Development of a Hybrid Brain-Computer Interface for Autonomous Systems

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) am Fachbereich Mathematik und Informatik der Freien Universität Berlin

von

Omar Mendoza Montoya

aus Mexiko

Berlin 2017

Development of a Hybrid Brain-Computer Interface for Autonomous Systems

DISSERTATION

Autor: Omar Mendoza Montoya

Tag der Disputation: 19. Dezember 2017

Erstgutachter: Prof. Dr. Raúl Rojas (Freie Universität Berlin)

Zweitgutachter: Prof. Dr. Petra Ritter (Charité – Universitätsmedizin Berlin)

Diese Arbeit wurde gefördert von der CONACYT-DAAD.

Summary

This work presents an EEG-based brain-computer interface (BCI) for controlling autonomous devices. The purpose of this system is to restore some mobility to people with severe neuromuscular disorders who cannot operate an electric wheelchair or other devices with physical interfaces. This BCI was tested with an autonomous wheelchair in different scenarios, including open places.

The BCI combines P300-evoked responses and steady-state visual evoked potentials (SSVEPs) to decode the user's intentions. Target places and other navigation commands are selected with a P300-based interactive menu, whereas the device is stopped with the modulation of SSVEP oscillations. To reduce the risk of producing inadvertent commands when the user is not using the interface, the P300-based BCI can only be used after unlocking this function with the SSVEP-based BCI. The main advantages of this hybrid design compared to P300-based interfaces are that the wheelchair can be stopped quickly without cognitive effort, and false positives for the options of the interactive menu are reduced to a level close to zero.

The autonomy of the P300-based BCI is guaranteed by adapting the classifier during the normal operation of the interface. Newly recorded data is used to adjust the classifier so that the performance of the BCI is not affected by non-stationarities in long experimental sessions. Furthermore, this thesis presents a novel method for auto-calibrating the P300-based BCI that does not require labeled data and the user's intervention. The auto-calibration and the adaptation of the P300 classifier allow operating the interface in optimal conditions at all times without assistance.

The control of false positives in the SSVEP-based BCI is also addressed in this work. To this end, it is proposed a new method for detecting SSVEP-related activity which requires minimal training and is easy to adapt. This new approach only requires resting state data to find the parameters of the classifier that maintains the false positive rate below the desired level. In the worst cases, no more than 0.067 false positives per minute were detected in the online tests with the wheelchair.

The proposed BCI also incorporates two passive subsystems to quantify mind states and error-related potentials (ErrPs). The former is used to measure brain oscillations related to task engagement and fatigue. The quantification of both variables has the potential to improve the interaction between the user and the smart device. On the other hand, the other BCI is used to detect machine errors automatically when the subject perceives that the P300-based BCI has selected one option incorrectly. This automatic feedback may be used to improve the performance of the hybrid-BCI when the P300 classifier has low accuracy. However, if the system detects P300 responses correctly most of the time, the ErrP-based BCI is not necessary and may slow down the application.

The final result is a hybrid BCI that analyzes, at the same time, different components of electrical neural activity. This work includes an integral evaluation of the four implemented BCIs and a final test of the hybrid architecture with the autonomous wheelchair. The results presented in this document demonstrate that the proposed interface is very efficient in communicating the user's intentions to the autonomous device. It is shown that subjects without a priori knowledge of BCI technology can master the interface after a few minutes of training and achieve accuracies of 100%.

Acknowledgements

I wish to thank, first and foremost, my advisor *Prof. Dr. Raúl Rojas* for his extraordinary support and guidance in this thesis project. I am in debt to him for giving me the opportunity to work in this amazing research field.

I would like to express my gratitude to *Prof. Dr. Petra Ritter* for giving me the chance to collaborate with her research group. The most important steps in the development of my project were possible thanks to many fruitful discussions on neurobiology and BCI technology that took place at Charité.

I also thank *Dr. Adalberto Llarena* for helping me to test my BCI with the autonomous wheelchair. I greatly appreciate his constant support and advice.

Special thanks go to my friend *Dr. Salvador Sierra Murillo* for taking the time to review this thesis. His expert feedback helped me to improve the structure and quality of the contents of this document.

My deepest gratitude to all my friends and colleagues *Margarita*, *Fernando*, *Sebastián*, *José*, and *Ricardo*, as well as all the members of the *BrainModes* research group. Every comment, suggestion and advice has been taking into account in this long development process.

Finally, I would like to thank my wife *Ana Belem*, and my daughter *Amanda* for their unlimited love, patient and support. You truly are each an inspiration and my life.

This project would have been impossible without the support of the *Consejo Nacional de Ciencia y Tecnología* (CONACYT) and the *Deutsche Akademische Austauschdienst (DAAD)*.

Contents

List of Abbreviations

AR	**A**uto**R**egressive
ASR	**A**rtifact **S**ubspace **R**econstruction
ALS	**A**myotrophic **L**ateral **S**clerosis
BCI	**B**rain-**C**omputer **I**nterface
BMI	**B**rain-**M**achine **I**nteraction
BSS	**B**lind **S**ource **S**eparation
CAR	**C**ommon **A**verage **R**eference
CCA	**C**anonical **C**orrelation **A**nalysis
CSP	**C**ommon **S**patial **P**attern
CSSP	**C**ommon **S**patio-**S**pectral **P**attern
CSSSP	**C**ommon **S**parse **S**pectral-**S**patial **P**attern
ECG	**E**lectro**C**ardio**G**ram
ECoG	**E**lectro**C**orticO**G**raphy
EEG	**E**lectro**E**ncephalo**G**raphy
EMG	**E**lectro**M**yo**G**raphy
EOG	**E**lectro**O**culo**G**ram
ERD	**E**vent-**R**elated **D**esynchronization
ERP	**E**vent-**R**elated **P**otential
ErrP	**E**rror **P**otential
ERS	**E**vent-**R**elated **S**ynchronization
FDR	**F**alse **D**iscovery **R**ate
FEP	**F**inal **P**rediciton **E**rror

FES	**F**unctional **E**lectrical **S**timulation
FFT	**F**ast **F**ourier **T**ransform
FIR	**F**inite **I**mpulse **R**esponse
fMRI	**f**unctional **M**agnetic **R**esonance **I**maging
FNR	**F**alse **N**egative **R**ate
fNIRS	**f**unctional **N**ear-**I**nfra**R**ed **S**pectroscopy
FPR	**F**alse **P**ositive **R**ate
GDA	**G**eneralized **D**iscriminant **A**nalysis
GUI	**G**raphical **U**ser **I**nterface
ICA	**I**ndependent **C**omponent **A**nalysis
ITR	**I**nformation **T**transfer **R**ate
IIR	**I**nfinite **I**mpulse **R**esponse
LDA	**L**inear **D**iscriminant **A**nalysis
LOOCV	**L**eave-**O**ne-**O**ut **C**ross-**V**alidation
LRT	**L**ikelihood **R**atio **T**est
MVD	**M**y **V**irtual **D**ream
MISE	**M**ean **I**ntegrated **S**quared **E**rror
OVL	**OV**er**L**apping Coefficient
PCA	**P**rincipal **C**omponent **A**nalysis
PSD	**P**ower **S**pectral **D**ensity
QDA	**Q**uadratic **D**iscriminant **A**nalysis
RBF	**R**adial **B**asis **F**unction
RCSP	**R**egularized **C**ommon **S**patial **P**attern
SCI	**S**pinal **C**ord **I**njury
SCP	**S**low **C**ortical **P**otential
SQF	**S**inusoidal **Q**uadrature **F**ilter
SVD	**S**ingular **V**alue **D**ecomposition
SVM	**S**uport **V**ector **M**achine

SSVEP **S**teady-**S**tate **V**isual **E**voked **P**otential

SWCSP **S**pectrally-**W**eighted **C**ommon **S**patial **P**attern

TVEP **T**ransient **V**isual **E**voked **P**otential

TPR **T**rue **P**ositive **R**ate

VAR **V**ector **A**uto**R**egressive

VEP **V**isual **E**voked **P**otential

Dedicated to my daughter Amanda...

Chapter 1

Introduction

For generations, the ability to dig into a person's mind or being able to manipulate machines through thought alone have captured the imagination of people all over the world. Fantasy stories and science fiction novels are evidence of how these ideas have fascinated humanity for a long time. However, in the last two decades, researchers have begun to build devices able to interpret neurological processes that occur within the human brain. The rapid development of computer hardware and software engineering and recent advances in cognitive science have provided the theoretical and practical foundations of a new type of technology known as brain-computer interfaces.

A brain-computer interface (or BCI) is any system that monitors brain's physiological processes (electrical signals, blood oxygenation, etc.), and translates this activity into meaningful information. It allows communicating the user's intentions to a computer, robot or any other device without the need of traditional input channels (mouse, keyboard, joystick, microphone, etc.). Also, BCIs obtain real-time information about the subject's mental states (task engagement, relaxation, boredom, satisfaction, frustration, etc.), which can be used to provide a more natural interaction between the users and the machines around them.

BCI research is a relatively young multidisciplinary field which involves scientists and engineers from neuroscience, rehabilitation, computer science, electrical engineering, robotics, applied statistics, ergonomy, and other technical and health-care areas (Nicolas-Alonso & Gomez-Gil, 2012). Since the first proofs of concept of this technology in the early 2000s, hundreds of active research groups have contributed to developing new brain-controlled applications, driven mainly by the possibility of bringing new communication channels to people with different capabilities.

In spite of all the efforts in the last two decades to develop robust and reliable BCI systems, brain-controlled applications are still at a very early stage. There are several challenges to solve before seeing this technology in everyday life. So far, the most known and reliable BCI control strategies provide low transfer rates of information (Nicolas-Alonso & Gomez-Gil, 2012), and remain useful only in highly controlled environments. Also, decoding brain data is a challenging task because of their high variability. The system must find relevant information from signals generated by multiple superposed mind processes related and not related to the control task. The time spent on preparation, the training and calibrations routines, and the design of devices and computer programs that truly exploits the possibilities of BCIs, are other open topics in the research community.

This thesis presents the development of a new BCI system designed to interact with an autonomous wheelchair (Figure 1.1). This smart device can navigate safely between places and includes other functions to assist people with mobility limitations. The BCI is based on the *electroencephalogram* (or EEG), which is a non-invasive technique that measures electrical potentials on the scalp. The proposed system incorporates three main strategies to communicate the user's intentions to the autonomous device. The first strategy consists of selecting through the BCI the next destination or another *high-level* function of the autonomous wheelchair. In the second strategy, the user navigates semi-freely by choosing the direction of the next step (forward, backward, left or right), while the autonomous wheelchair corrects the path and avoids obstacles. Finally, the user can also select *low-level* functions for rotating and translating the wheelchair, which is useful to modify the pose and the position of the device.

Figure 1.1. Autonomous wheelchair.

The implemented BCI combines two control paradigms: *P300 evoked potentials*, and *steady-state visual evoked potentials* (SSVEP). Moreover, it also incorporates a passive system for monitoring cognitive states, and another BCI to detect machine errors perceived by the user during the operation of the P300-based interface. The final result is a multi-paradigm brain-computer interface (or hybrid BCI) that analyses different aspects of the electrical brain activity in concurrent processes. This document explains how these elements are combined to provide the information required by the autonomous device to navigate to the desired destination.

Another significant contribution of this work is the development of a novel method for auto-calibrating P300-based BCIs. This method finds a new set of observations for training the system without user's intervention. Additionally, the P300-based interface has a mechanism for auto-adapting the BCI as new data is processed and classified. Both strategies were implemented to improve the autonomy of the BCI by reducing the external intervention to train or adjust the classifier. In this work, it is demonstrated that the P300-based BCI works all the time in optimal conditions even after several hours of operation.

The classification accuracy and the control of false positives are other problems addressed in this thesis. To this end, it is proposed a new adaptive method for finding the optimal number of flashes required to classify P300 evoked potentials correctly. The BCI calculates this quantity according to the user's performance and the classification accuracy of the system estimated in the training phase.

Furthermore, this work also presents a novel method for classifying SSVEP-related activity. This new approach has the advantage of simplifying the process of searching the optimal configuration to control false positives in SSVEP-based BCIs without sacrificing speed.

Finally, this document also sketches the software architecture of the hybrid brain-computer interface. To this end, it is explained how the four implemented BCI paradigms (P300, SSVEPs, passive BCI for cognitive states and passive BCI for error detection) interact in the same application, sharing preprocessed data, and analyzing different components of the EEG signals.

1.1 Motivation

The potential medical and social impact of BCI technology has motivated the research community to develop a variety of brain-controlled applications that target completely different needs, ranging from communication interfaces for severely impaired people to mini-games for merely entertainment purposes. Opportunely, BCIs are just coming out the laboratory, and there is a long way until brain-computer interfaces are another viable option for anybody to interact with machines. This work is an effort in this direction. It addresses the problem of designing, implementing and testing a multi-paradigm BCI for controlling smart devices in different scenarios.

Electric wheelchairs represent a critical assistive device because this technology helps people with motor impairments to restore some level of mobility. For this reason, the wheelchair control is one of the most important potential applications of BCI technology. The proposed system was designed considering the limitations of this target population, reducing the cognitive effort for solving complex navigation tasks. It is shown in this document that the proposed brain-computer interface provides all the information needed by the autonomous system to navigate inside a building and other places regardless the physical limitations of the user.

It is important to mention that neuroscience can also greatly benefit from BCI research. The development of brain-computer interfaces implicates the collaboration of scientists that in the past were not involved at all in the study of cognitive processes, but now they can contribute to the community with new questions and ideas. Furthermore, any new method developed to find relevant information from brain signals can be used in both cognitive experiments and BCIs. Although it is not demonstrated formally, the methods proposed in this work could eventually be applied to data collected in another kind of experiments.

Finally, BCI could mean a significant change in how cognitive scientists design experiments and collect data. Typically, the EEG device (or any other recording system) plays a passive role in the whole experiment. The participants are instructed to perform a task in a determined moment, while the device records the brain activity. After repeating the same task several times, scientists run some tests on the data and extracts the required information. In this setup, all the environmental conditions are extremely controlled, and rarely the subjects realistically solve the task. However, thanks to the introduction of wireless technology for EEG recording, now it is possible to conduct formal tests outside the laboratory. Furthermore, the experiment can also be adapted dynamically according to the subject's performance. The real-time processed information provided by the brain-computer interface may help to investigate subject-specific characteristics which can only be discovered during the run of customized experiments.

1.2 Current Challenges in the Design of BCI Systems

There are many examples that demonstrate how brain-computer interfaces are promising tools for different clinical conditions. Researchers have shown their usefulness in various levels and diverse applications. Unfortunately, despite the most recent advances in cognitive science and computational technology, the design of practical brain-controlled systems faces several challenges which must be solved before employing BCI technology in real-world tasks. At present, medical applications of BCIs are restricted to the laboratory, and most of the published works have only proved how they could eventually be incorporated in assistive technologies. For instance, Priftis (2014) made an extensive review of P300 spellers and concluded that the goal of translating these BCIs into mental prostheses for everyday use for patients with *amyotrophic lateral sclerosis* (ALS), has not been met yet. The same conclusion can be extended to other types of BCIs, except systems designed to measure cognitive states, which have been introduced to the general public by small companies that sell low-cost EEG headsets.

The electroencephalography is by far the data acquisition technique most used in BCI systems. This is because it offers an acceptable signal quality and combines low cost and easy-to-use equipment. Invasive techniques such as *electrocorticography* (ECoG) have proved to be more efficient and more stable than their non-invasive counterparts (Leuthardt et al., 2011; Schalk & Leuthardt, 2011). However, the risks for implanting electrode arrays inside the skull, the costs of these techniques, and the durability of the internal implants, outweigh the possible benefits of these brain imaging modalities. Other non-invasive techniques such as *functional magnetic resonance imaging* (fMRI) and *magnetoencephalography* (MEG) are not practical for daily use because they require a special oversized equipment of high cost. For this reason, they are confined to the laboratory and clinical studies. *Functional near infrared spectroscopy* (fNIRS) might be a good alternative to EEG in the future, but the communication speeds of this technique are limited by the hemodynamic responses, which are slower compared to electrical signals elicited by brain processes (Nicolas-Alonso & Gomez-Gil, 2012).

Standard EEG recording systems typically require the application of conductive gel or another substance on electrodes to provide accurate measurements. This task can be tedious, and even trained technicians need between 30 and 60 minutes to prepare a full cap of 32 or 64 electrodes for a single session. For patients with ALS, this problem reduces the autonomy of the system drastically, requiring assistance for positioning the device correctly. Even for healthy people, the proper set up of a BCI session is laborious and messy. For this reason, practical BCI systems reduce the number of electrodes and sometimes replace the standard electrodes with dry sensors which do not require conductive gel. These new systems reduce the preparation time significantly, however, dry electrodes are not comfortable, and it is still exhausting to wear an EEG cap for several hours. Perhaps this is the most elementary challenge of EEG-based BCIs, but this problem has persisted since the first developments and probably will continue for the next years.

Since the point of view of software design, some critical challenges in the development of brain-controlled applications are described below.

- BCIs provide information transfer rates too low for a natural interaction between the user and the controlled device, which means that the application must compensate these limitations

and reduce the risks for errors. A bad system design may magnify any mistake because the user needs to produce the cancel command to stop the last action.

