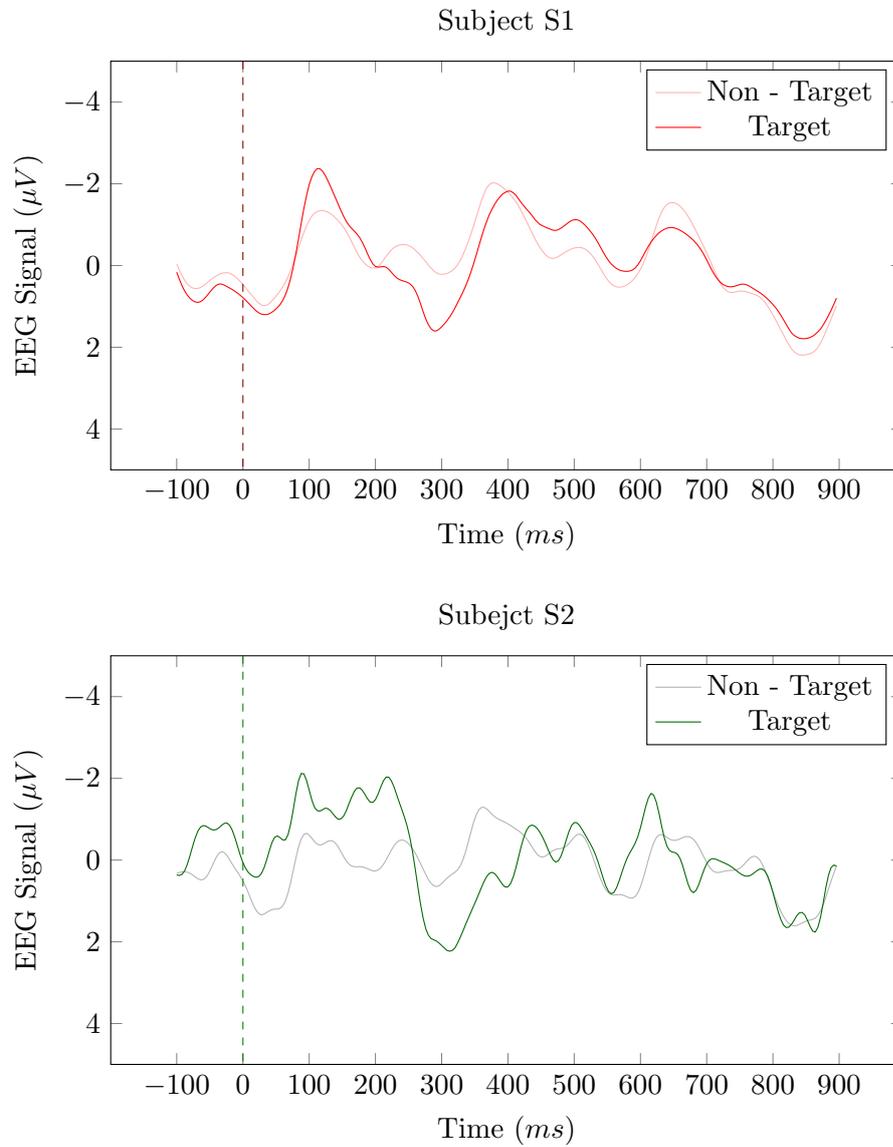


Figure 28: Average target and non-target epochs for the two subjects during the initial evaluation of the device. In the average target epochs for the two subjects, it is possible to see a positive peak at 300 ms.

TABLE IV: DESCRIPTION OF THE TRAINING SET OF THE TWO SUBJECTS.

Subject	Letters	Epochs	Hours	Repetitions
S1	555	66600	5	10
S2	127	7620	1	5

### 3.2 Classifier Training Phase

Upon completing the initial phase, the subjects participated in a training session. As we already shown in Figure 25, during the training session the subjects were asked to spell a series of words. Before each trial, the letter the subject had to focus on was highlighted in red for 2 seconds. During each trial, flashing of the row or the column that contained the letter of interest elicited a P300 response . The spelling process lasted until the sequence of chosen letters was completed. Each epoch was then be labeled as target, if extracted around a target stimuli, or non-target.

The training set of target and non-target epochs was then used to perform automatic feature extraction and training of the logistic classifier. Both of these tasks were accomplished using the genetic algorithm described in Section 2.3.2.1. Information about the training set used for each subject is shown in Table IV. Subject S2 preferred 5 repetitions instead of 10, finding it easier to focus on shorter length sessions.

In Figure 29 the average of targets and non-targets epochs obtained from the training set for each subject.

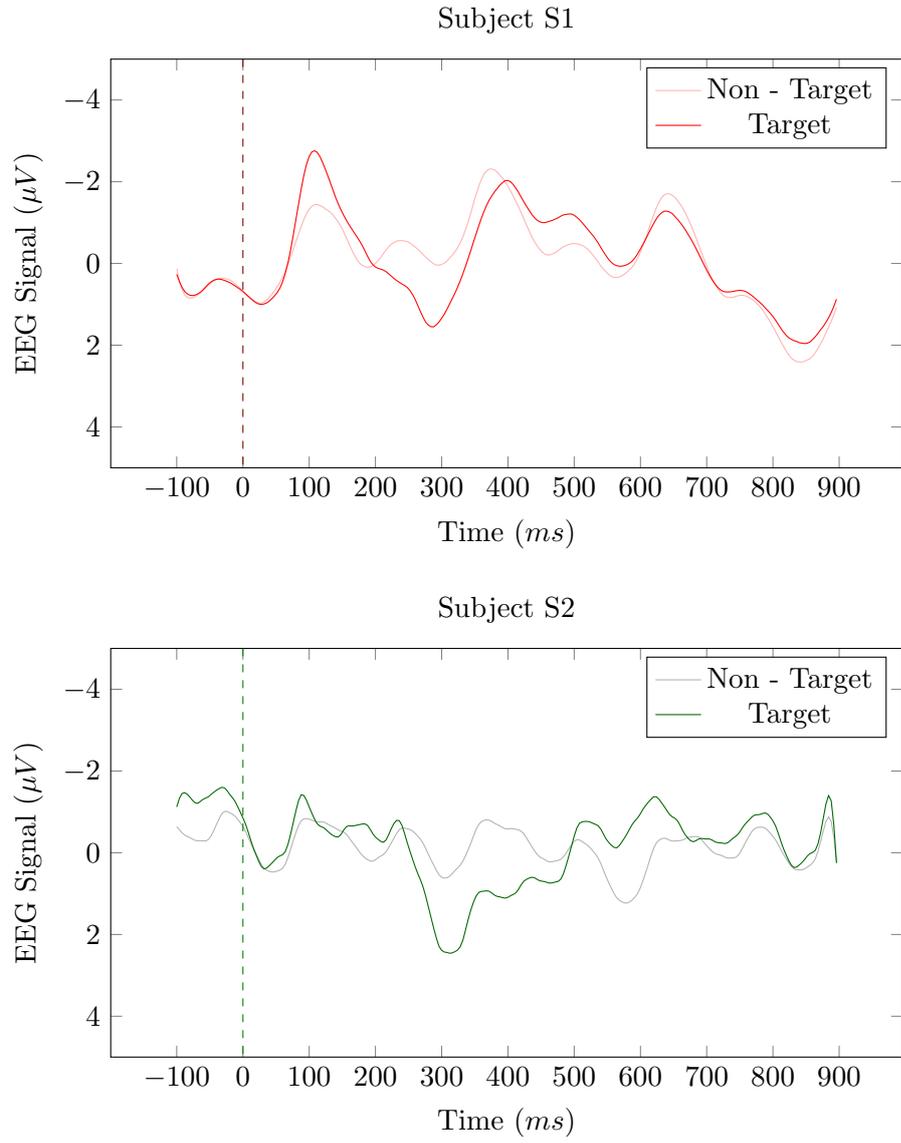


Figure 29: Average target and non-target epochs of the training set.

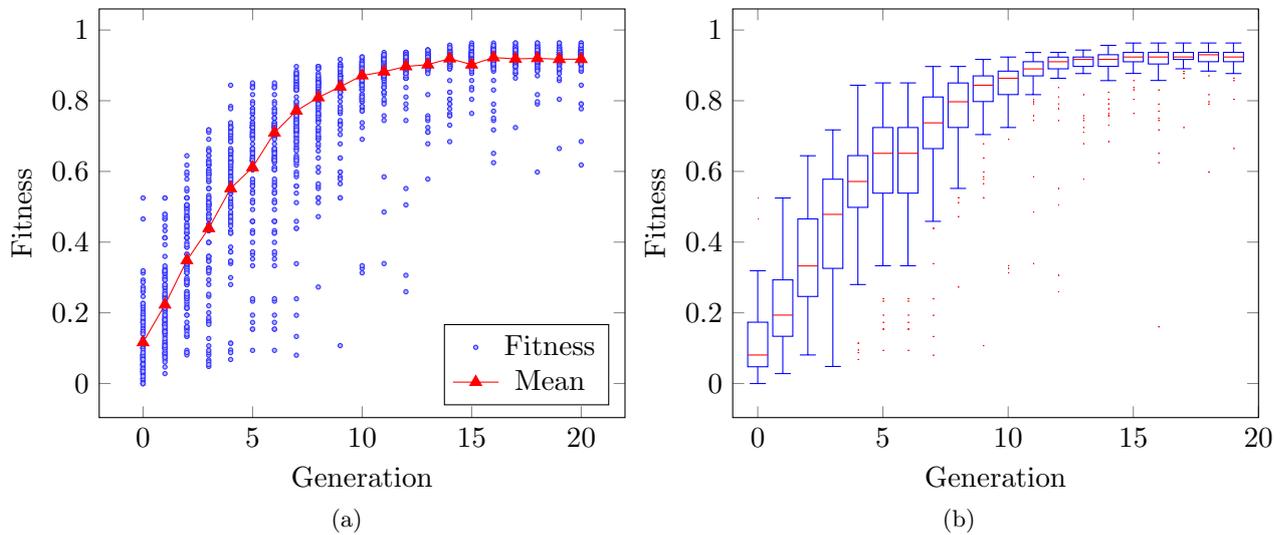


Figure 30: *Left*: Fitness value of each of the 100 chromosomes at each generation is shown in blue. Mean fitness value is shown in red. *Right*: Box plots of fitness values as a function of genetic algorithm generation. [Data obtained with training set of subject S2].

Fitness values of the population of 100 chromosomes across the generations of the genetic algorithm, computed using Equation 2.12, are shown in Figure 30a. The value of fitness for each chromosome is reported, together with the mean value of fitness for each generation. Fitness values, as expected, increase as the evolution proceeds. As selection, crossover, and mutation modify the population of chromosomes, the set of features encoded by the chromosomes improve. Therefore, the average performance of the logistic classifier on the 4 fold cross validation scheme increases.