- Brain activity tends to exhibit high variability among subjects. For this reason, the most successful strategies to decode the user's intentions require subject-specific data in the training stages instead of databases of several users. Beyond the system validation and the calibration of some parameters, it is not clear how to use the large databases registered in multiple studies to compute the classification models with the aim of reducing or eliminating the training sessions.
- Several environmental and user-related factors alter the brain activity. Noise, illumination, the time of the day, user's motivation, fatigue, etc., represent other sources of variability in the measured signals. For long-term use, the BCI needs to adapt the implemented machine learning approach to avoid a decrease in the subject's performance.
- The operation of a BCI is demanding and require a high cognitive load. Furthermore, some control paradigms rely on a constant focus on visual or auditory stimuli, which is exhausting after only a few hours of operation. The use of a BCI is by itself one factor that change the brain activity.
- Not all users can produce brain signals that the BCI can recognize and classify successfully. Such an *illiteracy* obligates to explore different control paradigms in the same application and develop new algorithms to increase the sensitivity of the system.
- A practical BCI must identify the *idle* or *rest* class, which represents when the user does not want to control the interface. In many published works, the control paradigms assumed that the user generates control signals continuously. However, in more realistic scenarios, the subject does not want to interact with the system most of the time. Unfortunately, the incorporation of this class reduces the sensitivity of the system to detect actual commands. Fazli, Danóczy, Popescu, Blankertz, and Müller (2010) observed this trade-off between the speed of the interface and sending inadvertent commands.

The proposed system addresses the described challenges, giving more attention to the adaptability, the detection of the idle class, and the control of false positives.

1.3 Research Objectives and Scope

The main goal of the presented research project is the development of a multi-paradigm BCI for controlling autonomous wheelchairs. The smart device tested in this work can navigate between places without intervention, and also accepts other high-level and low-level commands such as *"go straight"*, *"turn left"*, *"turn right"*, *"stop"*, etc. The proposed system was tested in two different scenarios, inside a building with offices and other locations, and outside in a public place.

The hybrid BCI system consists of two control paradigms for selecting orders or commands: P300 evoked potentials and steady-state visual evoked potentials. In addition, another BCI was incorporated to find errors in the P300-based interface. This document gives a detailed description of the implementation of each paradigm, and how they are combined to operate the wheelchair.

The system also includes a passive BCI for measuring cognitive states. The purpose of this subsystem is to provide a tool for evaluating certain phenomena in our experiments that can be of interest for further research. In this work, task engagement and fatigue were monitored in real-time

during the operation of the wheelchair, with the aim of finding peaks of these cognitive states and a possible correlation with the user's actions.

Moreover, the passive BCI system was also tested in a massive multimedia experiment (named *My Virtual Dream*), which consists of dynamic visual and musical elements that change according to the users' focus and relaxation levels (Figure 1.2). This application is an example of a different type of experimentation provided by BCI technology.

Figure 1.2. My Virtual Dream experiment. Four players control different elements of the interface at a time using low-cost EEG headsets.

Another goal of this project is the development of new methods for controlling false positives and classification errors in BCI systems. The acceptance of this new technology strongly depends on the ability of the BCI to interpret the user's intentions accurately. Classification errors may lead to important time losses, and eventually to the rejection of this communication channel. In this regard, it is proposed a new adaptive method for determining the optimal number of flashes needed to detect event-related potentials (ERPs) in P300-based BCIs. This algorithm helps to control the misclassification rate by performing a selection only when the probability of error is below the desired level.

The second strategy implemented to reduce errors in the interface consists of a novel classification method for SSVEP-based BCIs. This approach analyzes resting state data to determine the probability of false positives when the user is not trying to control the interface. This new algorithm is one of the most important components of the hybrid BCI because the SSVEP-based BCI is used to determine the moment when the user wants to select one option with the P300-based interface. In this way, low false positive rates in the SSVEP-based interface also lead to low error rates in the P300-based BCI.

Finally, this work also addresses the problem of auto-calibrating and adapting the P300-based BCI. The goal is to provide new tools to increase the autonomy of this kind of systems and reduce the dependency on external assistance to train or adjust the classification model. The long-term use of this technology demands the implementation of new strategies to adapt the system to different scenarios and new conditions.

1.4 Overview of the Thesis

The remainder of this document is organized in the following manner. *Chapter 2* describes the most important control paradigms for EEG-based BCIs. It also surveys some applications and the state of the art of brain-controlled wheelchairs. Then, the hardware and software components of the hybrid BCI are sketched in *Chapter 3*. It is included an overview of the autonomous wheelchair used in the online tests.

The preprocessing stage of the hybrid architecture is presented in *Chapter 4*, while *Chapter 5* gives a detailed description of the software implementation of the ERP-based BCI. This subsystem is used to detect P300 responses and error-related potentials. In the same direction, *Chapter 6* presents a novel approach for auto-calibrating and adapting the P300-based BCI.

The implementation of the SSVEP-based BCI is given in *Chapter 7*. Two classification strategies for SSVEPs are described here, one based on the multi-label classification of power values, and another based on the detection of significant increments of SSVEP features.

The passive BCI implemented to measure cognitive states is detailed in *Chapter 8*. *Chapter 9* presents the online tests of the complete hybrid BCI. And finally, *Chapter 10* overviews the main contributions of this work and presents the conclusions and some open questions.

Chapter 2

Control Paradigms in BCIs and Applications

In almost all the BCIs that have been developed for control applications, it is possible to identify four main components:

- A data acquisition device for measuring the user's brain activity.
- Software for real-time signal processing, statistical analysis, and classification of brain data. These processing stages decode the acquired signals into labels, indices or metrics that represent the user's states or intentions.
- A *control interface* which translates the results of the processing stages into semantic control signals for operating the device (Mason & Birch, 2003). This component also displays the state of the system and provides the visual and/or auditory cues that the user needs to interact with the interface.
- A device that receives the control signals and executes the corresponding tasks.

All the described elements of a brain-controlled system are designed according to the selected control paradigms, which define how the user interacts with the interface. A *BCI paradigm* specifies the brain patterns to identify, the mental tasks or external stimuli needed to produce the desired activity, the functionality of the control interface and other conditions required to operate the application. The following sections describe some commonly used approaches to developing brain-controlled systems.

This chapter ends with a review of medical BCI applications and other brain-controlled wheelchairs reported in previously published works.

2.1 Control Tasks for EEG-Based BCIs

The operation of a BCI involves a mental task or other actions that the user must perform to voluntary produce changes in brain activity. These tasks can be of many different types, including silent speech, counting, imagined movements, focused attention, relaxation, math calculations, etc. According to the method used to modulate the brain signals, the control tasks can be divided into two main categories (Moore Jackson & Mappus, 2010b; Nicolas-Alonso & Gomez-Gil, 2012):

- *Exogenous* (evoked) paradigms that demand the user's attention on a set of stimuli (visual, auditory, tactile, etc.), which produce an automatic response in the brain. These paradigms

Neurological Phenomenon	Description	Paradigm Type	Number of Choices	Transfer Rate (bits/min)
Slow Cortical Potentials	Slow positive and negative voltage polarizations in the EEG.	Endogenous	2	5-12
Sensorimotor Rhythms	Modulations of mu (7-13 Hz) and beta (13-30 Hz) rhythms as a result of motor-related tasks.	Endogenous	2, 3, 4 or 5	3-35
P300 Evoked Potentials	Positive event-related peaks because of infrequent stimuli.	Exogenous	> 5	20-25
Visual Evoked Potentials	Modulations in the visual cortex for external stimuli.	Exogenous	> 5	60-100

Table 2.1. Summary of neurological phenomena used in control paradigms of EEG-based BCI systems. Transfer rates were obtained from (Nicolas-Alonso & Gomez-Gil, 2012).

usually require minimal training, provide high bit rates, but the permanent attention to external stimuli may cause tiredness.

- *Endogenous* (self-generated) paradigms, in which the user performs a mental task that the system associates with one command or action to execute. The system can be operated without any stimulation (which is useful for users with visual and auditory problems), but the mental task can be demanding and require more training.

BCIs can also be classified according to how the system processes the brain signals into two categories (Nicolas-Alonso & Gomez-Gil, 2012): synchronous and asynchronous. *Synchronous* BCIs only allow the user to interact with the interface during predefined time windows. This strategy helps the users to concentrate their efforts to produce brain modulations for short periods of time, reducing in this way the cognitive noise and the complexity of the BCI. On the other hand, *asynchronous* BCIs process and decode the user's intentions all the time. These systems provide a more natural interaction between the user and the interface. However, they must analyze all the incoming brain data and find when the user wants to interact with the BCI, resulting in more complex BCIs.

Among the possible strategies reported in the literature, so far the most successful paradigms for EEG-based BCIs are based on the measurement and classification of the following neurological phenomena: *slow cortical potentials* (SCPs), modulations of *sensorimotor rhythms*, *P300 evoked responses* and *visual evoked potentials* (VEPs). This group of signals is summarized in Table 2.1.

2.1.1 Slow Cortical Potentials

Slow cortical potentials are positive or negative voltage shifts in the EEG that last from 300 ms to several seconds (Birbaumer, 1999). These changes are associated with excitatory (negative slow potentials) and inhibitory (positive slow potentials) activity of cortical networks. Humans can learn how to self-regulate these potentials using devices that provide a direct visual or auditory feedback of the SCP amplitudes.

Usually, SCP-based BCIs only allow selecting one option between two classes at a time. One class corresponds to positive shifts, whereas negative changes correspond to the second class. Despite this limitation, it's possible to build speller systems and other applications using this paradigm. For

instance, Hinterberger et al. (2004) described a control interface in which the alphabet is broken down into a sequence of binary selections, similar to a decision tree. The user navigates in this tree from the root to the leaves that represent the letters of the alphabet. In each selection step, the interface shows one of the partitions. Then, the user selects or rejects the displayed group by producing one of the two possible voltage shifts. If the partition is not selected, then the other one is displayed. Two consecutive rejections served for erasing the last selected letter. In this interface, it may take one minute to write one letter, but this is not a limitation for completely paralyzed patients who need a communication channel for improving their lives.

Unfortunately, this kind of BCI requires long training periods (sometimes months) and continuous technical support from familiars and caretakers (Birbaumer, 2006).

2.1.2 Modulation of Sensorimotor Rhythms

Sensorimotor rhythms are oscillations whose amplitudes vary when cerebral activity is related to any motor task (Nicolas-Alonso & Gomez-Gil, 2012). They comprise two bands in the spectrum: mu (7-13 Hz) and beta (13-30 Hz). People can modulate these rhythms without the need of actual movements. The mental rehearsal of a motor act is enough to observe changes in brain activity.

Self-control of sensorimotor rhythms is not an easy task. People tend to imagine situations or pictures related to real movements, instead of trying to reproduce a kinesthetic experience that eventually may induce the desired modulation (Neuper, Scherer, Reiner, & Pfurtscheller, 2005). The instructions provided during the training must emphasize these differences in order to achieve a good performance.

User training usually consists of a *neurofeedback* protocol in which the subject is asked to perform certain motor imagery task (for example, imaginary movements of a limb). The system provides real-time information about the user's performance after comparing the sensorimotor components against the reference activity which represents no imaginary movements (resting state). The feedback can be auditory or visual, and sometimes complex visualizations of the brain activity encourage the user to produce the modulations (Hwang, Kwon, & Im, 2009).

Unfortunately, an EEG-based BCI can only interpret a few motor imagery tasks successfully. Arm, leg, and tongue movements are the most common mental imaginations in this kind of systems. Decoding more than two classes requires high-density EEG caps with more electrodes, increasing the complexity of the interface and the processing stages.

BCIs that decode sensorimotor rhythms have been used extensively to control 1-D cursors, mouse pointers, or devices in 3-D environments. For instance, Doud, Lucas, Pisansky, and He (2011) developed a system that manipulates a virtual helicopter using six types of motor imagery tasks, including resting state.

2.1.3 P300 Evoked Potentials

The P300 evoked response is an event-related potential (ERP) elicited by improbable auditory, visual or somatosensory stimuli. It is characterized by a positive peak in the electroencephalogram that appears between 250 and 500 milliseconds after the event that triggers the potential (Polich, 2007). Typically,

this signal is stronger in electrodes placed over the parietal lobe, but it is possible to detect it in frontal and temporal areas.

This evoked response has been widely studied in neuropsychology because it is believed that it plays a significant role in cognitive processes such as attention and working memory (Linden, 2005). In addition, its dysfunction can be an important biomarker of neurological and mental disorders.

Studies that explore P300 evoked potentials usually follow the *oddball* paradigm (Picton, 1992). This experimental design consists of a stream of repetitive stimuli (*non-target*) that are infrequently interrupted by a deviant stimulus (*target*). Every time the target stimulus is presented, the subject most react either by pressing a button or performing a mental task such as counting. The frequency of the target stimulus has a direct influence on the component peak. The less probable is the target, the larger is the P300 response (Polich, Ellerson, & Cohen, 1996).

P300-based BCIs follows a similar principle. Randomly flashing characters, syllables, words or icons are shown on a computer screen while the user focuses attention on one of the items. In this paradigm, the subject increases an internal count or perform another task every time that the selected item is highlighted. This elicits a P300 signal synchronized with the flickering of the target element. Since the non-target stimuli do not cause evoked potentials, the system only needs to find the P300 component and identify which item triggered it. Figure 2.1 shows an example of average signals calculated for target and non-target elements in a BCI experiment. Only the target stimulus produced the evoked response.

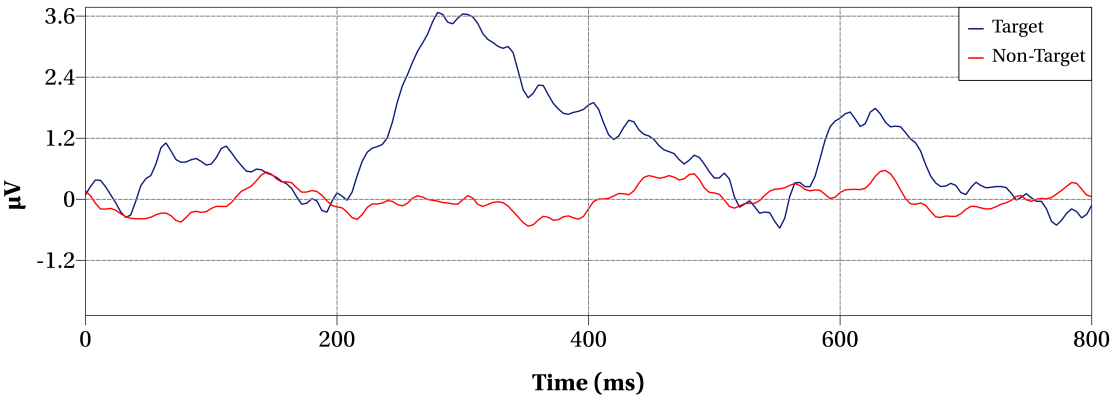


Figure 2.1. Average potentials measured at Fz for the target (in blue) and non-target (in red) options in a P300-based BCI. The peak of the evoked potential in this channel appeared 300 ms after the stimulus presentation (0 ms). Only the target stimulus elicited the evoked potential.

P300 spellers follow the original design proposed by Farwell and Donchin (1988). In this system, the alphabet and other characters are arranged in a square matrix whose rows and columns are highlighted randomly (Figure 2.2). The intersection of row and column with the highest P300 potentials indicates the target character.

Newer P300 spellers improve the precision of the system by reducing perceptual errors that happen as a consequence of adjacent rows and columns to the target character. To this end, Fazel-Rezai and Abhari (2009) proposed a new distribution of the alphabet called region-based P300 speller, in which the selection is done at multiple levels (Figure 2.3). Instead of rows and columns that flash randomly, the user selects a group of characters using the P300 system. Then, the BCI redraws the

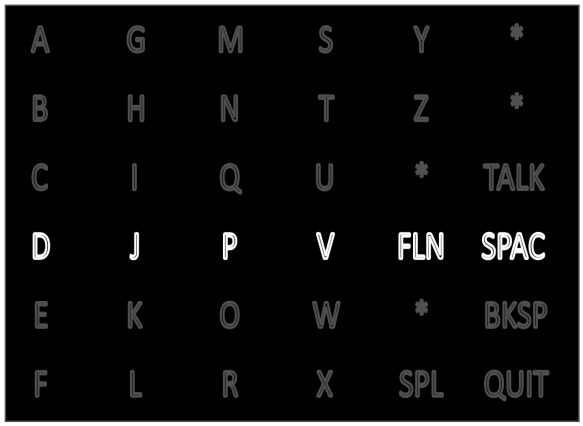


Figure 2.2. P300 matrix speller, adapted from (Farwell & Donchin, 1988). This BCI identifies the row and column that elicit the strongest evoked potentials to find the symbol selected by the user.

graphical user interface (GUI) and shows only the characters or symbols of the selected group. Finally, the user makes a second selection, but now only with a reduced number of options.