In Figure 30b, the same information about fitness evolution is represented through box plots. As expected, the mean fitness values converges to high values, close to 1, and the variation in

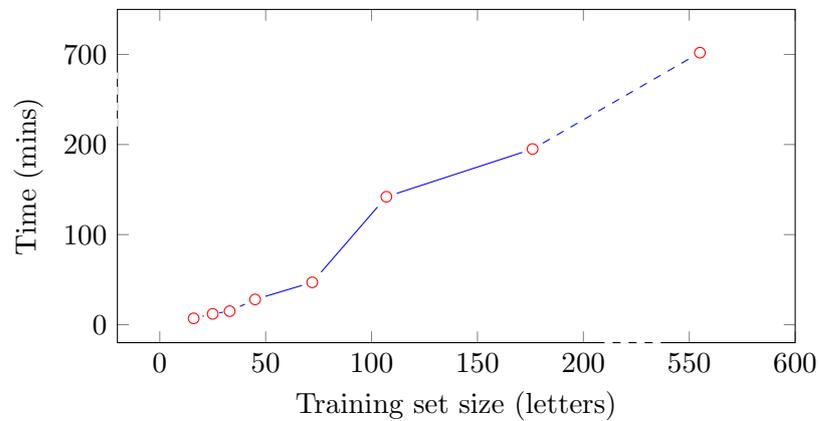


Figure 31: Time required to complete a run of the genetic algorithm as a function of the size of the training set. This set was performed on a 64 bit laptop.

fitness across chromosomes reduces. A plateau in the fitness values is typically reached after approximately 12 generations. Therefore, we limited the maximum number of generations to 15. For the biggest training sets, we limited it to 12.

Figure 31 represents the time required to complete a run of the genetic algorithm as a function of the number letters spelled by the subjects during the training session. In the presented data, the runs were completed after 12 generations of the GA. As expected, larger sizes of the training set requires greater amount of time. For the largest training set, composed of 555 letters, the GA took 11 hours to complete the evolution process on a 64 bit laptop.

The training procedure needs to be performed the very first time the subject uses the BCI system. Upon successfully completing the training, the subject would be able to use the BCI live.

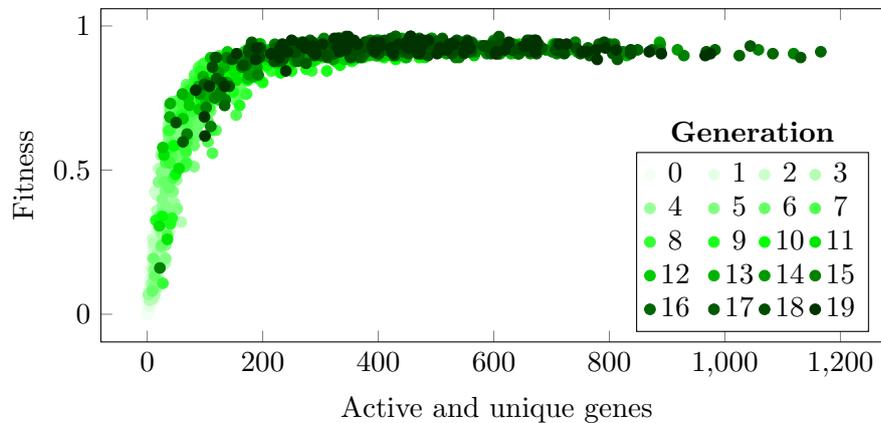


Figure 32: Fitness values as a function of number of features and of genetic algorithm generation. [Data were obtained with subject S2 training set].

In Figure 32 we evaluate the fitness values and the number of features encoded by only active and unique genes across generations. Greater number of features encoded by the chromosome results in a higher fitness value. The plot reaches a plateau at  $\sim 200$  features, close to generation 12. As shown in the Figure, even if chromosomes increase in size, and encode more features for classification, there is minimal to no improvement in the fitness value after generation 12<sup>th</sup>.

### **3.3 Classifier Validation Phase**

After performing automatic feature extraction and training the logistic regression classifier, the obtained classifier was validated on a validation data set never used before by the genetic algorithm. For both subjects, the validation set was composed of data recorded on various days. It is important to evaluate the classifier using data recorded on various sessions, to assure performance independence from external variations that could be present in recording sessions.

TABLE V: DESCRIPTION OF THE VALIDATION SET OF THE TWO SUBJECTS.

Subject ID	Sessions	Letters)	Epochs	Maximum (Repetitions)
Subject S1	6	62	7440	10
Subject S2	4	57	3420	5

Later in the chapter, we will refer to a measure of accuracy of our BCI device. Accuracy is computed in terms of correctly predicted symbols (letters, robotic arm movements, high level actions). If  $l$  represents the number of correctly predicted symbols in a session, and  $n$  the total number of symbols spelled, accuracy is computed as:

$$ACC = \frac{l}{n} \quad (3.1)$$

Three analysis were performed on the validation set. First analysis, accuracy as a function of the number of repetitions. Second analysis, accuracy as a function of the size of the training set. Third analysis, inter - subjects dependence.

### 3.3.1 Accuracy vs Repetitions

In this section we evaluate accuracy changes as a function of the number of repetitions. This evaluation is done on both classification methods: *probability averaging* and *epochs averaging* method.

In Figure 33, accuracy measured with *probability averaging* and the *epochs averaging* methods as a function of the number of repetitions is represented.

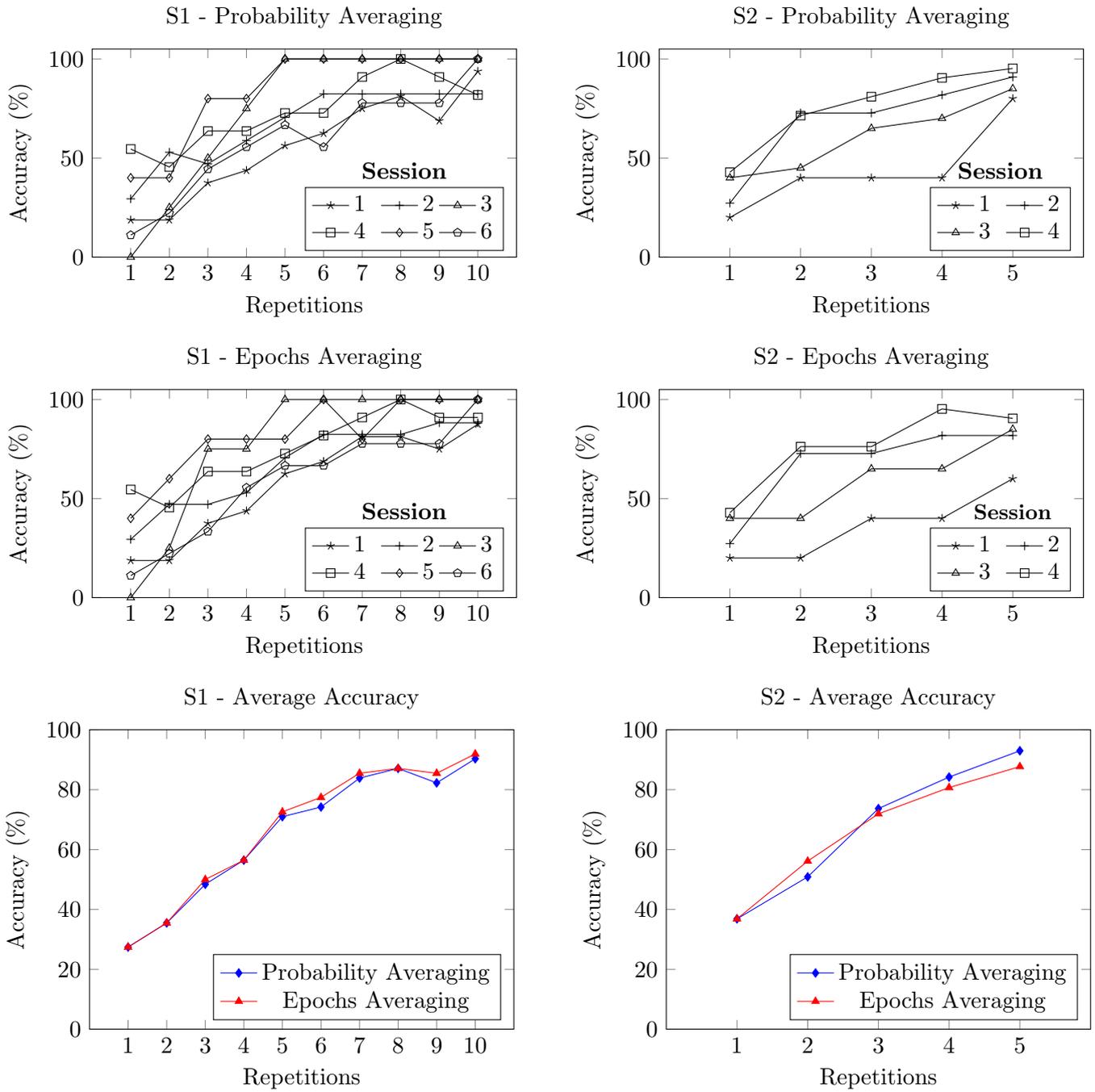


Figure 33: Average accuracy as a function of the number of repetitions for the validation set. In the first two rows, individual sessions are considered. In the last row, average accuracy is reported.

As expected, for both classification methods accuracy increases when the number of repetitions is increased. Regarding the two different classification methods, *probability averaging* and *epochs averaging*, they both perform well. Throughout our experiments, we noticed that for a noisy signal, the *epochs averaging* method performs better than the *probability averaging* method. Since, by averaging the epochs, the amount of noise present in the signal is reduced. This is not true for the probability averaging method, since the averaging is performed on the classification score space and not on the data space.

For subject S1, accuracy greater than 80% is reached when using 7 repetitions (83.87% with probability averaging method, 85.48% with epochs averaging method). For subject S2, accuracy greater than 80% is reached when using 4 repetitions (84.21% with probability averaging method, 80.7% with epochs averaging method).

### **3.3.2 Accuracy vs Training Set Size**

Another analysis we performed was related to how accuracy changes as a function of the size of the training set. The training set, as already explained, is used to perform automatic feature extraction and training of the logistic regression classifier. Therefore, an increase in the accuracy is expected when increasing the size of the training set. To perform this analysis, the accuracy on the same test set was computed by gradually increasing the size of the training set. In Figure 34, the result of this analysis is shown. The training set size is shown as a percentage of the total training set size for each subject.