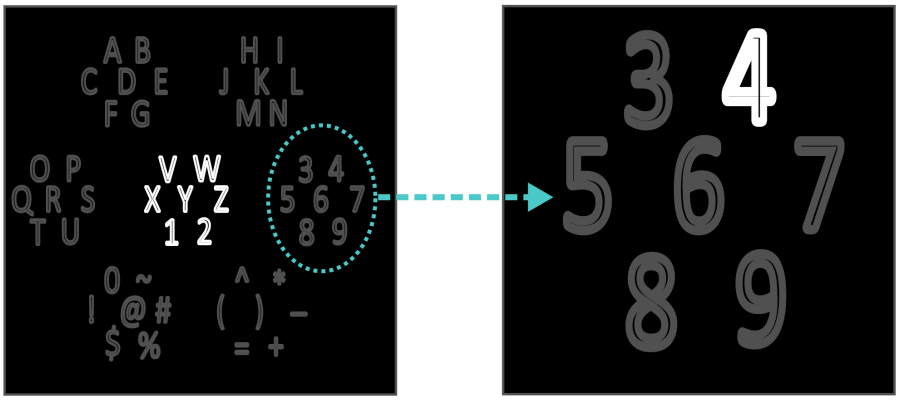


Figure 2.3. Region-based P300 speller, adapted from (Fazel-Rezai & Abhari, 2009). The character selection is made in two steps. First, the user selects the group of symbols in which the desired character belongs. Then, the interface draws the selected group, and the user makes the final choice between the displayed characters.

There are many other redesigns of the original speller matrix reported in the literature. For instance, Yeom, Fazli, Müller, and Lee (2014) modified the row-column flashing paradigm by creating random groups of characters that flash together. These groups are selected in such a way that after a certain number of flashes, all the symbols are highlighted the same number of times. In the same work, the presentation of the stimuli was modified, alternating faces and symbols instead of the standard flickering. On the other hand, other researchers have explored auditory stimuli for people with visual impairment (Furdea et al., 2009).

Two characteristics of P300-based BCIs are valuable assets: they do not require long training periods, and the interface design exclusively limits the number of possible choices. These features were the driving force to propose a BCI implementing a paradigm based on P300 evoked responses.

2.1.4 Visual Evoked Potentials

Visual evoked potentials are systematic changes in brain activity in response to a change in what a person is viewing (O'She, Roeber, & Bach, 2009). These signals can be observed in scalp electrodes situated over the occipital lobe. Similar to other ERPs (e.g., P300), VEPs have a direct link to the events that produce the potentials and are time-locked with respect to the stimulus onset.

Flashing lights and graphical patterns such as chequerboard lattices elicit these signals without the need of a mental task or other user actions. The subject only must gaze at a particular point of the visual stimulus. If the frequency of the stimulation is below 6 Hz, the potential is known as *transient VEP* (TVEP) (Nicolas-Alonso & Gomez-Gil, 2012). On the other hand, stimulation frequencies above 6 Hz produce steady-state visual evoked potentials.

The most common stimuli used in cognitive experiments and BCIs to induce VEPs are the following (Odom et al., 2010):

- *Pattern-reversal stimuli.* The phase of a high contrast black and white pattern changes abruptly (i.e., black to white and white to black). In cognitive experiments, the standard VEP response is obtained at a rate of two reversals per second (or 1 Hz). In BCIs, the frequency usually is greater, especially if the system is based on SSVEPs.
- *Pattern onset/offset stimuli.* A black and white pattern is abruptly exchanged with a diffuse gray background. In clinical studies, the pattern is presented for 200 ms separated by 400 ms of gray background.
- *Flash stimulus.* Flashing screens, stroboscopic lights, LEDs, LCD monitors or other devices can be used to induce the potential. The rate should be 1 Hz for medical applications.

SSVEPs tend to exhibit steady-state components that can be observed in the electroencephalogram. For instance, the corresponding response of a flashing stimulus resembles a sinusoidal-like waveform, whose fundamental frequency is the same as the blinking frequency (Zhang et al., 2014). The amplitude remains almost constant without phase shifts for long periods, which means that methods that analyze oscillatory activity are suitable to detect SSVEP modulations. Furthermore, it is relatively easy to select different flashing frequencies above 6 Hz for multiple stimuli. These characteristics of SSVEP-based paradigms allow designing application with more than 20 options.

A typical SSVEP-based BCI consists of flashing stimuli that represent characters, symbols or commands. The user only stares at one item to modulate the brain activity. The frequency, phase or flashing pattern varies among stimuli to generate SSVEPs with unique properties for each possible option. In this way, the system can associate the measured signals with the elements of the interface.

The visual stimuli are presented on a screen or produced with LEDs or lights. The decision of what device is preferred depends on the complexity of the design and the hardware possibilities. On the one hand, computer monitors limit the number of possible symbols because of their refresh rates, but they may provide more information and feedback to the user. On the other hand, LEDs allow to include more options, but they cannot display more information. For complex designs with more than 20 choices, LEDs usually are the best option (Zhu, Bieger, Molina, & Aarts, 2010).

The steady-state properties of the SSVEP modulations allow designing strategies that handle the idle class in such a way that users do not need to be aware of spontaneous selections when they do not want to interact with the interface. This is an important feature for the proper operation of a

wheelchair. For this reason, some functions are controlled with SSVEP modulations. This is the case of the *"stop"* command, which requires short selection times and a strict control of false positives.

2.2 Passive Detection of Cognitive Information

In the BCI paradigms described above, users consciously perform a mental task or another action that modifies the brain activity to control the interface. At all times, the behavior of the system depends on explicit orders or commands that reflect the user's intentions. However, the EEG also brings the possibility to measure and quantify mental and physiological information which cannot be chosen voluntarily. This implicit communication channel may help the system to interpret context-related information about the current interactions for a better adaptation of the machine.

Zander, Kothe, Welke, and Roetting (2008) introduced the concept of *passive BCI* to describe systems that provide implicit information without the purpose of voluntary control. In this definition, three important aspects characterize these interfaces:

- *Complementarity*. A passive BCI is complementary to other human-machine interactions without interfering their operations.
- *Composability*. An application can include many passive BCIs without conflicts between them.
- *Controlled cost*. A passive BCI does not demand conscious effort. The cost-benefit of its incorporation into the application is determined by the detection errors and the potential use of the provided information.

There are two types of cognitive information that a passive BCI can measure (Zander, Kothe, Jatzev, & Gaertner, 2010): *cognitive states* (arousal, fatigue, working load, relaxation, attention, etc.) and *cognitive events* (processing of errors, surprise, etc.). Cognitive states are aspects of conditions that are always present but vary over time, as a consequence of internal or external factors. On the other side, cognitive events are time-locked processes triggered by perceived changes in internal or external conditions.

Multiple cognitive states have been linked to the spectral properties of neural oscillations. For this reason, a common strategy used to quantify mental states consists of separating the different frequency components of the EEG (shown in Table 2.2), and compute power and connectivity features for each extracted signal (Bashivan, Rish, & Heisig, 2015). Then, a statistical model or a classification rule indicates the strength of the measured cognitive state according to the calculated values. For example, task engagement can be measured by computing the ratio of rhythms beta/(alpha + theta) (George & Lécuyer, 2010).

The detection of cognitive events depends on whether the BCI can identify clearly in the timeline the event that produces the mental change. Spontaneous activity without time information is difficult to decode because multiple brain functions may induce similar EEG activity. One example of a cognitive event that a BCI can detect successfully is the automated detection of machine errors (Zander et al., 2008). When the user perceives a wrong selection performed by the computer, the corresponding ERP is characterized by a negative peak with a latency of 350 ms followed by a positive peak 100 ms after. If this signal is detected, the passive BCI can notify to the system that the last selection is incorrect.

Band Name	Frequency Range
Delta	1-4 Hz
Theta	4-8 Hz
Alpha	8-12 Hz
Beta	12-30 Hz
Gamma	30-70 Hz

Table 2.2. Frequency bands of oscillatory brain activity.

2.3 Medical Applications of BCIs

This section describes some brain-controlled applications related to this work. In particular, medical applications of BCI's in aiding people with disabilities. A more extensive work is presented in (Moore Jackson & Mappus, 2010a).

Many BCI researchers have focused their efforts on building systems for assisting people with severe impairments. New communication and control channels which do not require peripheral nerves and muscles could potentially improve the lives of people who do not have any other way to express their needs and desires. Those who suffer from neurodegenerative diseases such as *amyotrophic lateral sclerosis* (ALS), which attacks the nerve cells responsible for controlling voluntary muscles and leaves cognitive function intact, are an important target population for this technology. In Hinterberger et al. (2004) was demonstrated that even completely paralyzed patients can achieve some degree of control of a BCI. In this work, the task consisted of moving the vertical position of a cursor freely. Likewise, Silvoni et al. (2009) evaluated a P300-based BCI with ALS patients and healthy people and proved that these interfaces are suitable for patients in early or middle stages of the disease before entering the locked-in stage.

The most known BCI applications that deal with communication and mobility disabilities typically allow selecting letters, blocks of characters, words or icons from virtual menus. The selected elements may represent any action that a computer, robot or any other device can perform. Virtual keyboards (Vos, Kroesen, Emkes, & Debener, 2014), wheelchairs (Rebsamen et al., 2007; Iturrate, Antelis, Kubler, & Minguez, 2009; Philips et al., 2007), robots (Millan, Renkens, Mourino, & Gerstner, 2004), environmental controls (Cincotti et al., 2008) and prostheses (Murguialday et al., 2007), are just a few examples of controlled devices that use this strategy.

In this category, BCI spellers are very popular and have been adapted to different technologies such as web browsers (Sirvent, Azorín, Iáñez, Úbeda, & Fernández, 2010) and speech synthesizers (Brumberg, Nieto-Castanon, Kennedy, & Guenther, 2010). Furthermore, these BCIs have been improved by adding machine learning methods and dictionaries that predict or correct the spelled words, similar to the technology used in mobiles. For example, Saa, de Pesters, McFarland, and Çetin (2015) incorporated a probabilistic model to predict the most likely responses to improve the accuracy and speed of the interface.

Brain-computer interfaces have also been tested on patients with *spinal cord injury* (SCI) and other diseases associated with loss of sensory and motor functions (Nicolas-Alonso & Gomez-Gil, 2012). Gert Pfurtscheller, Müller, Pfurtscheller, Gerner, and Rupp (2003), pioneers in the research and

development of brain-controlled neuroprostheses, built a system that combined a BCI with *Functional Electrical Stimulation* (FES) to restore movements in quadriplegic patients. FES is a technique that causes a muscle to contract through the use of electrical pulses. If the patient has muscles not paralyzed, this method can activate the nerves that produce the movements of these muscles. In this application, the BCI was used to trigger the electrical stimulations. Later, Gollee, Volosyak, McLachlan, Hunt, and Gräser (2010), Ortner, Allison, Korisek, Gaggl, and Pfurtscheller (2011) and Horki, Neuper, Pfurtscheller, and Müller-Putz (2010) developed other BCIs that substitute the traditional control interfaces in FES systems.

Other groups have explored a different approach coupling neuroprostheses and BCIs without the need of electrical stimulation. For instance, Pfurtscheller, Guger, Müller, Krausz, and Neuper (2000) tested a BCI with a tetraplegic patient to control a hand orthosis system. After an extended training period, the patient was able to open and close the device without errors. Moreover, in (Müller-Putz, Scherer, Pfurtscheller, & Rupp, 2005) the training period was reduced to a few days, giving evidence that this kind of systems can also be an alternative approach for clinical purposes.

Neurofeedback and therapy are other areas of opportunity for BCIs. Neurofeedback is a technique that teaches how to modulate the brain function consciously. It uses real-time displays and other visual and auditory feedbacks to represent specific brain activity. This method allows the patients to identify when the activity behaves similar to a desirable pattern, and then they can replicate the same in different conditions.

In this context, brain-computer interfaces have been used as complementary tools in the therapy of patients with severe post-stroke motor disability. BCIs can provide feedback of the sensorimotor cortex, helping patients to regulate activity that may reorganize the cerebral pathways needed to recover the lost functions. Following this idea, Buch et al. (2008) developed a system that helped a patient that had a whole paralyzed hand because of an injured hemisphere. In this application, a BCI controlled a hand prosthesis according to the activity produced in this hemisphere, so that the brain could associate these modulations with hand movements provided as feedback.

Neurofeedback techniques can also improve some cognitive characteristics of humans such as speech skills, affection, pain management, etc., and they have been used to treat mental disorders such as epilepsy, attention deficit, schizophrenia, depression, alcohol dependence, etc. (Nicolas-Alonso & Gomez-Gil, 2012). By combining a neurofeedback protocol with interactive games, it is possible to reach a wider audience, making the applications more attractive for the general public. Friedrich et al. (2014) developed one of the first BCI games for children with autism, giving evidence that a rich interface that combines brain, body, and behavior could be more efficient in the treatment of this condition. The application shown in Chapter 7 is another example of a rich multimedia environment that combines a BCI and a neurofeedback protocol for relaxation learning.

2.4 BCIs for Wheelchair Control

Electric wheelchairs usually are controlled with joysticks manipulated by hand or with the chin. In more extreme cases in which the person does not have remanent motor functions, eye-gaze systems or suck-and-puff controls replace the traditional input, providing at least the four commands required by the wheelchair for navigating and steering: left, right, forward and backward.

Non-invasive BCIs may represent another alternative for people with severe motor impairments to control electric wheelchairs; the major challenge is the low transfer rates in BCI's which do not allow a dexterous navigation similar to the achieved with a joystick. Even so, researchers have presented different brain-controlled wheelchairs that overcome some of the limitations by implementing two strategies: *smart navigation systems* for motion planning, path correction, and obstacle avoidance, and *control interfaces* that modify the standard commands for other options such as target places and high-level functions for semi-free navigation.

Tanaka, Matsunaga, and Wang (2005) presented one of the first BCI applications for active wheelchair navigation. In Tanaka's work, the floor was divided into rectangles of 90 cm × 60 cm from which the user selected where the wheelchair moves next. The interface was based on imagined movements of limbs (motor imagery), only allowing to select between two classes (left or right). The wheelchair always makes a step forward (Figure 2.4), compensating in this way the lack of more options. They tested the system in an experiment that consisted of reaching one target between two possibles from a starting point. Even though this BCI is highly limited and impractical, it brings evidence of the feasibility of this kind of systems.

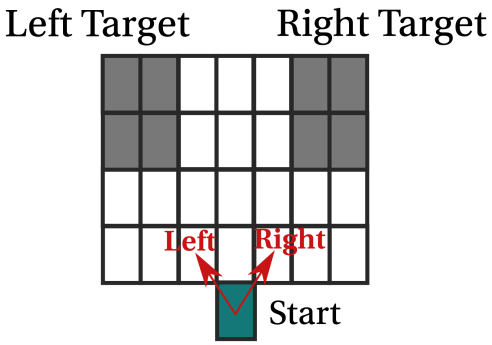


Figure 2.4. Tanaka's brain-controlled wheelchair, adapted from (Tanaka, Matsunaga, & Wang, 2005). The user selects the next square by imagining movements of limbs.

The smart navigation is an important feature that makes reliable the mental control of wheelchairs and other mobility devices. It allows dividing the responsibilities between the user, who gives high-level commands to express intentions and orders, and the wheelchair controller, which executes low-level interactions to navigate securely (Rupp, 2014). This shared control reduces significantly the amount of information that the wheelchair requires to reach a target place.

One system that implemented the shared control principle was shown by Vanacker et al. (2007). They improved the navigation by adding context-related information to filter those commands that are less probable to be selected according to the last position. The context information was obtained with laser sensors mounted on the wheelchair. Furthermore, the BCI implemented an asynchronous paradigm to detect three mental states for selecting the commands left, right and forward. Before the online operation of the wheelchair, the user was trained with multiple tasks (relax, mental left- and right-hand movements, imagined cube rotations, subtractions, and word association). After several days of training, the three tasks that reported the best results for the particular subject were used to control the wheelchair. Later, (Galán et al., 2008) presented more evidence of the robustness and reliability of this brain-controlled system performing several tests in a simulated environment.

Unfortunately, the control task of this BCI is demanding for the subject, and the options provided by the interface are limited to only three.

A newer version of the same system was described in (Carlson & del R. Millan, 2013). In Carlson's work, the shared control architecture was modified substantially by incorporating a computer vision system to detect obstacles. Two wheel encoders, sonar sensors and a monitor for real-time feedback were also included in this wheelchair. The control task was also modified. Instead of three commands, the BCI only decoded two (left and right) with the aim of reducing the cognitive load demanded to the user. If the system does not detect one of the commands, the wheelchair navigates forward. Although the shared control system proved to be more robust than the implemented in previous versions, the control task is still demanding. Also, the impossibility to stop the wheelchair voluntarily at any moment put into question the usability of this brain-controlled wheelchair.

Following a different BCI approach, Rebsamen et al. (2006) developed a P300-based brain-controlled wheelchair for navigating only along pre-defined paths. This constraint reduced the need of complicated sensors and hardware for localization, obstacle detection, and motion planning. In this application, the user selects one of the possible places shown on the graphical interface, then, the wheelchair follows the corresponding path. The interface also included the stop command while the wheelchair is moving to another place. P300 interfaces usually require several seconds to select one option; therefore, this system is highly insecure because it relies on a fast response from the subject to stop the wheelchair when an obstacle is on the guiding path.