For subject S1, accuracy reaches above 80% around 32% of the total training set size. This corresponds to a training set composed of 180 letters, equal to 21600 epochs when using 10

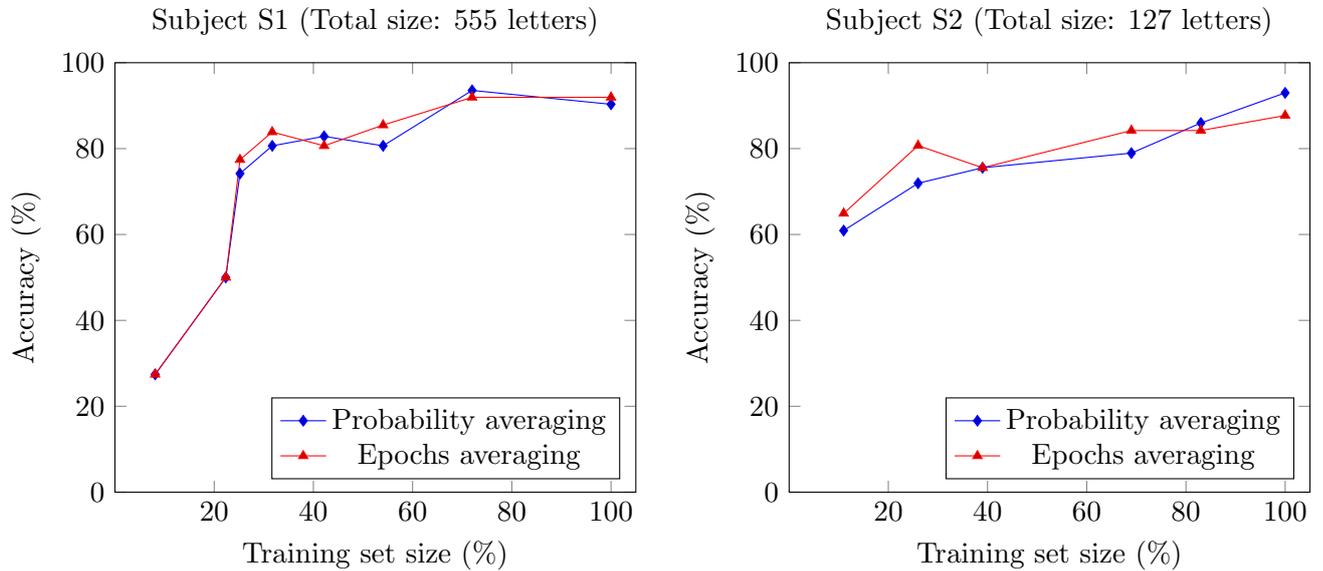


Figure 34: Average accuracy on the validation set as a function of the training set size. The maximum number of repetitions was used to compute the accuracy.

repetitions per trial. For subject S2, accuracy reaches above 80% around 70% of the total training set size. This corresponds to a training set composed of 89 letters, equal to 5340 epochs when using 5 repetitions per trial. As suggested by the plots, depending on the desired accuracy tolerance, the training set size could be decreased, therefore decreasing the training time for the user.

### 3.3.3 Inter-Subjects Dependence

The last analysis that we performed on the validation set is related to inter-subjects dependence. In Figure 35 the average target and non-target epochs of the validation set for subject S1 and S2 are shown. Visually comparing the two signals, a clear difference is visible among

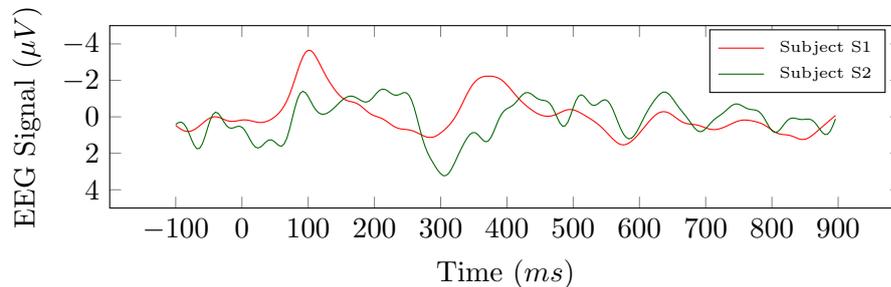


Figure 35: Average target and non-target epochs of the validation set for subject S1 and S2.

the two subjects. Therefore, emphasizing the fact that subject specific features have to be extracted with the genetic algorithm for each subject. To underline again the importance of subject specific training, we performed classification on the validation set of one subject using P300 templates extracted from the training set of the other subject. Results are shown in Table VI. In this table, the value of each cell represents the accuracy on the validation set of one subject (row) when tested with the templates of the other subject (column).

TABLE VI: INTER-SUBJECT DEPENDENCE: ACCURACY WAS COMPUTED USING THE MAXIMUM NUMBER OF REPETITIONS OF THE VALIDATION SET.

(a) Probability Averaging			(b) Epochs Averaging		
	P300 Templates			P300 Templates	
	S1	S2		S1	S2
S1	90.32	1.61	S1	91.35	1.61
S2	12.28	92.98	S2	10.35	87.71

TABLE VII: NUMBER OF CHOICES N AND MAXIMUM NUMBER OF FLASHES FOR THE THREE POSSIBLE MODES OF USE.

Mode of use	N	MAX FLASH
Speller	36	12
Cartesian robotic arm control	16	8
High level robotic arm control	5	5

As expected, classification performance is low when not using the subject specific templates, emphasizing that the fact that even though P300 responses are innate response of the brain, they are different among individuals. Therefore, each subject has to participate in a training session, otherwise it would be impossible to obtain good performance.

### 3.4 Online Classification

After performing classifier training and validation for each subject, we tested the online classification accuracy for the three modes of use: speller, cartesian control of the robotic arm, and high level control of the robotic arm.

Other than accuracy, that is computed using Equation 3.1, a common measure of performance in the BCI field is the bit rate. The bit rate can be expressed also in terms of *bits/min* [7], and is commonly called Information Transfer Rate (ITR). ITR contains information about communication speed and accuracy.

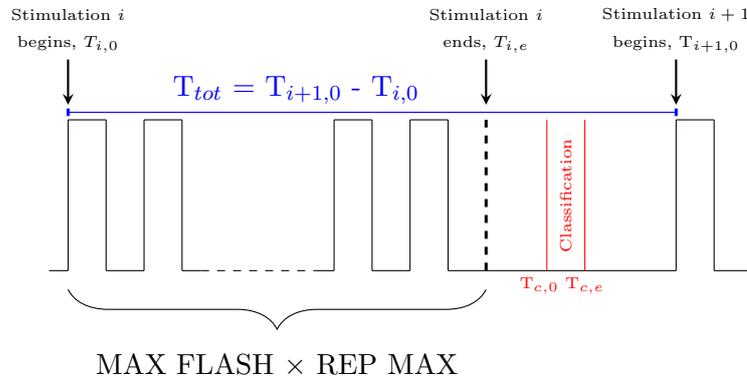


Figure 36: Protocol used to compute  $T_{tot}$ .

The formula used to compute bit rate is:

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2 \left( \frac{1 - P}{N - 1} \right) \quad (3.2)$$

where  $N$  represents the possible choices per trial (values of  $N$  for the different modes are shown in Table VII), and  $P$  represents the probability that the choice on which the subject is focusing on is selected. Therefore,  $P$  represents the accuracy of the system. The assumed hypothesis is that each choice has the same probability of being selected.

ITR can be computed as follow:

$$\begin{aligned} ITR &= B \cdot \frac{N}{min} = B \cdot \frac{60}{T_{tot}} \\ &= \log_2 N + P \log_2 P + (1 - P) \log_2 \left( \frac{1 - P}{N - 1} \right) \cdot \frac{60}{T_{tot}} \end{aligned} \quad (3.3)$$

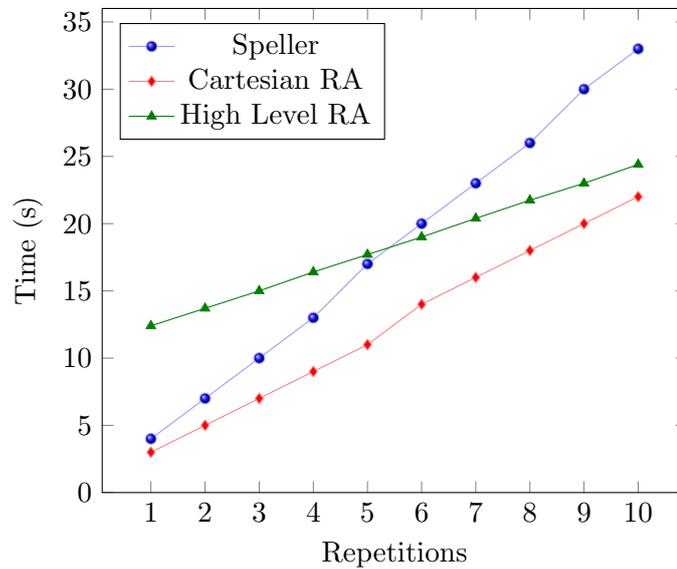


Figure 37: Time to complete a trial,  $T_{tot}$ , as a function of the number of repetitions for three BCI modes of use.

To compute the *bits/min* index, it is necessary to determine the total number of trials per minute,  $N/min$ . Therefore, we computed the time required by the software to complete a trial,  $T_{tot}$ . A graphical representation on how the computation of this value of time is computed is shown in Figure 36. The measurement of time starts from the beginning of the stimulation, and ends until the following stimulation starts.

The results of the time computation are shown in Figure 37.

### 3.4.1 Speller

In order to compute the online classification accuracy in the speller mode, the subjects informed us on the letters they were focusing on during the online test session.

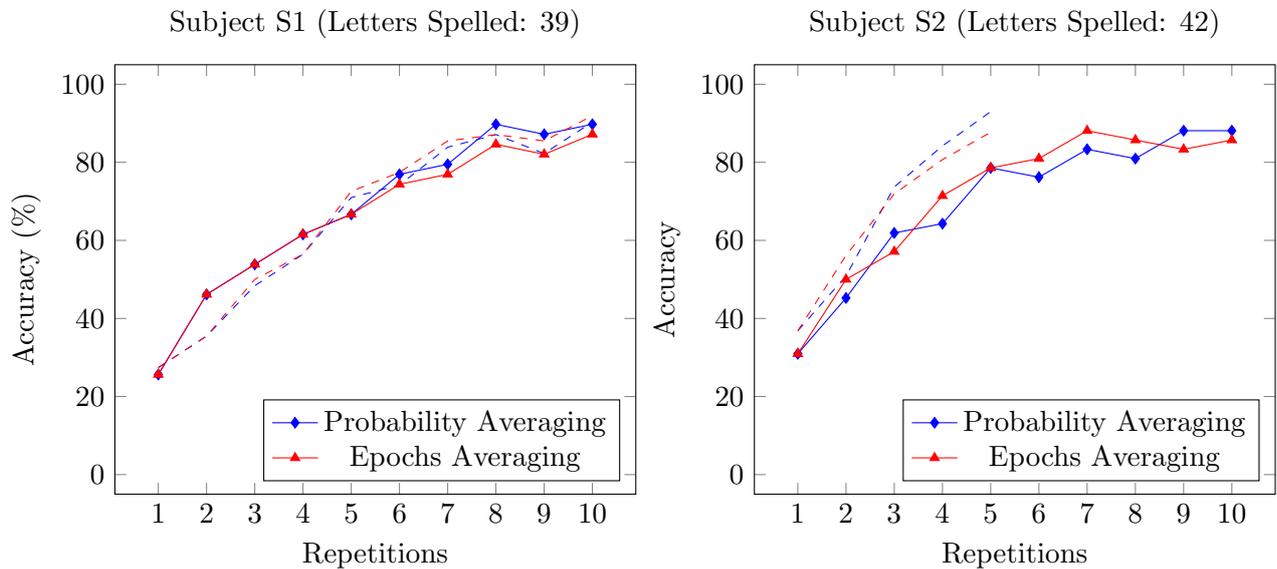


Figure 38: Online classification accuracy as a function of the number of repetitions for the speller mode. Accuracy on the validation set is shown with dashed lines.