Rebsamen et al. (2010) improved the previous design by adding a new mechanism that triggered the stop command. For this task, they tested two different approaches: a faster P300 interface, and an asynchronous BCI that detected imagined movements. This new BCI showed better response times to stop the wheelchair than the previous work, but it did not solve the problem of insecurity. Also, the P300-based BCI sometimes stopped the wheelchair involuntarily, while the detection of imagined movements required long training and was difficult to use.

Another P300-based BCI for wheelchair control was described in (Iturrate, Antelis, & Minguez, 2009). They proposed an interface that allowed the user to select where to move next between close and distant places shown on the screen (Figure 2.5). The graphical user interface shows a 3D environment with the possible destinations and other options such as turn left, turn right, validate, etc. The user first selects the next point of the trajectory, and then the wheelchair goes to the selected point after confirming this action by selecting the respective command. Once the wheelchair reaches the desired point, the user selects another one, creating in this way a path to the final destination. The wheelchair incorporated laser sensors to find obstacles and calculate the possible goals. The main problem with this system is the significant number of steps that are required to move along a complex trajectory. They reported that it took 11 minutes and nine decision steps to complete a path of 40 meters.

Recent designs implement hybrid strategies that combine more than one signal type to improve the efficiency of the interface and overcome some problems described before. For example, D. Puanhvuan and Wongsawat (2012) developed a P300-based wheelchair controller that incorporated a fast mechanism to stop the device by blinking two times. This action was also measured with the EEG. Three and four blinks were used to control other features of the BCI. Even though they reported that the detection of eye blinking is fast and reliable for this application, it is easy to produce patterns accidentally at any moment which may interpret as valid commands.

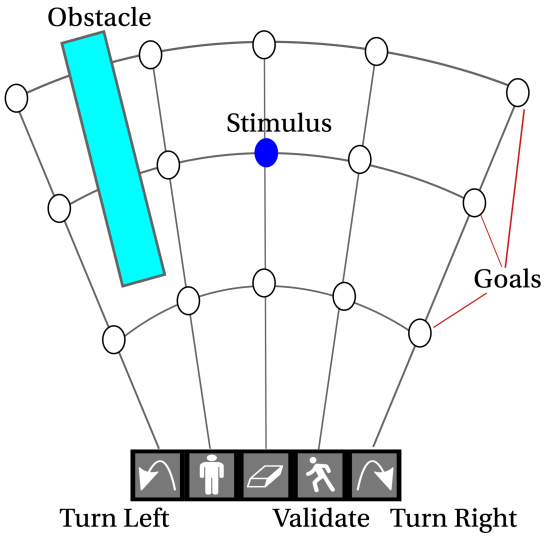


Figure 2.5. 2-D representation of the P300-based wheelchair controller implemented by Iturrate, Antelis, and Minguez (2009). The interface shows the possible goals, obstacles and other options.

Long et al. (2012) proposed another hybrid system that combined P300 evoked potentials and motor imagery. This BCI used left- and right-hand imagery to control the direction of the navigation, while imagined foot movements decreased the speed of the wheelchair. The processing of P300 evoked potentials helped the user to accelerate by selecting the corresponding command shown on the graphical user interface. Later, H. Wang, Li, Long, Yu, and Gu (2014) incorporated an electrooculogram (EOG) in this BCI to detect eye blinks used to select the commands forward, backward and stop.

In the same direction, Cao, Li, Ji, and Jiang (2014) developed a hybrid paradigm based on SSVEPs and motor imagery to control the direction and speed of a wheelchair. Motor imagery was used to select the commands *"turn left"* and *"turn right"*, while SSVEPs helped to accelerate and decelerate the device. The idle state of the motor imagery classifier indicated that the wheelchair must move forward. Other functions of the interface were selected by producing combined modulations of SSVEPs and sensorimotor rhythms.

It is important to mention that all the described brain-controlled wheelchairs are just proof of concepts, and almost all of them have been only tested under highly controlled conditions. None of the BCIs for wheelchair control have been used in daily life activities because there are several limitations in the implemented paradigms that do not allow a secure navigation and a dexterous control in all the possible environments. The risks for executing inadvertent commands is another challenge that is still open in the research community. In contrast, the hybrid control paradigm described in this work addresses the problem of misclassification by incorporating user-specific information during the online operation of the wheelchair. Additionally, this BCI provides a significant number of options for solving complex navigation tasks without limitations. This represents a significant move towards using brain imaging modalities to control smart environments securely and naturally.

Chapter 3

System Architecture

Novel developments of brain-controlled applications combine several BCIs or a BCI with devices that measure physiological processes other than brain activity (Allison et al., 2012). These new approaches distribute the functions within the interface among the BCIs, using the paradigm that fits the best in each situation. Sometimes, the input channels provide redundant information whose stability is evaluated to allow the system to select the most reliable sources of information (Kreilinger et al., 2012).

Hybrid BCIs overcome some of the limitations observed in systems based on a single paradigm (Rupp, 2014). For instance, if a P300 interface is prone to produce unintended commands, another paradigm may help to validate these decisions. For this reason, the proposed system follows the hybrid approach, combining the next four BCIs:

- A region-based (multilevel) P300-based BCI for selecting where to go and other navigation functions. The speed of this interface is not relevant because of either the execution of these commands takes much more time than the expended in producing the evoked responses, or the available options do not represent critical functions of the system.
- An asynchronous SSVEP-based BCI for choosing the *"stop"* function and unlock the P300-based interface.
- A passive ERP-based error detector for canceling the last selection of the P300-based BCI when a possible mistake has been detected. This subsystem can be enabled or disabled at any moment without interfering the functionality of other BCIs.
- A passive system for quantifying cognitive states. This BCI provides real-time feedback of a brain state (in this case task engagement) and helps to evaluate the behavior of the user during the operation of the wheelchair.

The following sections enlist the hardware and software components of the proposed brain-controlled wheelchair. Each BCI will be extensively described in the next chapters.

3.1 Autonomous Wheelchair

The autonomous device used to test the proposed hybrid BCI is based on a modified Otto Bock Xeno® electric wheelchair capable of reaching a top speed of 10 km/h (Figure 3.1). It includes a panel with buttons and a joystick that helps the user to control the direction and speed of the wheelchair.

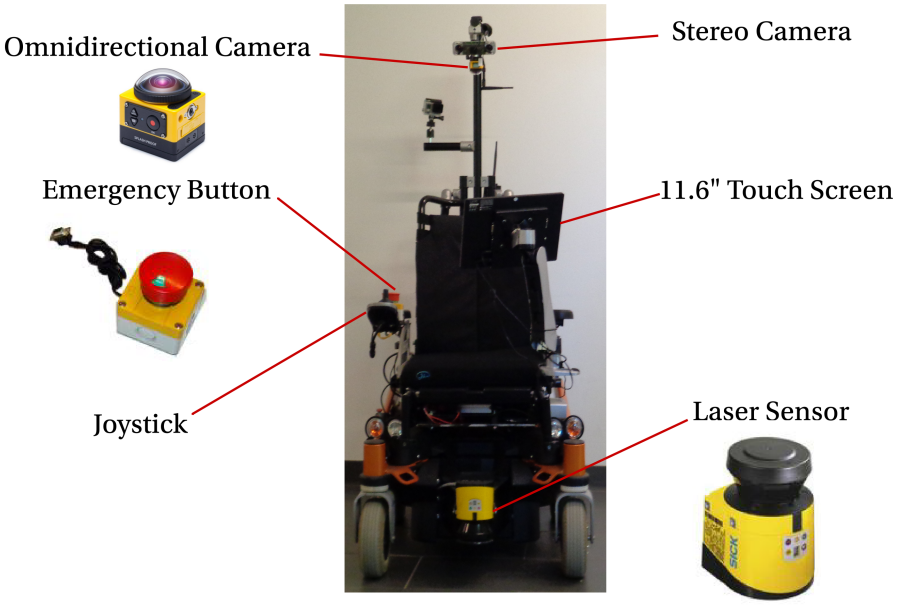


Figure 3.1. Components of the autonomous wheelchair.

The first version of this autonomous wheelchair was designed to navigate indoors to target places (Llarena & Rojas, 2016). To this end, the wheelchair controller estimates the current position in a map by integrating information from two sources: a laser sensor, which measures the distance to the surrounding objects, and wheel encoders, which sense the relative motion of the wheelchair. When a target place is selected, the wheelchair moves to that location avoiding obstacles and following a smooth trajectory. Figure 3.2 shows a portion of the 2-D map of the laboratory where the wheelchair was tested. The red circle indicates the position of the wheelchair and the blue and green lines represent the input generated by the laser sensor.

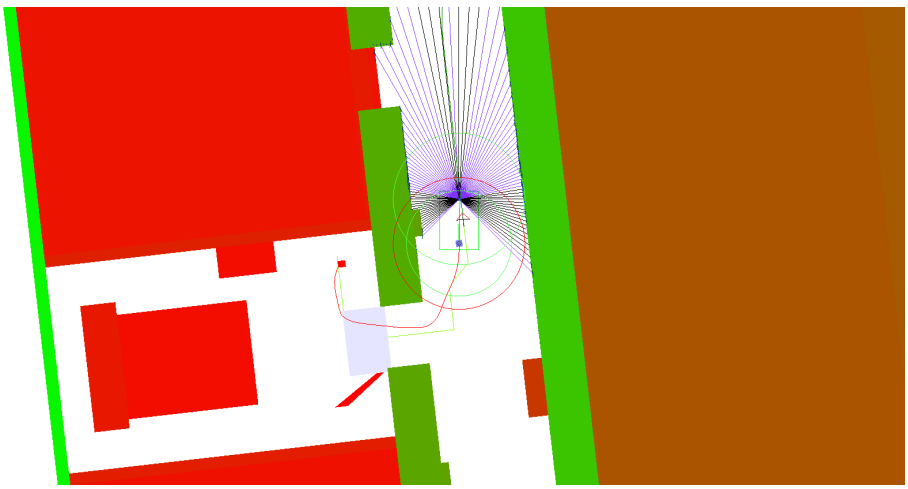


Figure 3.2. 2-D map of the building where the tests of the autonomous wheelchair were conducted. The red circle indicates the current position. The blue and green lines represent the measurements of the laser sensor.

The second version of the autonomous wheelchair supports outdoor navigation. In this case, the

laser sensor is not used because the place where the wheelchair operates has several inclinations that are difficult to handle with the 2-D measurements of only one laser rangefinder. A stereo camera, an omnidirectional camera, and a GPS help in the task of localization and obstacle detection instead.

The complete list of hardware components added to the wheelchair to provide the autonomous navigation is presented below:

- A frontal laser rangefinder with a maximum scanning distance of 60 meters and a maximum angle of 270 degrees. This sensor is mainly used indoors to localize the position of the wheelchair and detect obstacles.
- A stereo camera used to find frontal obstacles outdoors.
- An omnidirectional camera and a GPS for localizing the wheelchair outdoors.
- Two encoders that sense the relative wheelchair movements.
- A CAN bus interface to communicate with the wheelchair.
- A small-form-factor PC that processes information obtained from all the mounted input sensors and cameras to estimate the current localization of the wheelchair. This computer also calculates the navigation paths, finds possible obstacles and generates the commands that are sent to the wheelchair controller.
- An 11.6-inch touch screen used to present the visual stimuli and user feedback.

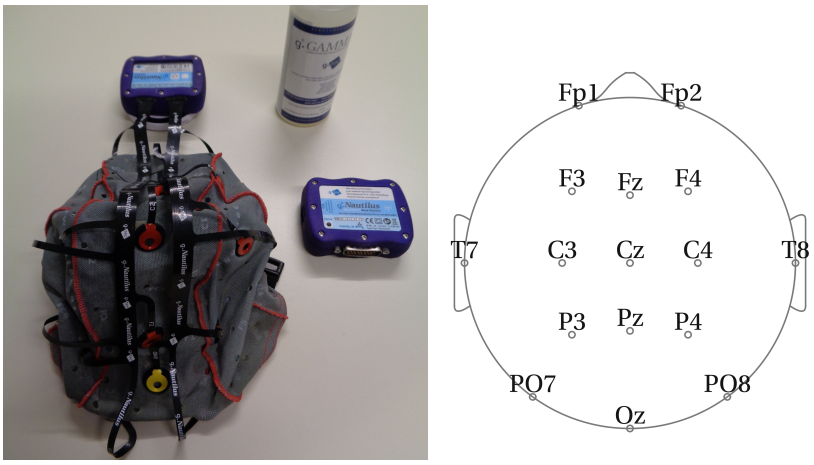
The autonomous wheelchair also includes an internal Ethernet network that brings access to the central computer. Any device connected to this network can send control commands to the autonomous system to indicate where to move next. These commands can be categorized into three types:

- High-level commands for navigating to target places (*"go to the kitchen"*, *"go to the front door"*, etc.) or along pre-defined paths (*"follow route 1"*, *"follow left path"*, etc.) autonomously.
- High-level commands for semi-free navigation. These functions indicate a direction (*"left"*, *"right"*, *"forward"* and *"backward"*) to follow until the wheelchair receives another command. This kind of control allows moving the device outside the target locations.
- Low-level commands for basic functions supported by the wheelchair (*"stop"*, *"turn left"*, *"turn right"*, *"rotate 15°"*, *"move 1 meter forward"*, etc.).

The proposed BCI allows selecting the three types of navigation commands. The purpose of this design is to cover different needs, from moving the device to fixed locations to exploring unknown places. If the environment is known, for instance, the first class of orders is suitable to navigate between locations. On the other hand, in unfamiliar places, the second and third sets are more useful for free exploration.

3.2 BCI Hardware

All the experiments in this thesis were conducted with a 16-channel g.Nautilus EEG recording system (Figure 3.3a). The sampling rate of the wireless amplifier in all the cases was set at 250 samples/second, and the high and low pass filter were set at 0.1 Hz and 60 Hz respectively. The 16 gel-based electrodes of this device have fixed locations according to the 10/20 international system (Figure 3.3b): FP1, FP2, F3, Fz, F4, T7, C3, Cz, C4, T8, P3, Pz, P4, PO7, P8, and Oz. The reference was short-circuited with the right earlobe.



(a) EEG cap and wireless amplifier. **(b)** Distribution of electrodes.

Figure 3.3. g.Nautilus EEG recording system.

A second computer installed on the wheelchair receives and processes the brain signals and generates the control commands for the autonomous wheelchair. This computer incorporates all the processing stages and the control interface. Additionally, it also receives the pose and the position of the wheelchair from the central computer. All the communication between the autonomous system and the BCI is done through the local Ethernet network of the wheelchair.

In this way, the hardware of the complete system consists of the following components:

- An Otto Bock Xeno® electric wheelchair.
- The instrumentation used for the device localization and obstacle avoidance (the laser sensor, cameras, encoders, etc.).
- A 16-channel g.Nautilus EEG recording system.
- A computer for processing the BCI code.
- Another computer for the autonomous navigation.

3.3 Processing Stages

The BCI described in this document is independent of specific hardware. It was designed to work with different data acquisition systems, and any autonomous system can be adapted to receive commands produced by the interface. The software components are flexible and allow customizing the processing stages according to the characteristics of the EEG equipment.

Typically, the operation of a BCI involves the execution of five consecutive stages (Khalid, Rao, Rizwan-i-Haque, Munir, & Tahir, 2009; Nicolas-Alonso & Gomez-Gil, 2012): signal acquisition, preprocessing, feature extraction, classification or statistical analysis of features, and control interface (Figure 3.4). All the implemented BCIs follow this schema, including the passive detection of cognitive states.

In the proposed hybrid system, the *preprocessing* stage performs the following tasks: noise reduction (signal filtering), downsampling, re-referencing, ocular and muscle artifacts suppression, and interpolation of missing data. The purpose of this stage is to enhance the signal-to-noise ratio of the EEG signals for further processing.

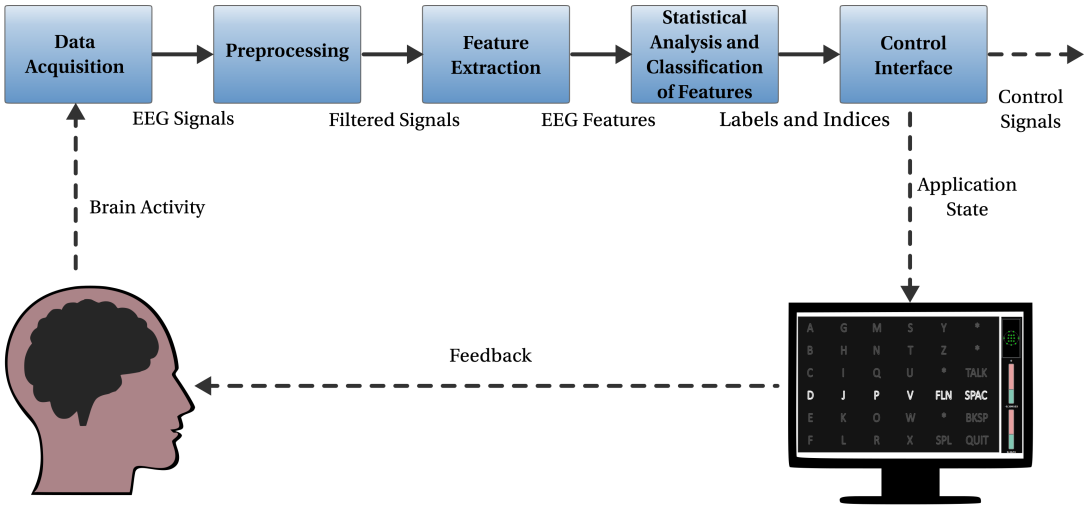


Figure 3.4. Processing stages in a BCI application.