Results of the online classification for both subjects are shown in Figure 38. For subject S1 online performance is comparable to the validation set results. For subject S2, up until five repetitions, online performance is lower than the validation test results.

After computing the online classification accuracy, we computed the performance of the speller mode in terms of *bits/min*. The values of *bits/min* as a function of the number of repetitions are shown in Figure 39. Bit rate takes into consideration both speed and accuracy of the system. Therefore, in order to increase communication speed, a lower number of repetitions is favorable. Versus, high number of repetitions results in higher classification accuracy.

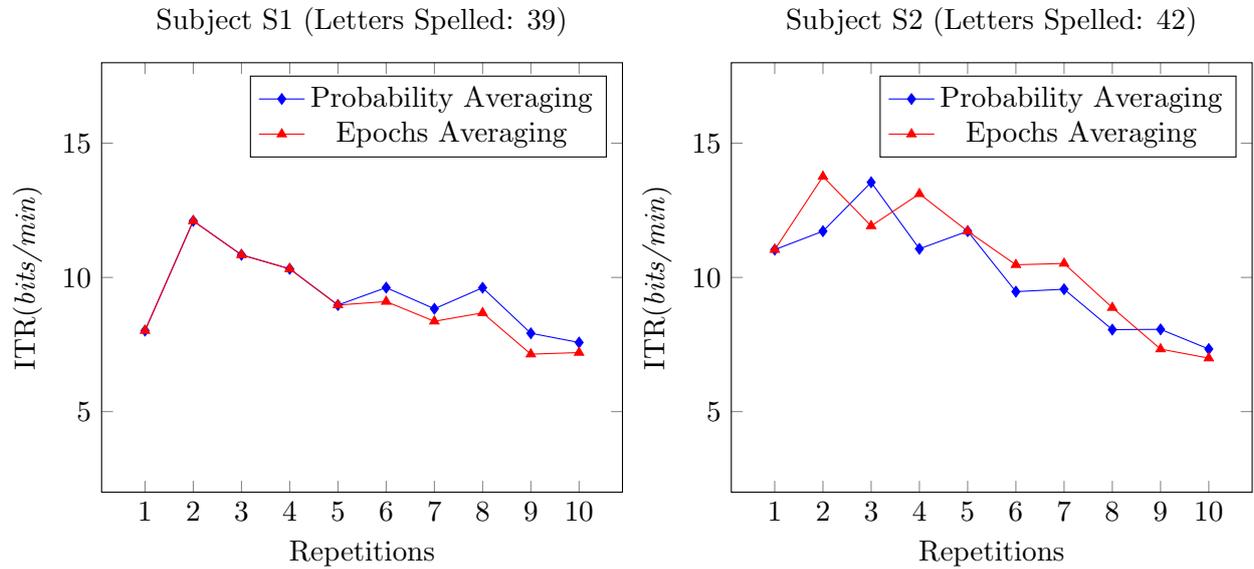


Figure 39: Information transfer rate as a function of the number of repetitions for the speller mode.

Therefore, the optimal number of repetitions should be chosen in order to maximize speed and accuracy.

For subject S1, an accuracy of 79.49% is reached after 7 repetitions using probability averaging method. When using epochs averaging method, 8 repetitions are necessary to achieve an accuracy of 84.62%. The correspondent ITR are 8.83 and 8.68 *bits/min*. The maximum value of ITR for subject S1 is 12.11 *bits/min*, reached using 2 repetitions for both probability averaging and epochs averaging.

For subject S2, when using 7 repetitions and probability averaging method, an accuracy of 83.33% is reached. With epochs averaging method, 6 repetitions are necessary to reach an

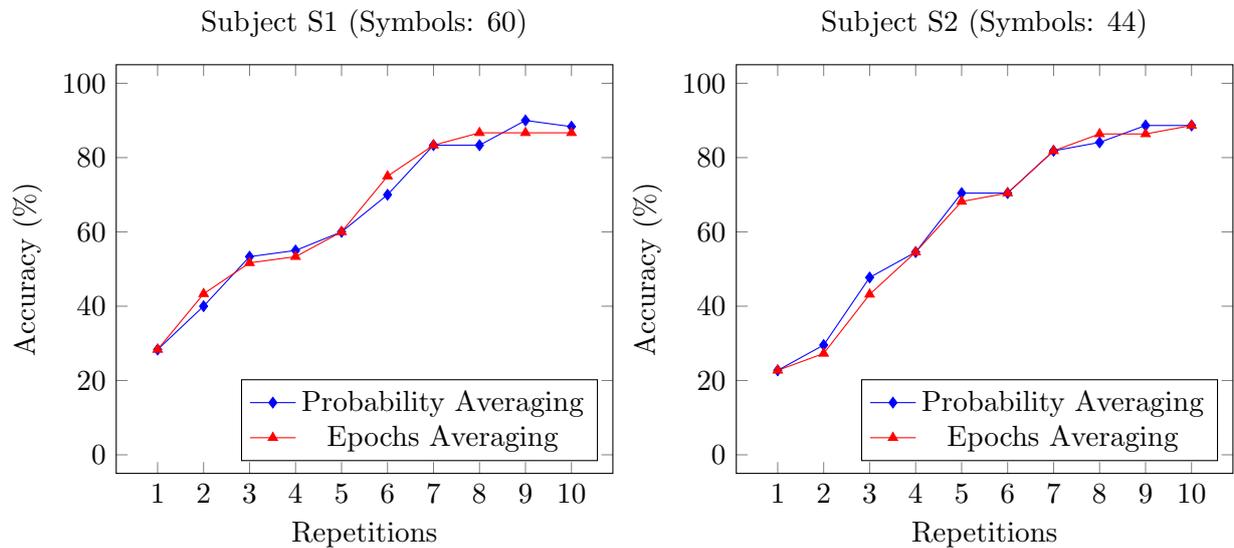


Figure 40: Online classification accuracy as a function of the number of repetitions for the cartesian control of the robotic arm.

accuracy of 80.95%. The correspondent ITR are 9.56 and 10.47 *bits/min*. Maximum values of ITR for subject S2 are 13.54 *bits/min* (3 repetitions, probability averaging) and 13.76 *bits/min* (2 repetitions, epochs averaging).

### 3.4.2 Robotic Arm Cartesian Control

Online classification accuracy for the robotic arm in cartesian control mode was tested. Similar to the speller mode, the subject had to focus on symbols of the stimulation rather than on letters or numbers. The subject would then communicate the target symbol he focused on during the session. As a feedback to the user, the detected symbol is shown in green on the top of the screen. The additional feedback provided to the user is represented by the movement of

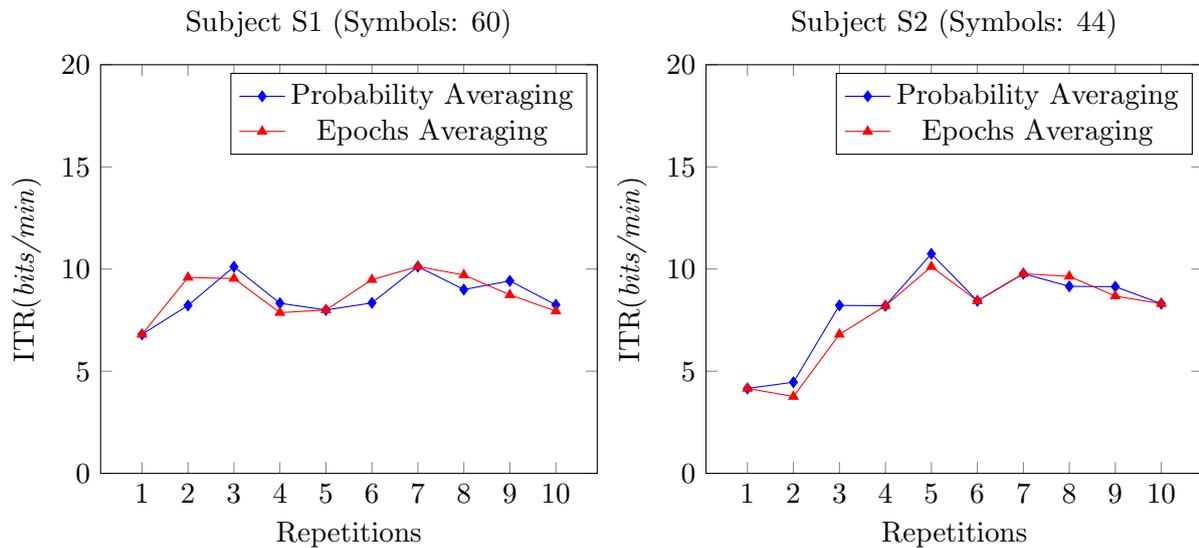


Figure 41: Information transfer rate as a function of the number of repetitions for the cartesian control of the robotic arm.

the robotic arm. Results for the cartesian control of the robotic arm in terms of accuracy and bits/min are shown in Figure 40 and Figure 41, respectively.

For subject S1, 7 repetitions were necessary to obtain an accuracy above 80% (precisely, 83.3%) with both *probability averaging* and *epochs averaging* method. The resulting ITR is 10.12 *bits/min*, that is also the highest value of ITR reached by subject S1. The highest value of accuracy was reached using 9 repetitions and the probability averaging method, leading to an accuracy of 90%.

For subject S2, 7 repetitions were necessary to achieve an accuracy of 81.82%, with both *probability averaging* and *epochs averaging* method. The correspondent ITR is 9.77 *bits/min*. The highest value of ITR was reached using 5 repetitions: 10.75 *bits/min*. This ITR corresponds

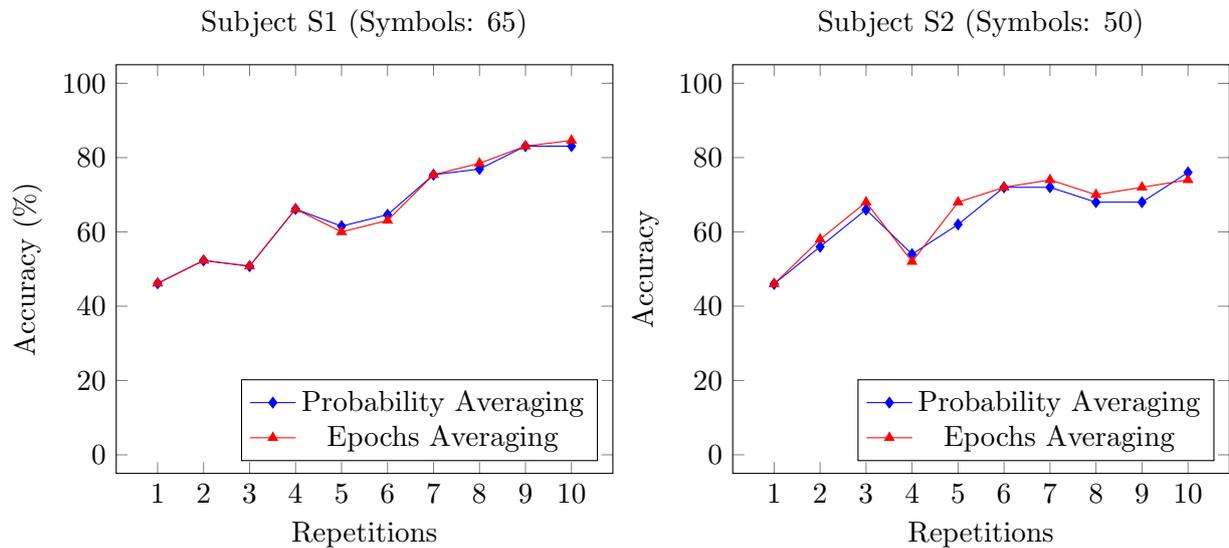


Figure 42: Online classification accuracy as a function of the number of repetitions for the high level control of the robotic arm.

to an accuracy of 75%. The highest value of accuracy, 88.64%, was reached with *probability averaging* method after 9 repetitions, and at 10 repetitions by the *epochs averaging* method.

### 3.4.3 Robotic Arm High Level Control

The final test was performed with the high level control of the robotic arm. To compute online classification accuracy, the subjects would communicate the target symbols they focused on during the trial. We remind in this section that, for the high level control of the robotic arm, in a single repetition one cell flashes one time. For the speller and the cartesian control of the robotic arm, that employ row/column flashing, in a single repetition each cell flashes two times (one when the correspondent row flashes, and one when the correspondent column flashes).

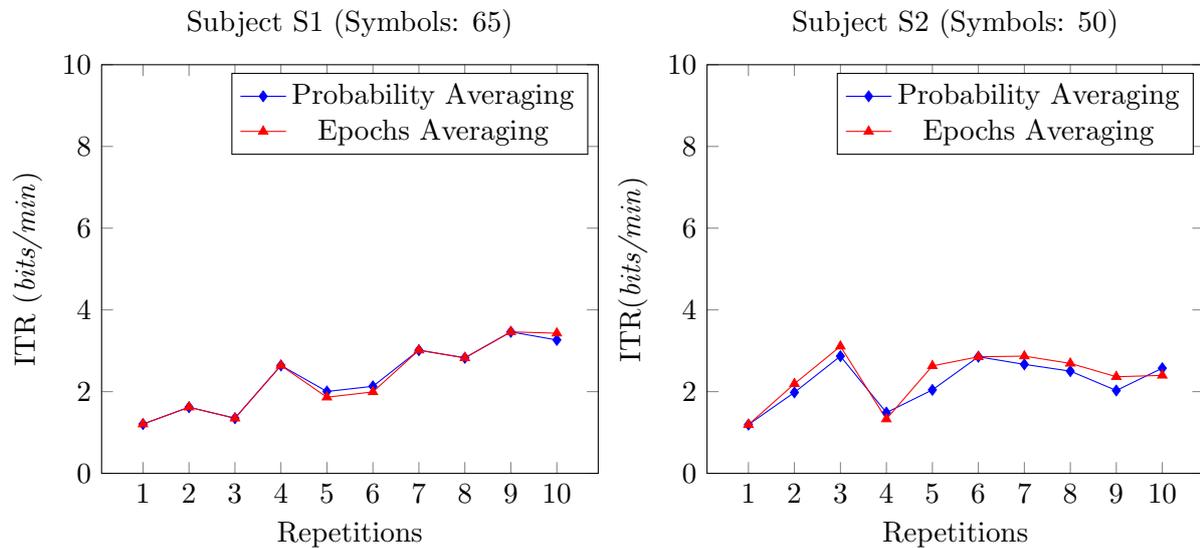


Figure 43: High level control of the robotic arm: information transfer rate as a function of the number of repetitions.

Results for the high level control of the robotic arm in terms of accuracy and ITR are shown in Figure 42 and Figure 43, respectively.

For subject S1, 9 repetitions are necessary to reach an accuracy of 83.08% when using probability averaging or epochs averaging method. The resulting ITR is 3.46 *bits/min*, which represents also the maximum value of ITR achieved by subject S1. The maximum value of accuracy achieved by the subject is 84.62%, reached at 10 repetitions and employing the *epochs averaging* method.

For subject S2, lower accuracy, in comparison to speller and cartesian control of the robotic arm, was obtained. 6 repetitions were necessary to achieve an accuracy of 72%, both with *probability averaging* method and *epochs averaging* method. The correspondent ITR is 2.85 *bits/min*.

Maximum value of accuracy, 76%, was achieved with 10 repetitions and *probability averaging method*. Maximum ITR, 3.11 *bits/min*, was reached with 3 repetitions and *epochs averaging method*.

Values of ITR in the high level control of the robotic arm may seem low in comparison to the ones achieved with both speller and cartesian control of the robotic arm. What has to be taken into consideration is that, during high level control of the robotic arm, no stimulation happens when the robotic arm is moving, since the action is not immediate. Therefore, as shown in Figure 37, time required to complete a trial for the high level control of the robotic arm is higher. As a consequence, the ITR is lower.

### 3.4.4 Average Performance

Average accuracy and ITR amongst the two subjects, for three modes of use (speller, cartesian control of the robotic arm, high level control of the robotic arm), are reported in Table VIII and in Table IX for *probability averaging* and *epochs averaging* method, respectively. The correspondent plots are shown in Figure 44a and Figure 44b. As it is possible to notice from both Tables and Figures, information transfer rate for the robotic arm in high level control is low if compared to the speller and the cartesian control of the robotic arm. This is due to the fact that the stimulation is stopped when the robotic arm is moving, and for this reason  $T_{tot}$ , which is used in Equation 3.3 to compute the ITR, is higher.

Using the *probability averaging* method and considering the maximum number of repetitions, equal to 10, the average accuracy is of 88.92% for the speller mode (7.45 *bits/min*), 88.48% for the robotic arm in cartesian control mode (8.28 *bits/min*), 79.54% for the robotic arm in high level control mode (2.91 *bits/min*). With 5 repetitions, half of the maximum value of repetitions, the average accuracy is reduced to 72.62% for the speller mode (10.35 *bits/min*), 65.23% for the robotic arm in cartesian control mode (9.37 *bits/min*), 61.77% for the robotic arm in high level control mode (2.02 *bits/min*).

If the *epochs averaging* method is employed, for the speller mode the values of accuracy are 86.45% (7.1 *bits/min*, 10 repetitions) and 72.62% (10.35 *bits/min*, 5 repetitions). For the cartesian control of the robotic arm, the results are 87.65% (8.12 *bits/min*, 10 repetitions) and 64.09% (9.06 *bits/min*, 5 repetitions). Last, for the robotic arm in high level control mode,

the obtained values are 79.31% (2.91 *bits/min*, 10 repetitions) and 67.54% (2.42 *bits/min*, 5 repetitions).

For all the modes of use, reducing the number of repetitions is fundamental, because a large number of repetitions could result in a low communication speed. In order to determine the number of repetitions to be used, it is possible to choose the number of repetitions that maximizes either ITR or accuracy, according to the user's preference. A user may want to communicate at high rate, at the expense of a lower accuracy, or to communicate at low rate, but with higher accuracy.

TABLE VIII: AVERAGE VALUES OF ACCURACY AND ITR FOR *PROBABILITY AVERAGING* METHOD.

[R = REPETITIONS, ACC. = ACCURACY]

(a) Speller			(b) Cartesian Control			(c) High Level Control		
R	Acc. (%)	ITR	R	Acc.(%)	ITR	R	Acc.(%)	ITR
1	28.30	9.53	1	25.53	5.48	1	46.08	1.20
2	45.70	11.91	2	34.77	6.34	2	54.15	1.80
3	57.88	12.19	3	50.53	9.17	3	58.38	2.11
4	62.91	10.69	4	54.77	8.26	4	60.08	2.06
5	72.62	10.35	5	65.23	9.37	5	61.77	2.02
6	76.56	9.55	6	70.23	8.39	6	68.31	2.49
7	81.41	9.20	7	82.58	9.95	7	73.69	2.84
8	85.35	8.84	8	83.71	9.08	8	72.46	2.66
9	87.64	7.99	9	89.32	9.28	9	75.54	2.75
10	88.92	7.45	10	88.48	8.28	10	79.54	2.92

TABLE IX: AVERAGE VALUES OF ACCURACY AND ITR FOR *EPOCHS AVERAGING* METHOD.

[R = REPETITIONS, ACC. = ACCURACY]

(a) Speller			(b) Cartesian Control			(c) High Level Control		
R	Acc. (%)	ITR	R	Acc.(%)	ITR	R	Acc.(%)	ITR
1	28.30	9.53	1	25.53	5.48	1	46.08	1.20
2	48.08	12.93	2	35.30	6.67	2	55.15	1.91
3	55.49	11.38	3	47.42	8.17	3	59.38	2.23
4	66.48	11.72	4	53.94	8.03	4	59.08	1.98
5	72.62	10.35	5	64.09	9.06	5	64.00	2.25
6	77.66	9.79	6	72.73	8.96	6	67.54	2.42
7	82.51	9.44	7	82.58	9.95	7	74.69	2.94
8	85.16	8.78	8	86.52	9.68	8	74.23	2.76
9	82.69	7.24	9	86.52	8.71	9	77.54	2.91
10	86.45	7.10	10	87.65	8.12	10	79.31	2.91