The *feature extraction* stage obtains discriminative information from the preprocessed signals. Then, the *classification* or machine learning model evaluates these features and returns labels that represent the selected options, or indices that indicate the strength of the measured phenomena. Each BCI paradigm requires specific methods for feature extraction and classification according to the nature of the modulated signals and the operation conditions.

In the final stage, the *control interface* transforms the obtained labels and indices into control messages that are sent to the wheelchair. This component also updates the graphical user interface according to the results of the different BCIs, providing direct feedback to the user. The BCIs also receive from the control interface the state of the graphical elements in order to synchronize the EEG measurements with the flashing stimuli. With the aim of reducing the time lag between the recorded signals and the stimulus flashing, the interface uses the vertical synchronization signal (V-Sync) of the monitor to provide accurate information about the timing of these events.

The implemented BCIs work independently without interfering the functionality of each other. They only provide the results of their respective machine learning model. The control interface is the element in this hybrid architecture that makes the final decision according to what is returned in the previous processing stages. In this system, the P300-based BCI represents a binary classifier that indicates whether the last highlighted option elicited evoked responses. Likewise, the SSVEP-BCI returns if there are SSVEP modulations and which of the choices is producing them. The ERP-based error detector returns whether the last target item selected by the control interface elicited an *error-related potential* (ErrP). Finally, the passive BCI for cognitive states returns an index which represents the strength of the measured mental state.

The software architecture of the hybrid BCI is shown in Figure 3.5. The preprocessing stage is common for all the BCIs. There is not need to replicate the signal enhancement in every single paradigm. In addition, another block that detects muscle and ocular artifacts is included in the system. This tool helps the users to visualize when they are producing artifacts, so that they can learn how to reduce them when they want to operate the interface.

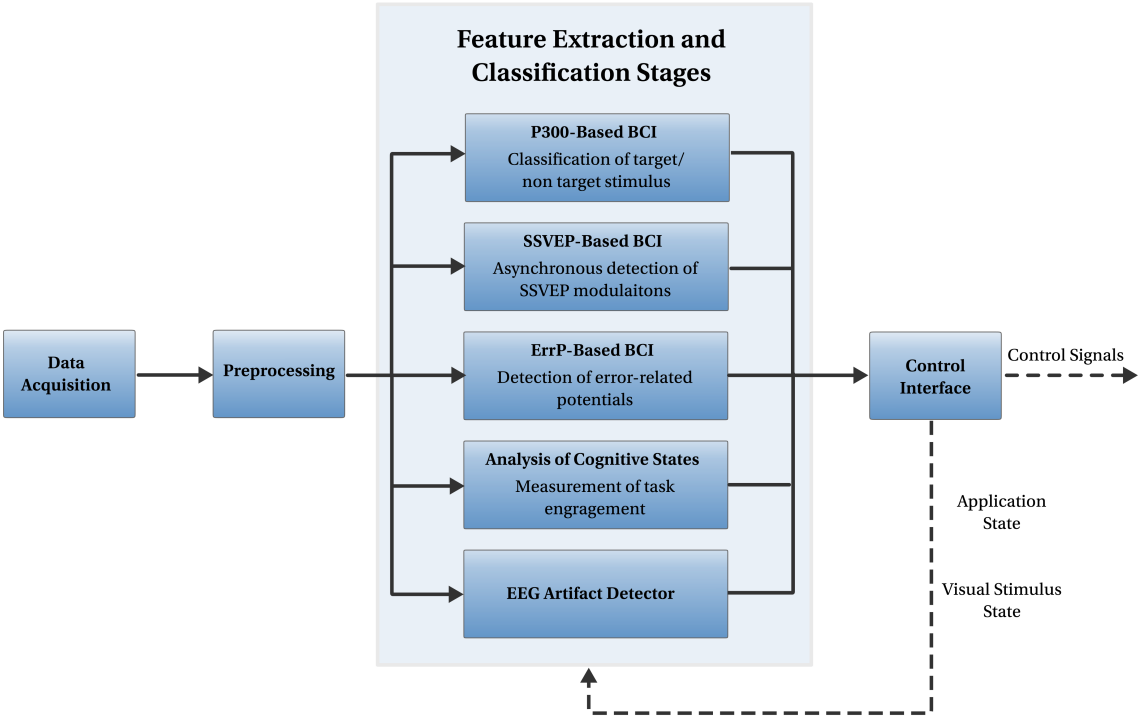


Figure 3.5. Hybrid BCI architecture.

3.4 Control Interface

The operation of the hybrid BCI involves the selection of one of the high-level or low-level commands supported by the autonomous wheelchair in two or more steps. In this architecture, the SSVEP-based BCI is responsible for providing two primary functions: the *"stop"* and *"start P300"* commands. The first option stops the wheelchair as soon as it finds a safe place. The second choice initializes the sequence of random flashes of the P300-based menu. In other words, this function unlocks the P300-based interface.

When the program starts, the P300-based interactive menu is locked. It can only be used after selecting the *"start P300"* option. Then, the flashing sequence starts and the user can choose one navigation option with the P300-based BCI. Finally, after selecting a target place, the P300-based menu is locked again, and the user can only select the options of the SSVEP-based BCI. The selection of other navigation commands does not lock the P300-based interface automatically. However, this function is always available in the interface by choosing the *"cancel"* option.

Two significant advantages of using the asynchronous SSVEP-based BCI may be mentioned at this point:

- Users can generate the stop command without the need of eliciting P300 responses. They only need to stare at the symbol that represents this option in the interface.
- The SSVEP-based BCI reduces the generation of inadvertent commands produced by the P300-based interface. As it will be shown later in this document, the asynchronous operation of a P300-based BCI has several drawbacks because many brain signals can be classified as P300

responses even when the user is not performing a mental task. However, if the P300 interface starts to operate at the moment when the user needs it, only a few flashes are required to achieve high classification rates.

When the P300-based BCI selects one option, the passive system for error detection evaluates the EEG responses and returns whether it found an ErrP. If this is the case, the last action is canceled without sending messages to the autonomous system. On the contrary, when this evoked response is not detected, the system executes the corresponding command.

The graphical user interface of the hybrid BCI is shown in Figure 3.6. The right side of the GUI contains the commands of the SSVEP-based BCI (*"start P300"* and *"stop"*). This interface also provides two visual feedbacks, a vertical bar that represents the measurements of a cognitive state, and a plot that indicates the state of the EEG sensors. The rest of the elements of the interface correspond to the choices of the P300-based BCI.

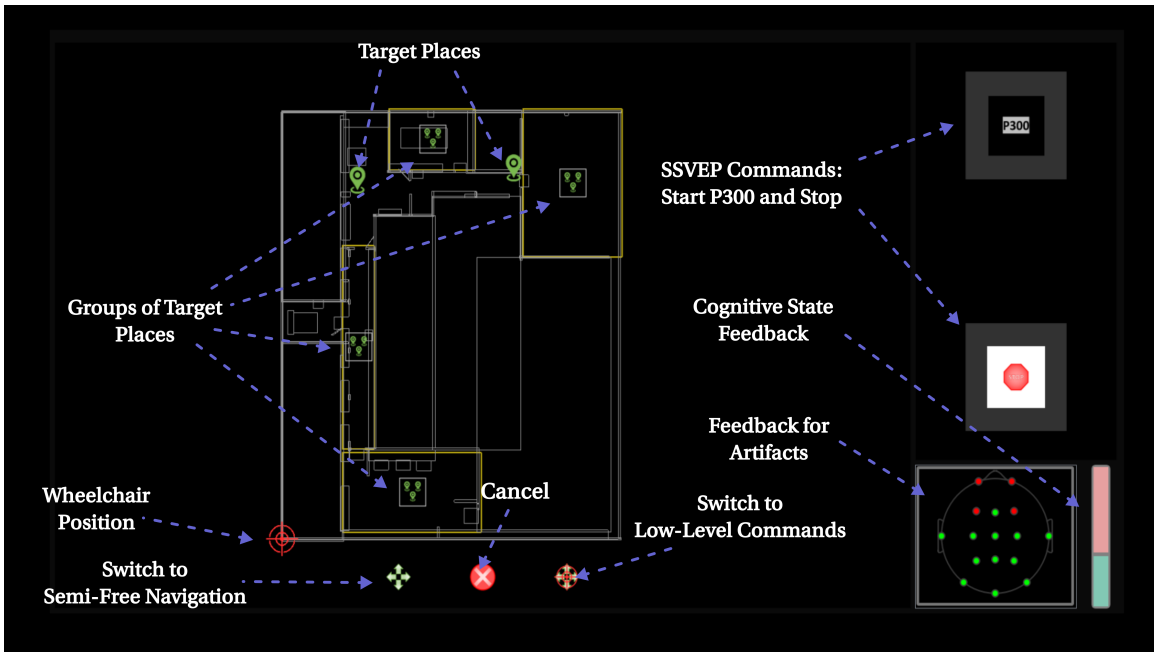


Figure 3.6. Control interface of the hybrid BCI system. The SSVEP-based BCI provides two choices: the *"stop"* command and the *"start P300"* signal. The rest of the options are selected through the P300-based BCI. The user interface also displays feedback of a cognitive state and the electrodes with muscle and ocular artifacts.

Colors of the vertical bar indicate the strength of the quantified cognitive state. When the cognitive state is low, the vertical bar is mainly red. On the contrary, the bar is green when the cognitive state is high. Moreover, sensors with green lights indicate signals free of artifacts; whereas red sensors represent electrodes with contaminated readings.

The P300-based BCI can operate in three different modes according to the category of commands to select. The interface also provides the possibility of switching between modalities by selecting the corresponding graphical icon.

The *multilevel-map* (Figure 3.6) allows selecting one of the possible target destinations reachable by the wheelchair without assistance. In this map, there are two types of choices represented in the interface, final targets or groups of target locations. If one target place is chosen, the interface sends the

respective command to the wheelchair to start the navigation. The map displays the current position of the autonomous system in real-time. Moreover, if one group is selected, the map is redrawn zooming in the chosen area and showing only the target places and groups of that section (Figure 3.7). Then, the user makes another selection but into this group. In this way, the main map does not need to contain all the possible target places. After selecting a target place, the P300-based menu is locked again, so that only the SSVEP-based BCI can generate control signals under this conditions.

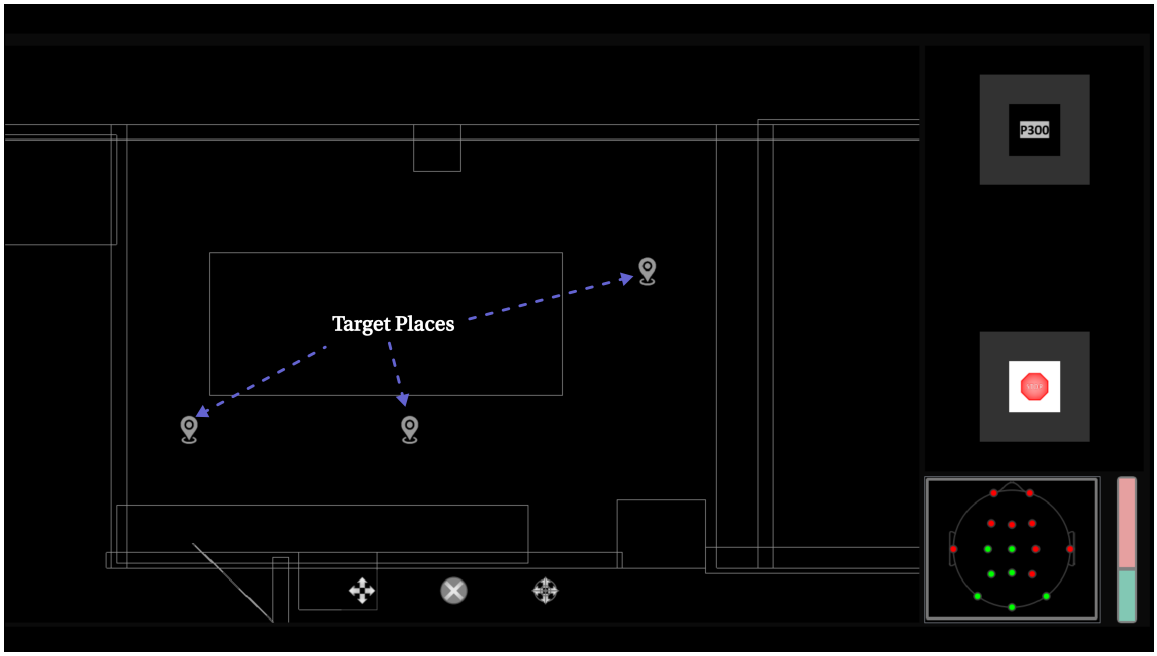


Figure 3.7. Control interface of the hybrid system when the P300-based BCI selects a group of target places. The chosen area is zooming in, showing the target locations and groups that belong to that particular section.

The second set of commands of the P300-based BCI represents high-level commands for *semi-free navigation* (see Figure 3.8). In this modality, the wheelchair follows the chosen direction avoiding obstacles until another command is sent to the controller. Finally, the third operation mode of the P300-based interface includes *low-level commands* for basic orders such as *"move 1 meter forward"*, *"turn left 45°"*, etc. The commands supported in this modality are shown in Figure 3.9. The selection of any of these options does not lock the P300-based interface. However, the *"cancel"* option or the *"stop"* command can be used to lock the interactive menu.

The system also includes another interface but controlled only with P300 evoked potentials (see Figure 3.10). In this case, the interface is always unlocked, and the *"stop"* command is selected with ERP responses. This interface is one alternative if one subject cannot control the SSVEP-based BCI. The online tests presented in this document compare the hybrid design with this alternative to highlight the advantages of each control interface.

Moreover, the system also provides active navigation controlled by SSVEP modulations (Figure 3.11). This option is used to manipulate the wheelchair freely using three low-level commands: *"forward"*, *"turn left"* and *"turn right"*. If the BCI detects that the user is gazing at one of flashing elements, the autonomous system moves forward or rotates until the subject stops focusing on the corresponding visual stimulus. If no modulations are detected, the wheelchair stands still and waits

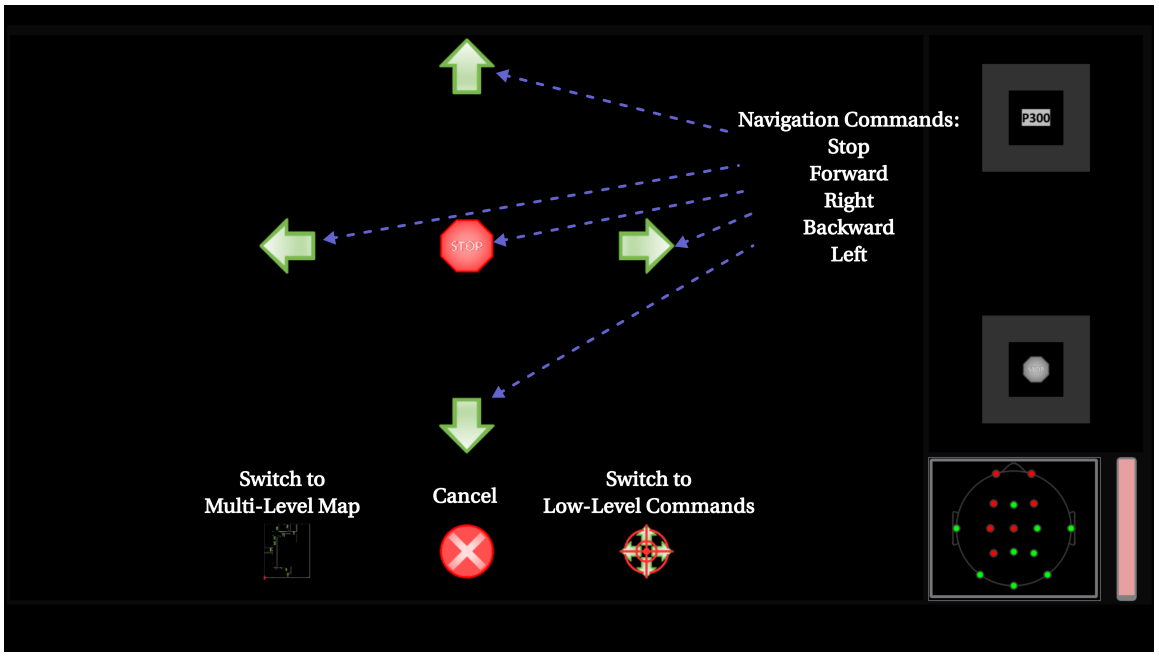


Figure 3.8. P300-based interface for semi-free navigation. Five commands are provided in this operation mode: *"forward"*, *"backward"*, *"left"*, *"right"* and *"stop"*.