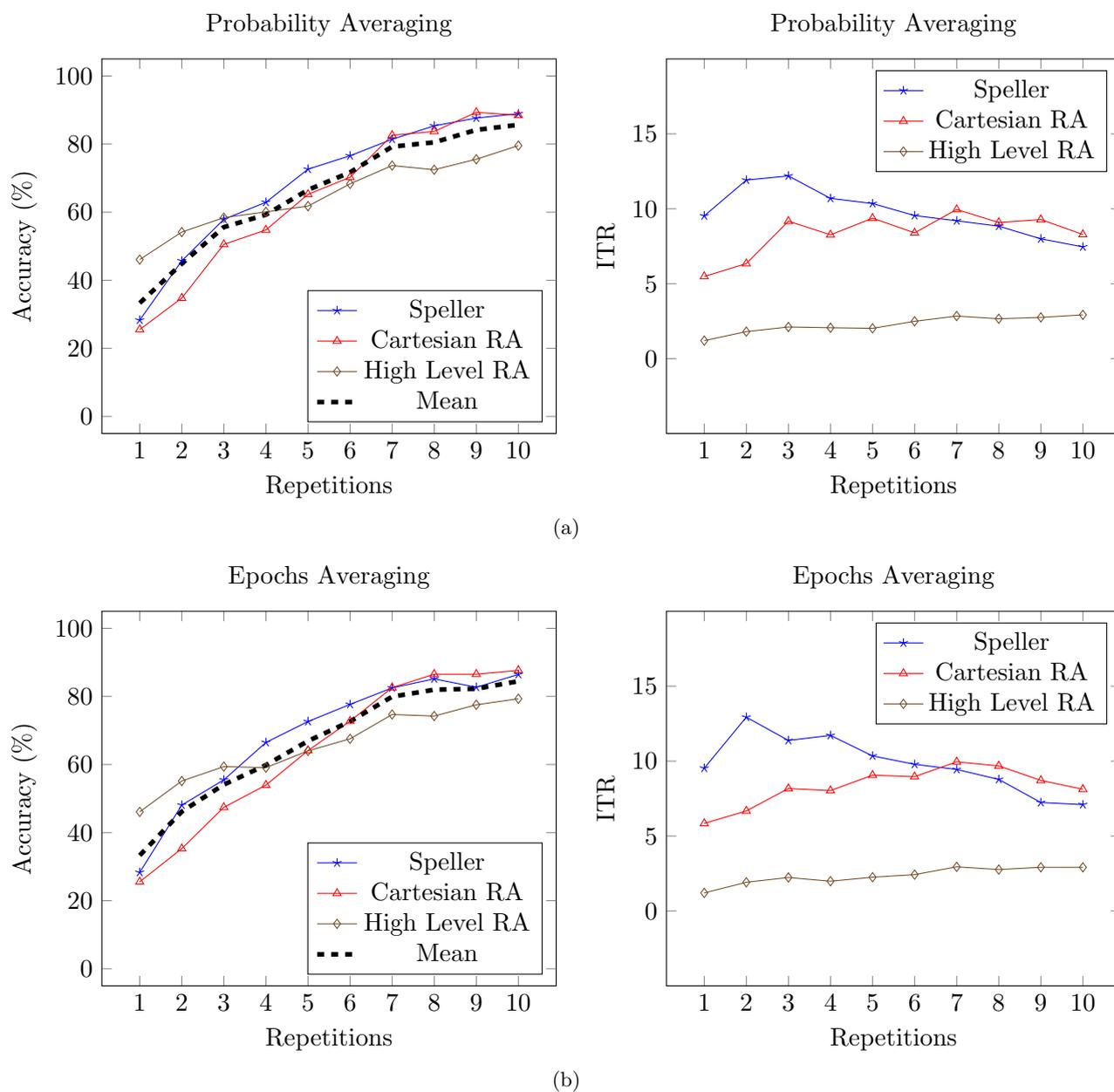


Figure 44: Average values of the accuracy and ITR amongst the two subjects as a function of the number of repetitions for the three modes of use.

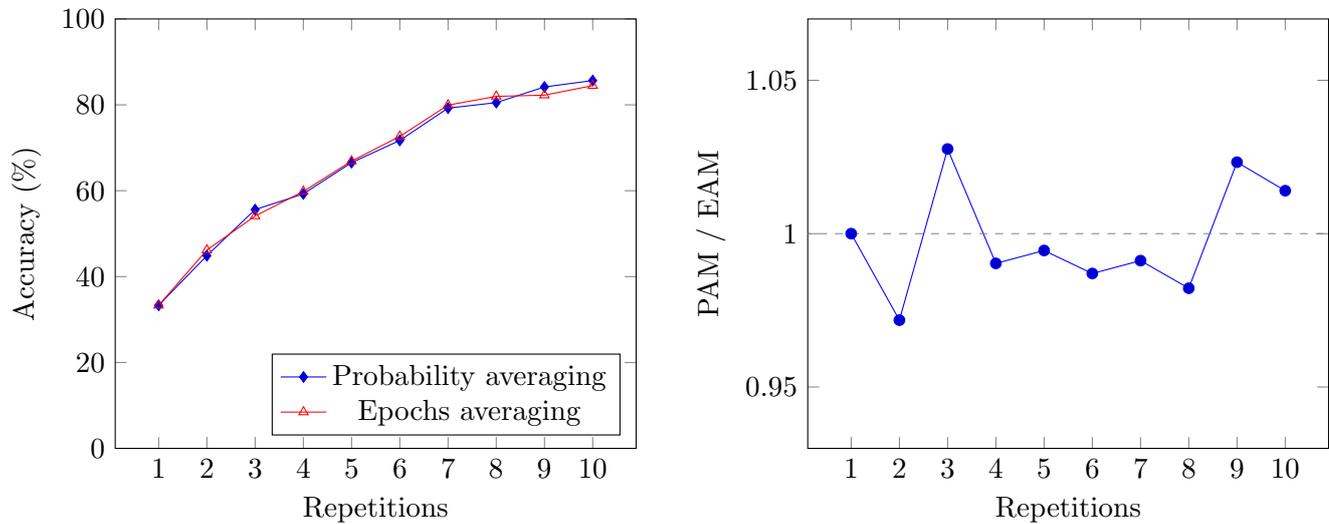


Figure 45: *Left)* Overall average accuracy among the two subjects for the three modes of use as a function of the number of repetitions. *Right)* Ratio between classification accuracy of probability averaging and epochs averaging method as a function of the number of repetitions.

### 3.4.5 Comparison between Classification Methods.

In this section, a comparison between the *probability averaging* method and the *epochs averaging* method is performed using the data shown in Table VIII and Table IX.

The ratio between between classification accuracy of the probability averaging method and classification accuracy of the epochs averaging is shown in Figure 45. In order to compute the ratio, the overall average accuracy between the two subjects and across the three modes of use was considered. A plot showing the difference between the overall average accuracy for the two methods is also shown in Figure 45. The mean ratio is  $\mu = 0.998$  and the standard deviation

is  $\sigma = 0.018$ . This results in a mean difference between classification accuracy of *probability averaging* method and *epochs averaging* method of  $-0.08 \pm 1.20\%$ .

As expected when designing the two classification methods, the two methods result in the same outcome. Data show that the maximum difference in accuracy (1.91%) happens when considering 9 repetitions. Furthermore, when using an intermediate range of repetitions (4 to 8), the difference between the two methods is negligible ( $\sim 1\%$ ).

The classifier is trained using probability averaging method. Therefore, using probability averaging method to perform online classification would be more suitable. However, in particular cases when a lot of noise is present in the EEG signal, epochs averaging methods should be employed. By averaging the epochs associated to the same stimulus, *epochs averaging* method would reduce the noise in the signal, leading to better results in terms of classification accuracy.

## CHAPTER 4

### CONCLUSIONS

In this work, a wearable and cost-effective BCI assistive device was presented. This assistive device is based on the P300 response of the human brain. Since the P300 response is an innate response of the brain, no training for the subject is required to use a P300 based BCI.

In the proposed device, EEG signal is recorded using the OpenBCI Cyton board [2]. The user wears a 3D printed headset that houses the electrodes. Due to portability of the system, the user can use the device wherever he/she desires. Furthermore, the hardware components of the system are cost effective, bringing the total cost of the device to \$ 900. Therefore, the developed device is both wearable and cost-effective. As a consequence, it could be used by many disabled individuals as an assistive technology device, helping to improve their quality of life by augmenting communication capabilities and allowing for the control of external devices.

Flashing symbols on a grid are employed as stimuli to elicit a P300 response. By analyzing the P300 responses in the recorded EEG signal, the device allows the disabled to type words on a computer screen and to control a robotic arm. Two modalities can be used to control the robotic arm. One modality (*cartesian* control) consists of controlling the robotic arm with discrete movements. The other modality (*high level* control) consists of sending high level commands to the robotic arm, which will move autonomously according to the selected action. Flashing grids are used as a form of visual stimulation. Dimensions of the grids are  $6 \times 6$  for the speller mode,  $6 \times 7$  for the speller mode including additional symbols,  $4 \times 4$  for the cartesian

control of the robotic arm,  $2 \times 3$  for the high level control of the robotic arm. Single cell flashing paradigm is used for the high level control of the robotic arm, while columns and rows flashing is employed for all the other modes.

EEG classification aimed at P300 detection is a fundamental component of the device. The genetic algorithm presented by *Dal Seno et al.* [3] is used to perform automatic feature extraction for P300 detection. Logistic regression is employed for online detection of the P300 response. Two classification methods, both based on logistic regression, were employed: *probability averaging* and *epochs averaging* method. The difference between the two methods is as follows: with *probability averaging* method, an average on the classification score space is performed; with *epochs averaging* method an average on the data space is performed. Even though training of the subject is not required, training of the classifier is necessary. Therefore, prior to being able to use the BCI live, the subject has to go through a training session which lasts approximately one hour. The training of the logistic regression classifier is performed by the genetic algorithm along with the feature extraction, and takes approximately 2 hours.

The device was validated on two male healthy subjects (20 and 24 years old). Results in terms of accuracy and information transfer rate (ITR), when using 5 and 10 repetitions, are shown in Table X. Amongst the two subjects, the average maximum ITR for the speller mode is 12.56 *bits/min*, for the cartesian control of the robotic arm is 9.95 *bits/min* and for the high level control of the robotic arm is 2.93 *bits/min*. No significant difference was found between the two employed classification methods.

TABLE X: ONLINE CLASSIFICATION RESULTS IN TERMS OF ACCURACY AND INFORMATION TRANSFER RATE. RESULTS OBTAINED USING *PROBABILITY AVERAGING* METHOD.

<i>Repetitions</i>	Accuracy		ITR ( <i>bits/min</i> )	
	5	10	5	10
Speller	72.62%	88.92%	10.35	7.45
Cartesian robotic arm control	65.23%	88.48%	9.37	8.28
High level robotic arm control	61.77%	79.54%	2.02	2.92

The achieved results are comparable to other BCI studies. The comparison we performed was limited to the online speller performance, since no other studies were done with the same setup we used for the control of the robotic arm. The comparison was done with *Dal Seno et al.*, because the same genetic algorithm employed in our system was used, and with *Thulasidas* and *Cecotti et al.* because exact values of accuracy as a function of the number of repetitions were reported in the studies [6; 4; 5]. In our study, amongst the two subjects the average values of accuracy were 72.62% when using 5 repetitions and 88.92 % when using 10 repetitions. *Dal Seno* reported the results of a P300 speller used online by two subjects [6]. EEG signal was acquired with *BE Light* by EBNeuro [47]. The software used for EEG signal acquisition and processing was BCI2000 [24]. One subject achieved 68% online accuracy when using 4 repetitions. The other subject achieved 68% online performance with 5 repetitions. *Thulasidas et al.* presented a robust classification method for P300 detection [4]. For this study, EEG

signal was acquired using Neuroscan SynAmps2 [48]. The software for the P300 speller was developed by the authors. Average accuracy across 9 subjects when using 5 repetitions was 79.2%. *Cecotti et al.* developed seven convolutional neural network based classifiers for P300 detection [5]. To test the classifiers, dataset from the third BCI competition was employed [69]. This data set contains a record of P300 evoked potentials from two subjects. The signals was recorded in five sessions with the BCI2000 software [24]. Average accuracy across the 7 classifiers between the two subjects, when using 5 repetitions, was 62.57%. In this study, data about ITR were also reported. When using 5 repetitions, average ITR amongst the two subjects is in the range of 8 - 13 *bits/min*, according to the chosen classifier. *Kronegg et al.* reported the average value of ITR for several BCI systems, which is equal to 11 *bits/min* [64] .

In all the above studies, clinical instrumentation for EEG acquisition was used. Our results on two subjects show that it is possible to obtain comparable results in terms of accuracy and ITR when using cost-effective hardware for EEG acquisition. More advanced techniques for classification could be employed to improve the functionality of the device by increasing the ITR. *Speier et al.* [59] implemented techniques of natural language processing and dynamic classification for a P300 speller. The use of these methods lead to a great increase in ITR when compared to a standard classification technique, SWLDA (33.15 *b/m* and 27.69 *b/m* vs 22.07 *b/m*). *Krusienski et al.* compared the effects of spatial channel selection, EEG referencing, decimation of channel data, and maximum number of features in order to improve P300 speller classification methods [62].

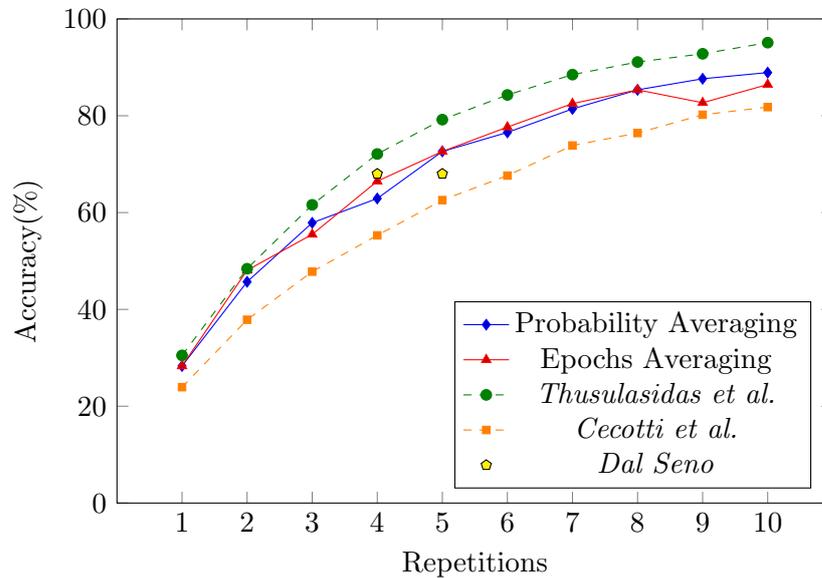


Figure 46: Comparison between the present study and *Thusulasidas et al.* [4], *Cecotti et al.* [5], *Dal Seno* [6] for online classification accuracy as a function of the number of repetitions in the speller mode.

In conclusion, in the present work we demonstrated that P300 response detection with cost-effective EEG acquisition devices is comparable to results obtained with clinical grade EEG equipment. Furthermore, using a P300 BCI, the control of a complex device, a 4 DoF robotic arm, was achieved.

According to the Amyotrophic Laterals Sclerosis Association, only one BCI device, the Intendix system, is commercially available at the moment [68; 66]. As stated by *Kübler*, low cost EEG acquisition devices render translational studies increasingly feasible [65]. We believe that our work will help to transition from BCI use to in-home use, allowing more and more individuals to improve their quality of life.

#### 4.1 Future Improvements

Improvements of the device were identified during the device validation phase, and are as follows.

1. **Improved EEG headset:** the current 3D printed EEG headset may not be comfortable for long term-use. A new design is necessary to improve the comfort of the headset. The new design should take into account the high electrode-to-skin impedance values often obtained with the current version. Decreasing the impedance would allow for recording higher quality signals, and to improve classification accuracy.
2. **Flashing of menu buttons:** since selection of menu items has to be performed with a mouse, the BCI software cannot be initialized by the disabled without the help of the caregiver. Employing automatic flashing menu buttons, an updated version of the software interface would improve upon the existing device.
3. **High level control of the robotic arm:** with the current firmware of the robotic arm, when using the high level control the robotic arm expects objects to be in predefined locations. An updated version of both firmware and hardware of the arm, would allow the robotic arm to automatically recognize where objects are located, thus improving the functionality of the system.
4. **Validation by disabled users:** for this work, test on healthy subjects were performed. Test on disabled individuals are necessary to discover possible key issues present when the device is used by disabled individuals.

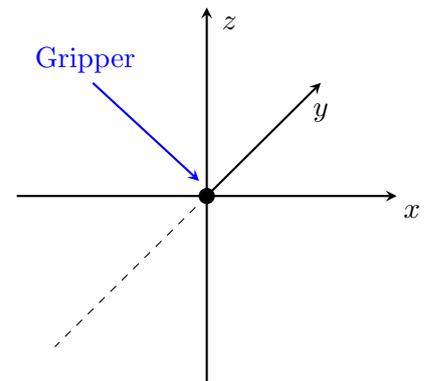
## APPENDICES

## APPENDIX A

## DESCRIPTION OF ROBOTIC ARM MOVEMENTS

TABLE XI: DETAILED DESCRIPTION OF THE SYMBOL USED FOR THE STIMULATION IN THE CARTESIAN CONTROL OF THE ROBOTIC ARM..

Movement	X	Y	Z	Description
	-		+	Move up and to the left
	+		+	Move up and to the right
			+	Move up
	-		-	Move down and to the left
	+		-	Move down and to the right
			-	Move down
	-			Move to the left
	+			Move to the right
		+		Move forwards
		-		Move backwards
				Rotate wrist up
				Rotate wrist down
				Exit cartesian control
				Back to start position
				Open the gripper
				Close the gripper



## APPENDIX B

## COPYRIGHT INFORMATION



Home Create Account Help 



**Title:** Control of a 9-DoF Wheelchair-mounted robotic arm system using a P300 Brain Computer Interface: Initial experiments

**Conference Proceedings:** Robotics and Biomimetics, 2008. ROBOT 2008. IEEE International Conference on

**Author:** Mayur Palankar

**Publisher:** IEEE

**Date:** Feb. 2009

Copyright © 2009, IEEE

LOGIN

**If you're a copyright.com user,** you can login to RightsLink using your copyright.com credentials. Already a **RightsLink user** or want to [learn more?](#)

**Thesis / Dissertation Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications\\_standards/publications/rights/rights\\_link.html](http://www.ieee.org/publications_standards/publications/rights/rights_link.html) to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

## APPENDIX B (continued)



RightsLink®

Home

Create Account

Help



**Title:** An SSVEP BCI to Control a Hand Orthosis for Persons With Tetraplegia

**Author:** Rupert Ortner

**Publication:** Neural Systems and Rehabilitation Engineering, IEEE Transactions on

**Publisher:** IEEE

**Date:** Feb. 2011

Copyright © 2011, IEEE

LOGIN

If you're a **copyright.com user**, you can login to RightsLink using your copyright.com credentials. Already a **RightsLink user** or want to [learn more?](#)

## Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications\\_standards/publications/rights/rights\\_link.html](http://www.ieee.org/publications_standards/publications/rights/rights_link.html) to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

## APPENDIX B (continued)



RightsLink®

Home

Create Account

Help



**Title:** A genetic algorithm for automatic feature extraction in P300 detection

**Conference Proceedings:** Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on

**Author:** Bernardo Dal Seno

**Publisher:** IEEE

**Date:** June 2008

Copyright © 2008, IEEE

LOGIN

If you're a **copyright.com user**, you can login to RightsLink using your copyright.com credentials. Already a **RightsLink user** or want to [learn more?](#)

**Thesis / Dissertation Reuse**

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to [http://www.ieee.org/publications\\_standards/publications/rights/rights\\_link.html](http://www.ieee.org/publications_standards/publications/rights/rights_link.html) to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

## CITED LITERATURE

1. World Health Organization: Fact Sheet n.384. World Health Organization, 2013.
2. OpenBCI: OpenBCI board <http://openbci.com> [Online; Accessed: 03/05/2017].
3. Dal Seno, B., Matteucci, M., and Mainardi, L.: A genetic algorithm for automatic feature extraction in P300 detection. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pages 3145–3152, 2008.
4. Thulasidas, M., Cuntai, G., and Jiankang, W.: Robust classification of EEG signal for brain-computer interface. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, 14(1):24–29, 2006.
5. Cecotti, H. and Graser, A.: Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(3):433–445, 2011.
6. Dal Seno, B.: Toward an integrated P300-and ErrP-based brain-computer interface. Politecnico di Milano, 2009.
7. Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M.: Brain-computer interfaces for communication and control. Clinical neurophysiology, 113(6):767–791, 2002.
8. Damper, R., Burnett, J., Gray, P., Straus, L., and Symes, R. A.: Hand-held text-to-speech device for the non-vocal disabled. Journal of Biomedical Engineering, 9(4):332–340, 1987.
9. Kubota, M., Sakakihara, Y., Uchiyama, Y., Nara, A., Nagata, T., Nitta, H., Ishimoto, K., Oka, A., Horio, K., and Yanagisawa, M.: New ocular movement detector system as a communication tool in ventilator-assisted werdnig-hoffmann disease. Developmental Medicine and Child Neurology, 42(1):61–64, 2000.

### CITED LITERATURE (continued)

10. Kobetic, R., To, C. S., Schnellenger, J. R., Bulea, T. C., CO, R. G., and Pinault, G.: Development of hybrid orthosis for standing, walking, and stair climbing after spinal cord injury. Journal of rehabilitation research and development, 46(3):447, 2009.
11. Luo, Z., Han, S., and Duan, F.: The development of a smart house system based on brain-computer interface. In 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 1012–1017, Dec 2015.
12. Ball, T., Kern, M., Mutschler, I., Aertsen, A., and Schulze-Bonhage, A.: Signal quality of simultaneously recorded invasive and non-invasive EEG. NeuroImage, 46(3):708–716, 2009.
13. Middendorf, M., McMillan, G., Calhoun, G., and Jones, K. S.: Brain-computer interfaces based on the steady-state visual-evoked response. IEEE Transactions on Rehabilitation Engineering, 8(2):211–214, Jun 2000.
14. Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kubler, A., Perelmouter, J., Taub, E., and Flor, H.: A spelling device for the paralysed. Nature, 398(6725):297–298, mar 1999.
15. Pfurtscheller, G. and Neuper, C.: Motor imagery and direct brain-computer communication. Proceedings of the IEEE, 89(7):1123–1134, Jul 2001.
16. Picton, T. W.: The p300 wave of the human event-related potential. Journal of clinical neurophysiology, 9(4):456–479, 1992.
17. Farwell, L. and Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and Clinical Neurophysiology, 70(6):510–523, 1988.
18. Iturrate, I., Antelis, J. M., Kubler, A., and Minguez, J.: A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation. IEEE Transactions on Robotics, 25(3):614–627, June 2009.
19. Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., Ang, M., and Laugier, C.: A brain-controlled wheelchair based on p300 and path guidance. In The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006., pages 1101–1106, Feb 2006.