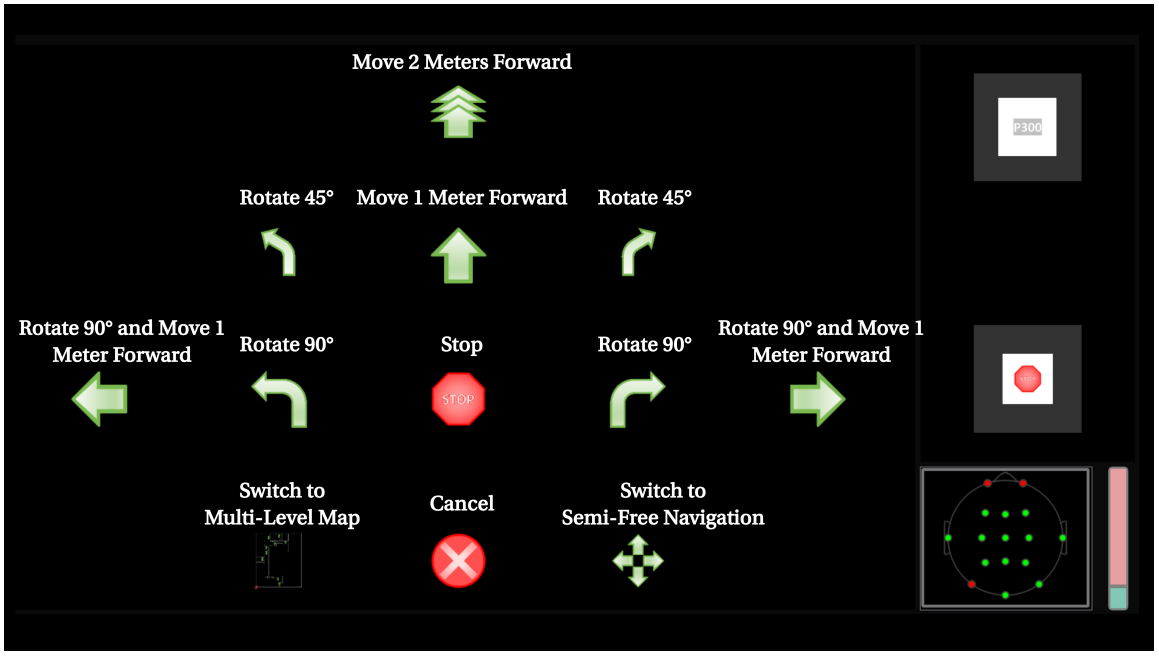


Figure 3.9. P300-based interface for low-level interactions. This interface provides basic commands for small steps and rotations.

for any other command.

For outdoor navigation, another control interface was implemented because the wheelchair is designed to reach only pre-defined target places (Figure 3.12a). In this case, the user selects the next destination through the P300-based BCI. Then, the user interface shows the options *"resume"*, *"stop"* and *"cancel"*, which can be chosen again by using the P300 paradigm (Figure 3.12b). Additionally, the

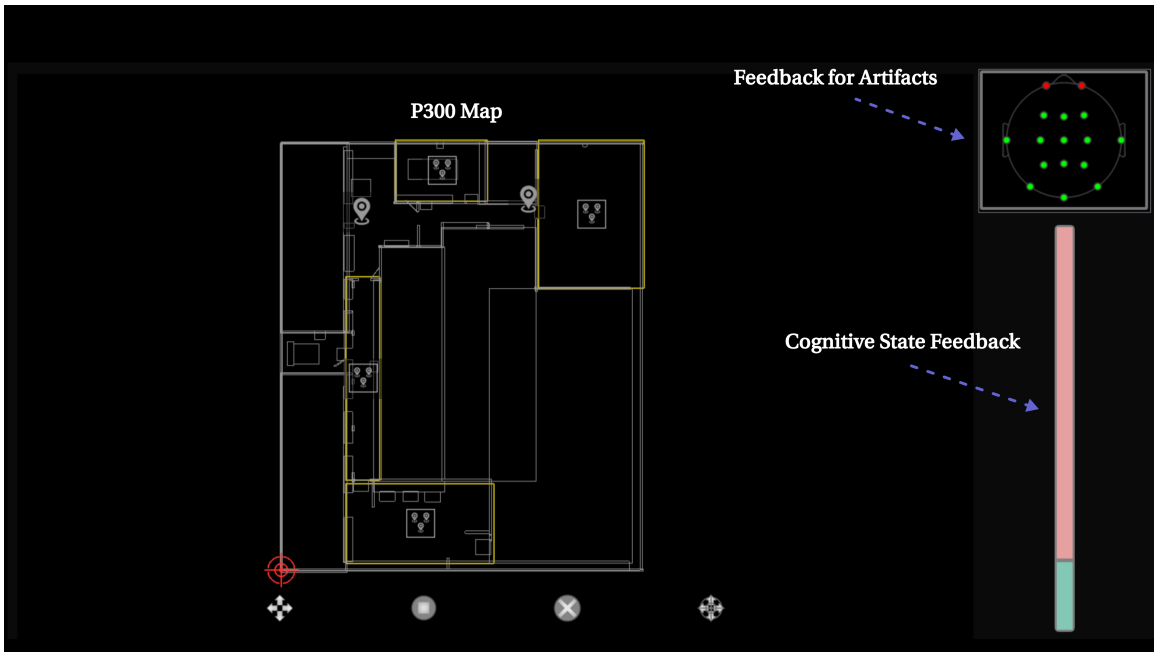


Figure 3.10. P300-based control interface. All the options are selected with P300 responses.

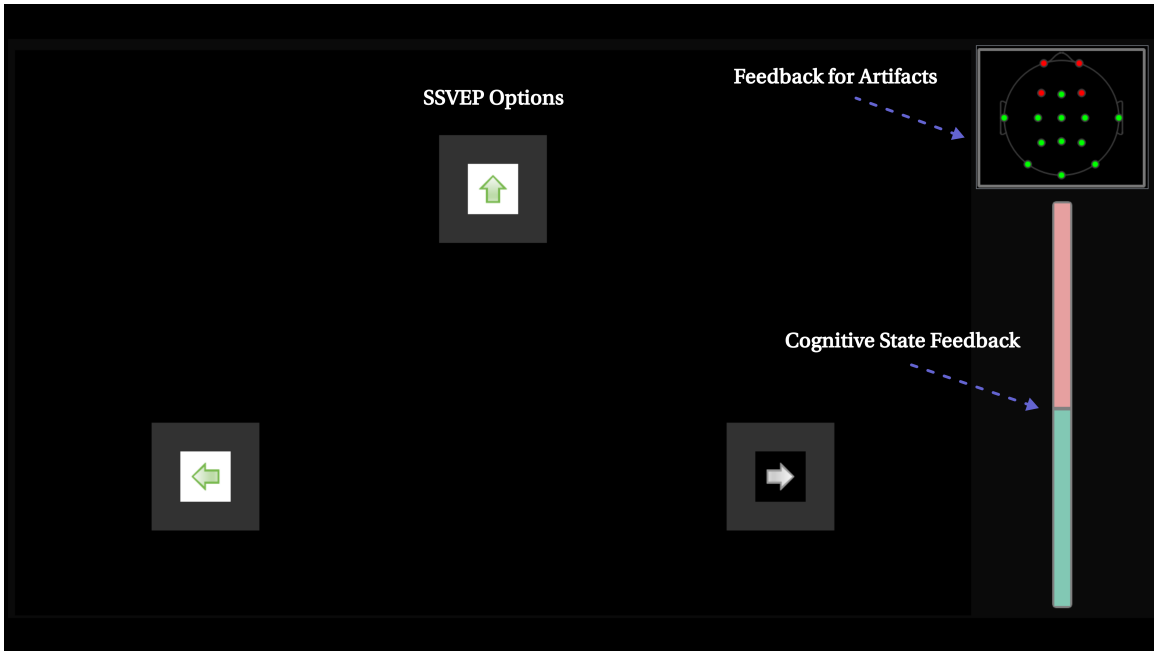
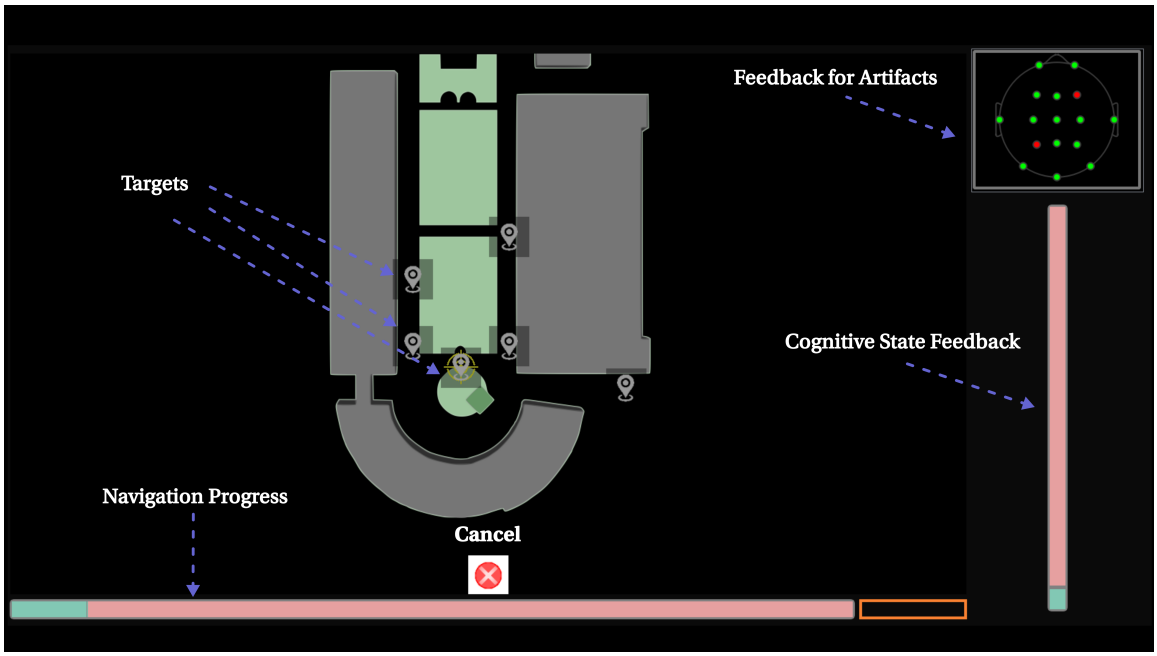


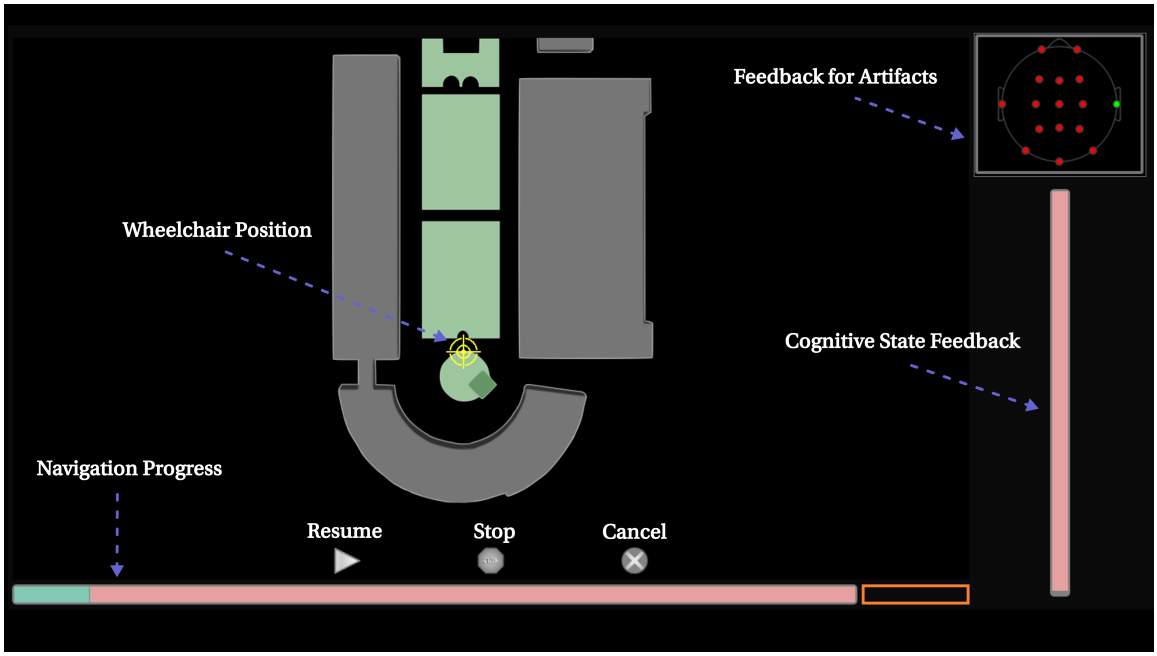
Figure 3.11. SSVEP-based control interface for active navigation. The wheelchair moves forward or rotates when the user stares at one of the flashing elements of the interface.

GUI shows the percentage of progress of the last navigation task and the current position. Unfortunately, the lighting conditions are not favorable to trigger the SSVEP modulations with the LCD screen mounted on the wheelchair. For this reason, all the functions for outdoor navigation are controlled with the P300-based BCI.

In summary, four different control interfaces were implemented to interact with the wheelchair: a hybrid interface for indoor navigation, a P300-based interface for indoor navigation, a SSVEP-based BCI for active steering, and a P300-based interface for outdoor navigation.



(a) Selection of a target place.



(b) Options shown when the wheelchair is moving to the target position.

Figure 3.12. P300-based interface for outdoor navigation. The user selects one of the target places. Then, the user interface shows the options *"stop"*, *"resume"* and *"cancel"*.

The proposed hybrid architecture allows solving complex navigation tasks without requiring a high cognitive effort. The system does not demand many orders to reach a particular place. For instance, if the final destination is close to one of the pre-defined target positions, the user can select that location from the map. Then, a few low-level commands or the semi-free navigation can help the user to move the wheelchair to the desired point. Furthermore, the low-level and semi-free navigation modes increase the freedom of the user and let the system to operate in cases when there is not an available map of the building where the wheelchair navigates.

Chapter 4

Raw Data Preprocessing

EEG signals are susceptible to different sources of noise that contribute to the variability of features calculated in the analysis of cognitive processes. Ocular movements, muscle contractions, cardiac rhythms, power line fluctuations, environmental conditions, etc., are just a few examples of factors that perturb the EEG recordings. Typically, the preprocessing stage in a BCI system suppresses some of the most known forms of noise with the aim of increasing the signal-to-noise ratio of the EEG signals and make easier the task of extracting relevant information.

Another important task of this stage is the downsampling, which reduces the dimensionality of the data. The selection of the downsampling rate depends on the bandwidth of the filtered signals. In the present work, the detection of P300 responses requires a filter with a bandpass of 4-14 Hz, meaning that a rate of 4 is suitable for a data acquisition device that samples at 250 Hz. For SSVEP detection, higher frequencies are necessary, so that the downsampling rate cannot be greater than 3 for the same device.

Complementary functions in the preprocessing stage that were included into the BCI system are the interpolation of missing data, re-referencing, artifact suppression, and epoch validation. Figure 4.1 shows the execution order of the processing blocks in this stage. Each block can be removed from the pipeline according to the needs of the BCIs that receive and process the filtered signals.

4.1 Interpolation of Missing Data

In acquiring data, it is possible to lose some samples of data packages when sending signals through a wireless channel. If the recording system can identify what measurements were not received successfully, a simple linear regression between the known sample points is enough to complete the signals with the aim of avoiding jumps in the timeline and provide data with a constant sampling rate. This approach works well only when a few samples are missing. On the contrary, if long data segments are lost during the transmission, it is better to reject the measured samples without completing the missing points. In this way, the BCIs do not obtain results based on unreliable time windows with unknown data.

Another common problem in the data recording is the presence of bad electrodes from which is not possible to recover meaningful information. In this case, the feature extraction and classification stages are adapted to use only the available electrodes. However, sometimes it is required to estimate

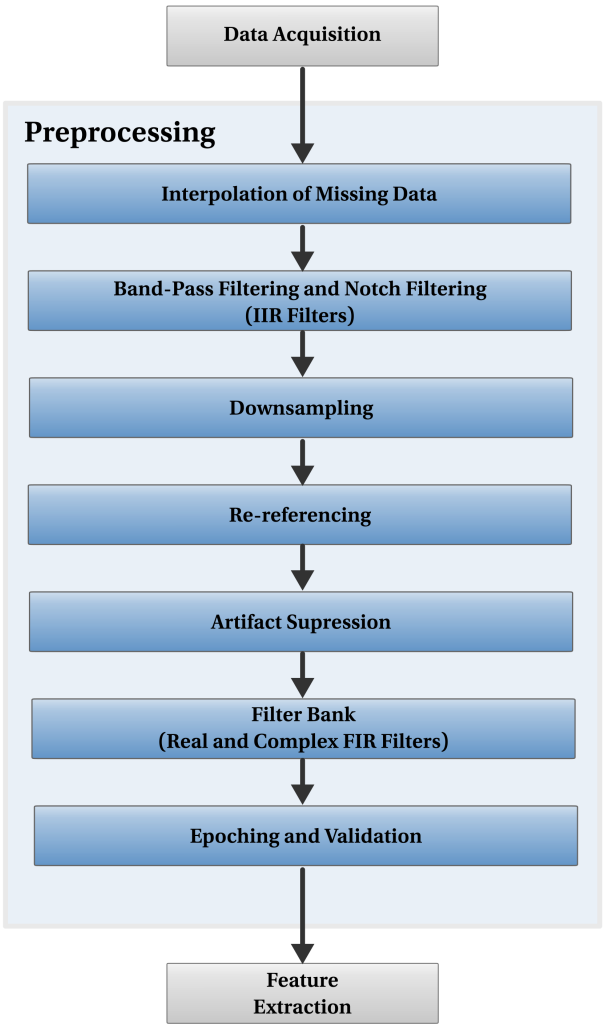


Figure 4.1. Preprocessing stage. The inclusion of each block depends on the characteristics of the recording system, the operation conditions, and the BCIs that receive the filtered signals.

the values at the missing positions using the rest of the EEG channels.