### CITED LITERATURE (continued)

20. Congedo, M., Goyat, M., Tarrin, N., Varnet., L., Rivet, B., Ionescu, G., Jrad, N., Phlypo, R., Acquadro, M., and Jutten, C.: Brain Invaders: a prototype of an open-source P300-based video game working with the OpenViBE platform. 5th International BCI Conference, Graz, Austria, 280-283, 2011(Bci):1–6, 2011.
21. Finke, A., Lenhardt, A., and Ritter, H.: The MindGame: A P300-based brain-computer interface game. Neural Networks, 22(9):1329–1333, 2009.
22. Palankar, M., De Laurentis, K. J., Alqasemi, R., Veras, E., Dubey, R., Arbel, Y., and Donchin, E.: Control of a 9-DoF wheelchair-mounted robotic arm system using a P300 brain computer interface: Initial experiments. 2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008, pages 348–353, 2008.
23. <https://www.nibib.nih.gov/sites/default/files/r-arm.pdf>, [Online; Accessed 02/18/2017], 2013.
24. Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., and Wolpaw, J. R.: Bci2000: a general-purpose brain-computer interface (bci) system. IEEE Transactions on Biomedical Engineering, 51(6):1034–1043, June 2004.
25. Bell, C. J., Shenoy, P., Chalodhorn, R., and Rao, R. P.: Control of a humanoid robot by a noninvasive brain-computer interface in humans. Journal of neural engineering, 5(2 PG - 214-220):214–220, 2008.
26. Chaudhary, U., Xia, B., Silvoni, S., Cohen, L. G., and Birbaumer, N.: Brain-computer interface-based communication in the completely locked-in state. PLoS Biology, 15(1):e1002593, 2017.
27. Ortner, R., Allison, B. Z., Korisek, G., Gaggl, H., and Pfurtscheller, G.: An ssvep bci to control a hand orthosis for persons with tetraplegia. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 19(1):1–5, Feb 2011.
28. Martius, I. and Damaevius, R.: A prototype SSVEP based real time BCI gaming system. Computational Intelligence and Neuroscience, 2016, 2016.
29. Brodmann, K.: Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues. Barth, 1909.
30. Berger, H.: Über das elektrenkephalogramm des menschen. European Archives of Psychiatry and Clinical Neuroscience, 87(1):527–570, 1929.

### CITED LITERATURE (continued)

31. Jasper, H. H.: The ten twenty electrode system of the international federation. Electroencephalography and clinical neurophysiology, 10:371–375, 1958.
32. Cohen, D. et al.: Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. Science, 161(3843):784–786, 1968.
33. Sitaram, R., Caria, A., Veit, R., Gaber, T., Rota, G., Kuebler, A., and Birbaumer, N.: Fmri brain-computer interface: a tool for neuroscientific research and treatment. Computational intelligence and neuroscience, 2007, 2007.
34. Birbaumer, N.: Slow cortical potentials: their origin, meaning, and clinical use. Brain and Behavior Past, Present, and Future, pages 25–39, 1997.
35. Duncan-Johnson, C. C. and Donchin, E.: On quantifying surprise: The variation of event-related potentials with subjective probability. Psychophysiology, 14(5):456–467, 1977.
36. Ruchkin, D. S., Sutton, S., Kietzman, M. L., and Silver, K.: Slow wave and p300 in signal detection. Electroencephalography and Clinical Neurophysiology, 50(1):35–47, 1980.
37. Sellers, E. W., Krusienski, D. J., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R.: A p300 event-related potential brain-computer interface (bci): the effects of matrix size and inter stimulus interval on performance. Biological psychology, 73(3):242–252, 2006.
38. Frey, J.: Comparison of an open-hardware electroencephalography amplifier with medical grade device in brain-computer interface applications. CoRR, abs/1606.02438, 2016.
39. Emotiv: Emotiv Epoch <https://www.emotiv.com> [Online; Accessed: 03/05/2017].
40. Kinova Robotics: Kinova Jaco <http://www.kinovarobotics.com/service-robotics/build-prices/> [Online; Accessed 03/11/2017].
41. gTech: g.USBamp <http://www.gtec.at/products/hardware-and-accessories/g.usbamp-specs-features> [Online; Accessed 03/11/2017].
42. Lynxmotion: Al5B Robotic Arm, <http://www.lynxmotion.com/c-126-al5b.aspx> [Online; Accessed 03/18/2017].

### CITED LITERATURE (continued)

43. Lynxmotion: BotBoarduino <http://www.lynxmotion.com/c-153-botboarduino.aspx> [Online; Accessed 03/18/2017].
44. Arduino: Arduino Duemilanove <https://www.arduino.cc/en/main/arduinoboardduemilanove> [Online; Accessed 03/18/2017].
45. Microchip: PIC32MX250F128B, <http://www.microchip.com/wwwproducts/en/en557425> [Online; Accessed 03/18/2017].
46. Instrument, T.: ADS1299 Low-Noise, 8-Channel, 24-Bit Analog-to-Digital Converter for Biopotential Measurements, <http://www.ti.com/lit/ds/symlink/ads1299.pdf> [Online; Accessed 03/18/2017].
47. EBNeuro: <http://www.ebneuro.biz/> [Online; Accessed 03/18/2017].
48. Neuroscan: <http://compumedicsneuroscan.com/tag/synamps/> [Online; Accessed 03/18/2017].
49. OpenBCI: Ultracortex Mark III headset , <http://docs.openbci.com/headware/02-ultracortex-mark-iii-nova-revised> [Online; Accessed 04/05/2017].
50. Processing Foundation: Processing, version 3.0, build 0253. Available at <http://www.processing.org> [Online; Accessed 03/03/2017].
51. Vercellis, C.: Business intelligence: data mining and optimization for decision making. John Wiley & Sons, 2011.
52. Kohavi, R. et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai, volume 14, pages 1137–1145. Stanford, CA, 1995.
53. Vidakovic, B.: Statistics for bioengineering sciences: with MATLAB and WinBUGS support. Springer Science & Business Media, 2011.
54. Sivanandam, S. and Deepa, S.: Introduction to genetic algorithms. Springer Science & Business Media, 2007.
55. Boostani, R., Graimann, B., Moradi, M. H., and Pfurtscheller, G.: A comparison approach toward finding the best feature and classifier in cue-based BCI. Medical and Biological Engineering and Computing, 45(4):403–412, 2007.

### CITED LITERATURE (continued)

56. Citi, L., Poli, R., Cinel, C., and Sepulveda, F.: Feature Selection and Classification in Brain Computer Interfaces by a Genetic Algorithm. Late breaking papers of the Genetic and Evolutionary Computation Conference (GECCO-2004), 2004.
57. Hall, J. E.: Guyton and Hall textbook of medical physiology. Elsevier Health Sciences, 2015.
58. Kemp, B. and Olivan, J.: European data format plus(edf+), an edf alike standard format for the exchange of physiological data. Clinical Neurophysiology, 114(9):1755–1761, 2003.
59. Speier, W., Arnold, C., Lu, J., Taira, R. K., and Pouratian, N.: Natural language processing with dynamic classification improves P300 speller accuracy and bit rate. Journal of neural engineering, 9(1):016004, 2012.
60. McFarland, D. J., Sarnacki, W. A., Townsend, G., Vaughan, T., and Wolpaw, J. R.: The P300-based brain computer interface (BCI): Effects of stimulus rate. Clinical Neurophysiology, 122(4):731–737, 2011.
61. Donchin, E., Spencer, K. M., and Wijesinghe, R.: The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. IEEE Transactions on Rehabilitation Engineering, 8(2):174–179, 2000.
62. Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R.: Toward enhanced P300 speller performance. Journal of Neuroscience Methods, 167(1):15–21, 2008.
63. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., and Lécuyer, A.: OpenViBE: An Open-Source Software Platform to Design, Test, and Use BrainComputer Interfaces in Real and Virtual Environments. Presence: Teleoperators and Virtual Environments, 19(1):35–53, 2010.
64. Kronegg J., V. S. P. T.: Analysis of bit-rate definitions for Brain-Computer Interfaces. Proceedings of the 2005 International Conference on Human-Computer Interaction, HCI'05, pages 40–46, 2005.
65. Kübler, A.: Quo vadis P300 BCI? Building on results for the benefit of BCI end-users. pages 36–39.
66. Intendix: <http://www.intendix.com/> [Online, accessed 04/01/2017].

**CITED LITERATURE (continued)**

67. Biosemi: <https://www.biosemi.com/products.htm> [Online, accessed 04/01/2017].
68. ALS: <http://www.alsa.org/als-care/resources/publications-videos/factsheets/brain-computer-interface.html> [Online, accessed 04/01/2017].
69. Blankertz, B., Muller, K.-R., Curio, G., Vaughan, T. M., Schalk, G., Wolpaw, J. R., Schlogl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., et al.: The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials. IEEE transactions on biomedical engineering, 51(6):1044–1051, 2004.

## VITA

NAME	Davide Marzorati
<hr/>	
EDUCATION	
Current	Master of Science in Bioengineering, University of Illinois at Chicago, USA. GPA: 4.00/4.00
Current	Master of Science in Biomedical Engineering, Politecnico of Milano, Italy. Track: Technologies for Electronics — GPA: 29.74/30.
Jul 2015	Bachelor of Science in Biomedical Engineering, Politecnico di Milano, Italy. Final grade: 103/110 — GPA: 26.23/30.
2007-2012	Scientific High School Degree, Liceo Scientifico “P.Giovio”, Como.
<hr/>	
BACHELOR THESIS	
Title	Analysis and evaluation of visuospatial memory span.
Advisor	Professor Giancarlo Ferrigno
Description	In this thesis we evaluated, through cognitive tests, the visuospatial memory span of a sample of 60 people; then, we analyzed the relation between memory span and age and between the three different components of visuospatial memory span.
<hr/>	
LANGUAGE SKILLS	
Italian	Native speaker
English	Full working proficiency 2014 - TOEFL examination (105/120) 2011 - FCE (75/100)
Spanish	Basic knowledge 2011 - DELE Certification (84.11/100)
<hr/>	
TECHNICAL SKILLS	
Basic level	C++, Atmel Studio, Phyton, R
Average level	C, MATLAB, Latex
<hr/>	