The most known interpolation methods for scalp potentials are based on continuous functions that are built using the available data. Given n observations $[v_1, v_2, ..., v_n]$ measured at distinct points $[p_1, p_2, ..., p_n]$, the problem is to find a real valued function $f(p)$ such that $f(p_i) \approx v_i$ for all $i \in \{1, 2, ..., n\}$. The regularized method of *thin-plate splines* has proved to be a reliable technique for interpolating data over the scalp, and also brings the possibility to find smooth solutions that reduce the spatial noise (Carvalhaes & Acacio de Barros, 2015).

Formally, the spline function in the three-dimensional Euclidean space has the following general form (Duchon, 1977; Roberts & Stals, 2004; Carvalhaes & Suppes, 2011):

$$f_{TPS}(p) = \sum_{i=1}^{n} c_i \left\| p - p_i \right\|^{2r-3} + \sum_{l=1}^{m} d_l \phi_l(p), \tag{4.1}$$

where the parameters r and m are integers satisfying $m = \binom{r+2}{3}$ and $m < n$; the functions $\phi_1, \phi_2, ..., \phi_m$ are linearly independent monomials in three variables of degree less than r expressed in the form

$$\phi_r(p) = x^{i-j} y^{j-k} z^k, \tag{4.2}$$

where $r = i + j + k + 1$, $0 \leq i \leq r-1$, $0 \leq j \leq i$ and $0 \leq k \leq j$.

Constants $c = [c_1, c_2, ..., c_n]'$ and $d = [d_i, d_2, ..., d_m]'$ are calculated by finding the least square solution of the linear system

$$\begin{bmatrix} K + n\lambda I & T \\ T' & 0 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}. \tag{4.3}$$

Here, the matrix A' denotes the transpose of A. The vector $v = [v_1, v_2, ..., v_n]'$ represents the instantaneous scalp potentials at the given electrode positions; K and T are matrices of dimensions $n \times n$ and $n \times m$ respectively, with $K_{ij} = \|p - p_i\|^{2r-3}$ and $T_{ij} = \phi_j(p_i)$, λ is the regularization parameter that controls the smoothness of the interpolation function. As usual, I denotes the identity matrix of suitable size.

If T is expressed in terms of its QR factorization

$$T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{4.4}$$

where $Q_1 \in \mathbb{R}^{n \times n}$, $Q_2 \in \mathbb{R}^{n \times n-m}$ and $R \in \mathbb{R}^{n \times m}$, the solution of Equation 4.3 can be expressed as follows (Roberts & Stals, 2004; Carvalhaes & Suppes, 2011):

$$c = Q_2 \left(Q_2' (K + n\lambda I) Q_2 \right)^{-1} Q_2 v, \tag{4.5a}$$

$$Rd = Q_1' (v - Kc - n\lambda c). \tag{4.5b}$$

Wahba, 1981 proposed a modification to Equation 4.1 for spherical data, replacing the Euclidean distance with the geodesic distance. The result is known as *spherical spline*, which follows the next expression:

$$f_{SPH}(p) = \sum_{i=1}^{n} c_i g_r(p, p_i) + d, \tag{4.6}$$

where

$$g_r(p, p_i) = \frac{1}{4\pi} \sum_{l=1}^{\infty} \frac{2l+1}{l^r (l+1)^r} \theta_l(\hat{p} \cdot \hat{p}_i). \tag{4.7}$$

The vectors $\hat{p}$ and $\hat{p}_i$ are unitary in the direction of p and p_i respectively; the parameter r is an integer larger than 1; and θ_l represents the Legendre polynomial of degree l. The constant values c_i's and d are again calculated by solving Equation 4.3, but now $K_{ij} = g_r(p, p_i)$, and T is a vector of ones of size n.

Unfortunately, interpolated electrodes do not provide more information to the other processing stages. They are just a weighted sum of the activity measured in other positions. Any feature calculated from these channels exhibit a spurious high correlation with characteristics obtained at other points, which reduces the rank of the covariance and correlation matrices of the calculated features. For

this reason, the BCIs are trained with only the channels that obtain good signals. Nevertheless, the interpolation of missing electrodes is useful when re-referencing methods, source reconstruction algorithms, and spatial filters require all the EEG channels, and one bad electrode may contaminate the results (Cohen, 2014b). Also, the interpolation allows comparing subjects and databases without excluding data.

4.2 Signal Filtering

In the preprocessing stage, it is possible to distinguish four types of digital filters:

- *Notch filters* for rejecting noise induced by the power line, typically tuned at 50 or 60 Hz depending on the region or country. Sometimes, the EEG recording system includes this feature, so this block is not always part of the signal enhancement stage. When required, this system uses one or more second-order IIR notch filters (C. M. Wang & Xiao, 2013) in cascade.
- *IIR band-pass filters* to reject the 0 Hz component and remove high-frequency oscillations that do not contribute with information in the feature extraction stages. *Infinite impulse response* (IIR) filters are preferred in the first filtering operations because it is possible to meet the bandwidth requirements with low-order filters. The system provides Butterworth, Chebyshev and Elliptic filters (Thede, 2004a; Podder, Mehedi Hasan, Rafiqul Islam, & Sayeed, 2014) of orders ranging from 2 to 8, which are applied in cascade with the notch filters. If the EEG device has built-in band-pass filters, this processing block is not included. Also, before any downsampling operation, a low-pass or band-pass filter is required to reduce the bandwidth of the EEG signals and avoid aliasing after the downsampling according to the Nyquist sampling theorem.
- *Band-pass FIR filters* applied in parallel after the artifact suppression block, which extract the band-limited oscillations that the feature extraction methods require. Linear-phase *finite impulse response* (FIR) filters designed in the frequency domain are used in this task. The system also provides filters that are built with the windowing technique (Thede, 2004b).
- *Quadrature FIR filters* (complex filters) to extract local properties (power and phase) of band-limited rhythms (Boukerroui, Noble, & Brady, 2004). These filters allow obtaining the time-frequency representation of the EEG signals, which is an important tool for analyzing event-related activity (Marroquin, Harmony, Rodriguez, & Valdes, 2004).

The selection of filters in the preprocessing stage depends on the rhythms and features that are extracted in the next steps. For instance, the detection of ERPs require at least one band-pass FIR filter with cut-off frequencies of 4 and 14 Hz, meanwhile the validation of epochs in the same BCI uses another filter with cut-off frequencies of 20 and 40 Hz to detect muscle artifacts.

4.3 Re-referencing

One disadvantage of the electroencephalography is that voltage measurements are relative to fixed points that can be located anywhere. The decision where to place the reference electrodes is important and has a direct effect on the recorded potentials. For instance, referencing to one lateralized site may induce bias, while references situated close to the electrodes where the main results are expected may

obscure the activity that the researcher is studying (Cohen, 2014b). Common choices for referencing are the nose, mastoids, and earlobes, but there is not a unique optimal solution that works for every situation.

It is possible to reduce the effects of the reference by subtracting the activity between pairs of electrodes, resulting in bipolar measurements. Some oscillations can be detected without problems in this modality. For example, some BCIs based on the motor imagery paradigm use bipolar signals obtained from electrodes situated over the motor cortex to detect the user's intentions. One drawback of this technique is that it essentially computes the first spatial derivative of the scalp potentials, which attenuates the activity that is common between pairs of electrodes (Dien, 1998).

Another solution to the reference problem is to transform the observations and obtain values that are free of reference. The *common average reference* (CAR) is the most known and straightforward method that provides theoretically variables with this characteristic. For each measured sample point, the new potential values are calculated in the following manner:

$$v_i^{CAR} = \frac{v_i}{\overline{v}}, \tag{4.8}$$

where

$$\overline{v} = \frac{1}{n}\sum_{i=1}^{n} v_i. \tag{4.9}$$

As before, v_i denotes the voltage value at electrode position i.

It is said that v_i^{CAR} is free of reference because $\overline{v}$ represents an estimation of the activity at the reference site which is equally represented in all the electrodes (Dien, 1998). If this potential is subtracted from all the electrodes, the resulting values are theoretically dereferenced.

The *surface Laplacian* is another technique that finds reference-free variables that can be used instead of the raw potentials. This method is a powerful technique to study EEG activity since it is related to changes in the component that is perpendicular to the scalp surface of the current density function (Carvalhaes & Acacio de Barros, 2015). A non-zero surface Laplacian indicates the presence of currents under the scalp.

If $v(\xi,\eta)$ represents the electric potentials in the parametric space $\xi\eta$ of the surface scalp, where $x = f(\xi,\eta)$, $y = g(\xi,\eta)$ and $z = h(\xi,\eta)$, the surface Laplacian at a point $p = [x,y,z]$ is defined as follows (Le, Menon, & Gevins, 1994):

$$\mathrm{Lap}_S\big(v(p)\big) = \frac{\partial^2 v}{\partial \xi^2} + \frac{\partial^2 v}{\partial \eta^2}. \tag{4.10}$$

To estimate the surface Laplacian, the standard Laplacian operator is applied over the local parametric surface around the point p, which can be done directly using finite differences or by computing the partial derivatives of interpolation functions such as the defined in Equations 4.1 and 4.6. If the thin-plate spline is expressed in its matrix form

$$v(p) = Kc + Td, \tag{4.11}$$

then the surface Laplacian is given by (Carvalhaes & Acacio de Barros, 2015):

$$\mathrm{Lap}_S\big(v(p)\big) = \widetilde{K}c + \widetilde{T}d, \tag{4.12}$$

where $\widetilde{K}_{ij} = \text{Lap}(\|p - p_i\|^{2r-3})$ and $\widetilde{T}_{ij} = \text{Lap}(\phi_j(p_i))$.

The common average reference and the surface Laplacian are tools that can improve the performance of the BCI. However, the usefulness of these techniques depends on the characteristics of the EEG equipment, the number of available electrodes, the position of the original reference, and the properties of the signals that the system is monitoring. In addition, both methods tend to attenuate the activity that is common to all the electrodes, increasing the local effects. If the BCI relies on signals that are observed all over the scalp, the original potentials are preferred. This document includes several tests to determine if the implemented BCIs obtain better classification rates when reference-free variables are processed instead of the original observations.

4.4 Artifact Suppression

Different physiological processes occurring outside the brain may have a direct influence on the EEG. In practice, the recording equipment measures any activity that induces electrical signals over the scalp. The most common sources of artifacts that can be observed in the electroencephalogram (shown in Figure 4.2) are the following (Cohen, 2014a; Urigüen & Garcia-Zapirain, 2015):

- *Cardiac activity* (ECG artifacts). The electrical activity produced by the heart can reach the scalp surface and perturb the EEG. It is characterized by regular patterns, whose amplitudes depend on the body type and the electrode positions (Urigüen & Garcia-Zapirain, 2015). Also, when electrodes are placed over arteries, it is possible to observe periodic pulses of low frequency that can be attenuated by moving the affected channels to other positions.
- *Myogenic activity* (EMG artifacts). Contracting muscles elicit electrical signals that travel through the body surface. The most common muscle artifacts happen when the subject talks, swallows, tenses the neck and walks. Commonly, the frequency components of these signals overlap with beta activity so that a significant increase in power between 20 Hz and 40 Hz may be related to this type of artifacts (Cohen, 2014a).
- *Ocular movements and blinks* (EOG artifacts). The electrical activity produced by eye movements is strong enough to be measured on the scalp, especially in frontal electrodes. Furthermore, blinking is another source of noise that produces abrupt changes in the EEG. The amplitude of these artifacts can be much greater than the typical range of the electroencephalogram.

Aiming to minimize the influence artifacts, BCI systems use to include two strategies in the preprocessing stage: the rejection of time windows with noisy data (explained in the next section), and the suppression of artifacts after separating the good EEG components from the measured signals.

Regarding artifact suppression, several methods assume that the EEG signals are a linear mixture of brain waveforms and artifacts, which are combined on the scalp. Since the point of view of the *blind source separation* (BSS) problem, the EEG measurements are given by the following expression (Stone, 2004; Christopher J James & Hesse, 2005):

$$X = AS + V, \tag{4.13}$$

where $X = [x_e(t)]_{n_e \times n_t}$ is the EEG data matrix, $e \in \{1, 2, ..., n_e\}$ and $t \in \{1, 2, ..., n_t\}$. The rows correspond

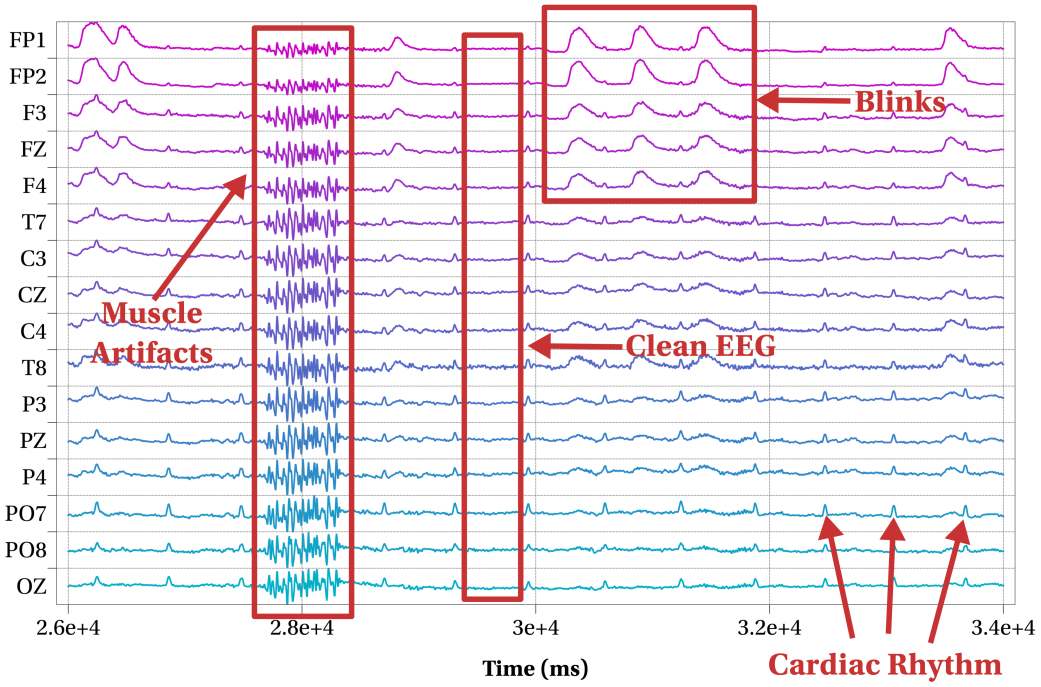


Figure 4.2. Typical artifacts that contaminate the EEG recordings. In this example, the amplitude of any contaminated time window is larger than the voltage ranges of clean EEG.

to the n_e electrode time-series, and columns are the n_t sample points or observations; A is the mixing matrix of size $n_e \times n_m$; S is the $n_m \times n_t$ matrix of unknown sources (separated brain signals and artifacts); and V is the $n_e \times n_t$ matrix of additive noise.

If it is possible to identify the artifact signals from the matrix S, new clean observations $\widetilde{X}$ can be reconstructed by projecting only brain sources back using the mixing matrix. For this operation, it can be assumed that the model is noiseless ($V = 0$) and the mixing matrix is constant over time (Urigüen & Garcia-Zapirain, 2015).

Let A_{trunc} be the truncated mixing matrix whose rows corresponding to artifact signals are set to zero, then the matrix of clean sources can be estimated by

$$\widetilde{S} = A^{+}_{trunc} X, \tag{4.14}$$

where A^{+} is the pseudoinverse matrix of A. From Equation 4.13, assuming that $V = 0$, the new clean matrix of observations is

$$\widetilde{X} = A A^{+}_{trunc} X. \tag{4.15}$$

Two BSS algorithms that have been used in EEG research to estimate the mixing model for artifact suppression are *principal component analysis* (PCA) (Ille, Berg, & Scherg, 2002; Fitzgibbon, Powers, Pope, & Clark, 2007) and *independent component analysis* (ICA) (C. J. James & Gibson, 2003; Makeig, Bell, Jung, & Sejnowski, 1996). Both methods make some assumptions about the nature of the sources, obtaining different decompositions and mixing matrices. On the one hand, PCA assumes that EEG signals and artifacts are orthogonal to each other, which is not necessary met in real conditions. On the other hand, ICA imposes that sources and artifacts are statistically independent. This assumption

is closer to the properties of some artifacts, which are not related to any monitored cognitive process. For this reason, various researchers have adopted ICA as the preferred technique for cleaning the EEG recordings (Urigüen & Garcia-Zapirain, 2015).

The automatic identification of artifact signals from the estimated sources is a laborious task. The influence of contaminant sources is not entirely well known and depends on the experimental conditions and physical characteristics of each subject. When other signals such as the EOG, EMG, and ECG are available, the proper identification of EEG sources is much easier. In this case, standard methods that compare pairs of time-series such as cross-correlation may be enough to determine artifact sources. However, when reference signals are not available, it is possible to use prototypes of artifacts that may work in certain conditions. Also, statistical measures and spectral properties of the estimated sources may help to identify the EEG components.

Mullen et al., 2013 proposed another method called *artifact subspace reconstruction* (ASR), which rejects short-time high-amplitude components that contaminate the EEG recordings. This is the technique included in the hybrid BCI to suppress artifacts. ASR finds a mixing matrix from a calibration dataset with clean EEG in resting state; which is used to estimate the principal components with a high variance that are rejected around each sample. Each clean observation is built by combining the retained PCA components of the corresponding local window and the mixing matrix that was estimated using the calibration data.

The ASR method consists of the following steps:

1. Divide the calibration dataset into epochs, and remove all the contaminated trials. The time windows of the training set are validated according to the criteria explained in the following section. Only clean data is used to train the algorithm.
2. Compute the covariance matrix C_k between channels, where $k \in \{1, 2, ..., n_k\}$, and n_k is the total number of epochs free of artifacts.
3. Calculate the mean covariance matrix C among the n_k covariance matrices. In this step, it is preferred to use a robust estimator of the mean (for example, the geometric median).
4. Compute the *singular value decomposition* (SVD) $C = VDV'$, where D is a diagonal matrix and V is an orthogonal matrix. Let $A = D^{1/2}V'$ be the mixing matrix of the training set, where $D^{1/2}$ is a diagonal matrix that contains the square roots of the entries on the diagonal of D.
5. For each column v_e in V, calculate the directional variances $s_{e,k} = v_e' C_k v_e$, , where $e \in \{1, 2, ..., n_e\}$, and n_e denotes the number of electrode positions.
6. Calculate the mean values $\mu = [\mu_1, \mu_2, ..., \mu_{n_e}]$ and the standard deviations $\sigma = [\sigma_1, \sigma_2, ..., \sigma_{n_e}]$ of the directional variances.
7. Calculate the threshold matrix $T = V\text{Diag}(\mu + 3\sigma)V'$, where $\text{Diag}(a)$ represents the diagonal matrix with the elements of vector a. The matrix T encodes the limits to accept or reject components in the local decompositions around each sample.
8. For each sample $x(t) = [x_1(t), x_1(t), ..., x_{n_e}(t)]'$ in the signal:
    a. Compute the principal components $W = [w_1, w_2, ..., w_{n_e}]$ and their respective variances $[s_1, s_2, ..., s_{n_e}]$ of the local window around the point.
    b. Calculate the directional variances $s_e^* = w_e' T w_e$. Reject all the components that $s_e > s_e^*$.
    c. Compute the reconstruction matrix R to estimate the clean sample. This matrix is obtained by rotating the matrix A into W, and truncating this rotation according to the

rejected components. If $A_W = W'A$, then

$$R = A(A_W)^{+}_{trunc}W'. \tag{4.16}$$

d. Calculate the new clean sample point $\widetilde{x}(t) = Rx(t)$.

Figures 4.3 and 4.4 illustrate one example of EEG contaminated by consecutive blinks and the results obtained with the ARS method. The cleaned signals preserve most of the information of the original data, but the high amplitude components are filtered out in the reconstructed EEG. The ARS method is useful to remove this kind of artifacts, which are common in EEG recordings regardless the quality of the acquisition system.

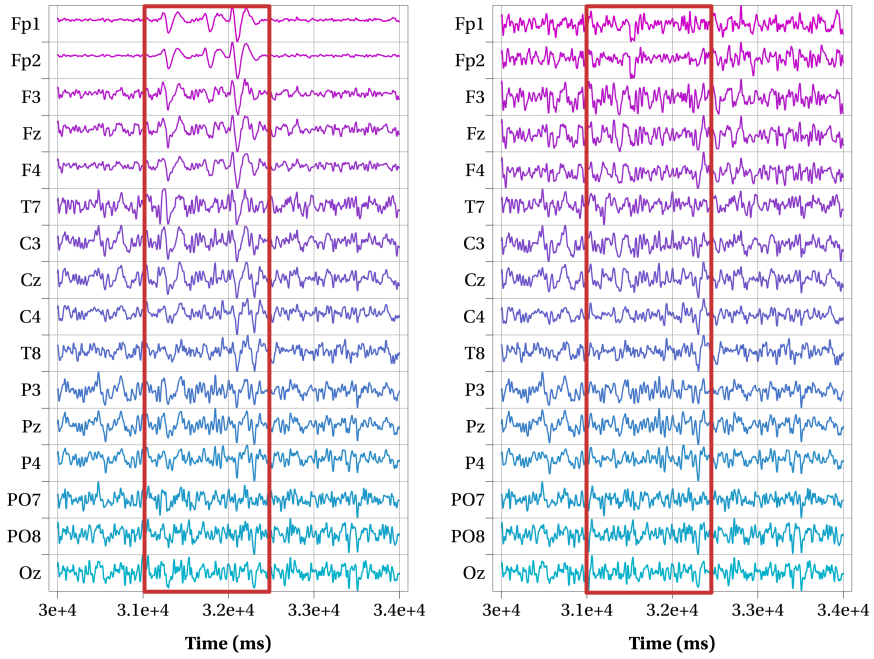


Figure 4.3. Suppression of high amplitude components using the ASR method. The plot at the left represents 4 seconds of EEG data contaminated by three consecutive blinks. The plot at the right shows the clean signals after the artifact removal. In this example, the ASR method removed the three blinks successfully.

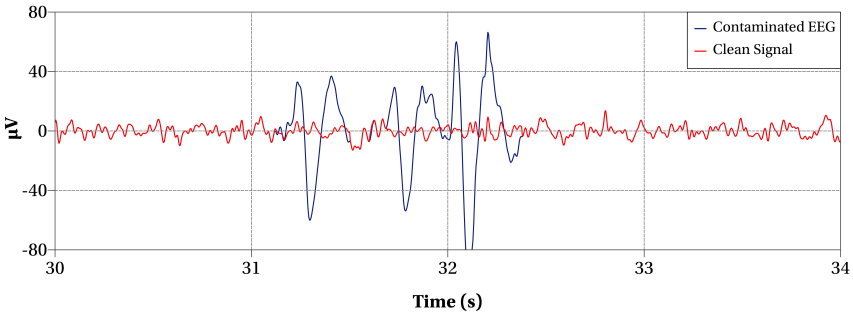


Figure 4.4. Comparison between a contaminated EEG signal and the corresponding processed data. The red line represents the clean EEG. The blue line corresponds to the original time-series. Both signals have the same amplitudes except when the subject blinks.

Feature	*Expression*
Peak-to-peak potentials	$V_e^{pp} = \max_t(x_e(t)) - \min_t(x_e(t))$.
Standard deviation	$\sigma_e = \sqrt{\frac{1}{n_t-1}\sum_{t=1}^{n_t}(x_e(t)-\mu_e)^2}$, where $\mu_e = \frac{1}{n_t}\sum_{t=1}^{n_t} x_e(t)$.
Normalized power (20-40 Hz)	$P_e^{norm} = \sum_{t=1}^{n_t}(y_e(t))^2 / \sum_{t=1}^{n_t}(x_e(t))^2$, where $[y_e(t)]$ is the filtered signal using a bandpass FIR filter between 20 and 40 Hz.

Table 4.1. Summary of features for validating epochs.

4.5 Epoch Validation

All the implemented BCIs obtain features from epochs or data segments of contiguous samples with variable length depending on the type of control paradigm. The time windows are selected according to specific time-locked events, or are chosen to represent the latest user's states. In any case, the previous step before the computation of features consists of determining whether the selected range of samples contains valid EEG data that is not drastically contaminated by artifacts or noise. In addition, some BCIs do not require a complete artifact suppression strategy, and just a previous verification of the time window to process is enough to achieve a good performance. Likewise, a second validation after the noise reduction stages may help the system to reduce errors.

In the epoch validation, each channel is tested individually. Then, the next processing stages determine whether the valid electrodes are enough to compute the required features. When possible, the characteristics are adapted to consider only those electrode positions free of noise. Otherwise, the system ignores the data and waits for another epoch.

Table 4.1 shows the variables that are computed for validating one epoch (or trial) $X = [x_e(t)]_{n_e \times n_t}$ of n_e electrodes (or channels) and n_t samples. Each feature is used to identify specific problems. For instance, peak-to-peak values that are above 200 or 300 microvolts may indicate the subject is blinking or some electrodes are placed incorrectly. Likewise, high standard deviation values reveal that the amplifier is saturated, the electrodes are not making good contact with the scalp, or there are some muscle artifacts. Finally, the normalized power values of the bandpass filtered signals are used to detect muscle artifacts.

The hybrid BCI considers that a channel e is valid only when the three described features are below certain limits. These threshold values may be calculated in advanced by using data sets recorded previously. Another option is to use the calibration routine of the BCI to obtain clean data before the online session. Furthermore, the threshold values can also be adapted dynamically during normal operation of the BCI by incorporating newly recorded signals in the estimation of thresholds.

If $Y_e = [y_e^k]_{n_k}$ are the calculated values of a feature in channel e for the n_k epochs extracted from a calibration data set, the upper threshold of the corresponding feature is obtained from

$$y_e^{max} = \min\left(B\left(Y_e\right), y_e^*\right), \tag{4.17}$$

where y_e^* is a pre-defined constant limit, and $B(Y_e)$ is the adjusted *boxplot* rule for asymmetric distributions, which is given by (Hubert & Vandervieren, 2008):

$$B(Y_e) = \begin{cases} Q_3(Y_e) + 1.5e^{3K(Y_e)}\text{IQR}(Y_e) & \text{if} \quad K(Y_e) \geqslant 0, \\ Q_3(Y_e) + 1.5e^{4K(Y_e)}\text{IQR}(Y_e) & \text{if} \quad K(Y_e) < 0. \end{cases} \tag{4.18}$$

Q_3 represents the third quartile of a set of observations; IQR is the interquartile range; and K is a robust estimation of the skewness (*medcouple*) (G. Brys, 2004).

In the proposed BCI, the predefined constant limits y_e^* were set to 200 microvolts for peak-to-peak potentials, 50 for standard deviations, and 0.5 for normalized power values.

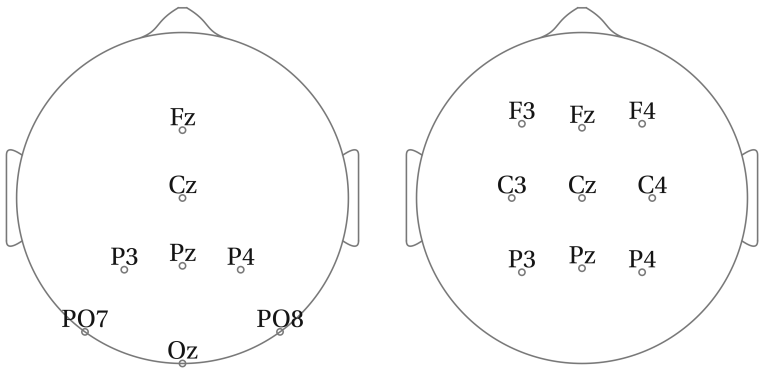
Chapter 5

ERP-Based BCI for Command Selection and Error Detection

Only BCI paradigms that allow selecting a significant number of choices can handle the wide range of options provided by the autonomous wheelchair tested in this work. This condition implies that the core functionality of the proposed hybrid architecture is based on the detection of P300 evoked potentials. The original oddball paradigm based on random flashing elements, combined with nested options, offers the possibility to include in the interface all the supported low-level and high-level commands for navigating in familiar and unknown environments.

Detecting event-related potentials mostly depends on the extraction of temporal and spatial information from the EEG activity (Fabien Lotte, 2014). Spectral features are more relevant in measuring rhythms that do not exhibit rapid changes in time. Band-pass filtering, downsampling, and spatial filtering algorithms are standard processing tools in ERP-based BCIs.

The distribution of electrodes used to identify P300 evoked responses is shown in Figure 5.1a. As mentioned in Chapter 2, these evoked potentials are stronger in electrodes located on the parietal lobe. On the other hand, error-related potentials are observed in electrodes placed on the frontocentral area (Scheffers & Coles, 2000). Figure 5.1b indicates the names of the electrodes used to classify this ERP.



(a) P300 evoked potentials. **(b)** Error-related activity.

Figure 5.1. Distribution of electrodes of the ERP-based BCIs.

This chapter describes the implementation of the ERP-based BCI implemented to detect P300

evoked responses and error-related potentials. Both cases require the same processing stages, even though they are used in different contexts. The last section includes some offline and online tests of this tool with the aim of comparing the implemented features and classification strategies.

5.1 Temporal Features for Event-Related Potentials

The analysis and classification of event-related potentials mainly depend on the identification of temporal variations after the stimulus onset. For this reason, features in several ERP-based BCIs are taken directly from the raw EEG or the preprocessed versions of these signals. A typical approach for this type of BCIs consists of classifying the band-pass filtered and downsampled signals with a linear discriminant model (Fabien Lotte, 2014). Other processing tools are spatial filters and point-wise transformations which may increase the signal-to-noise ratio of the original potentials and improve the performance of the classification model. The hybrid BCI follows all these strategies to recognize ERP components.

The feature extraction algorithms for ERP data that are presented in this document require a training set of preprocessed data $X^{training} = [X_1, X_2, ..., X_{n_k}]$, where n_k is the number of trials, and $X_k = [x_e^k(t)]_{n_e \times n_t}$ is the data matrix of the *k-th* epoch. All the trials are validated before in order to reduce the influence of artifacts. The observations are labelled for the target ($k \in K_{target}$) and non-target ($k \in K_{non\text{-}target}$) classes. In addition, epochs or trials are selected in such a way that they contain samples before and after the stimulus presentation, so that the time indices in each epoch are split into two subsets: $T_{pre} = \{-n_{pre}, ..., -2, -1\}$ and $T_{post} = \{0, 1, ..., n_{post} - 1\}$, where n_{pre} and n_{post} are the length of the pre-stimulus and post-stimulus segments respectively, and $n_t = n_{pre} + n_{post}$. The time index $t = 0$ indicates the moment when the stimulus is presented (i.e, when an item flashes on the screen in the P300 BCI, or when the interface selects one choice in the automatic detection of machine errors).

In the standard ERP analysis, trials are corrected before the analysis of the evoked responses. This operation is done by subtracting the mean values of the pre-stimulus segment, which represents the condition when the subject is only waiting for the stimulus onset. Additionally, these corrected potentials can be normalized to obtain variables less sensitive to scale variations. If $X = [x_e(t)]_{n_e \times n_t}$ is the data matrix of a single epoch, the *baseline correction* ($\tilde{X}$) and the *baseline normalization* ($\widehat{X}$) are given by the following expressions:

$$\tilde{x}_e(t) = x_e(t) - \mu_e^{pre} \quad \text{(baseline correction)}, \tag{5.1a}$$

$$\widehat{x}_e(t) = \frac{x_e(t) - \mu_e^{pre}}{\sigma_e^{pre}} \quad \text{(baseline normalization)}, \tag{5.1b}$$

where

$$\mu_e^{pre} = \frac{1}{n_{pre}} \sum_{t \in T_{pre}} x_e(t), \tag{5.2a}$$

$$\sigma_e^{pre} = \sqrt{\frac{1}{n_{pre} - 1} \sum_{t \in T_{pre}} \left(x_e(t) - \mu_e^{pre}\right)^2}. \tag{5.2b}$$

Sometimes, depending on the characteristics of the EEG equipment, it is possible to observe linear trends in the registered signals. In this case, *linear detrending* is necessary to ensure that the

baseline data is distributed around a constant value so that the post-stimulus activity can be centered correctly around zero after the baseline correction. Linear detrending is applied before the baseline correction (or baseline normalization), by computing the least-squares fit of a straight line for each epoch and channel, and subtracting the resulting predicted points to the data.

A set of features can be extracted from X, $\tilde{X}$ or $\widehat{X}$. For instance, the vector $y = [y(1), y(2), ..., y(n_c)]'$ of n_c features or characteristics for an epoch X and a downsampling rate m is obtained from the set of values $\{x_e(t) | t \in T_{post}, t \mod m = 0\}$ written in a vector form.

In a P300 experiment with flashing items, trials of the target class overlap with epochs of the non-target class, which means that it is possible to observe evoked responses in the pre-stimulus segment of trials that correspond to the latter condition. Only the pre-stimulus segment in the target class represents a right baseline in which the subject is focused without performing the mental task. For this reason, the corrected and normalized potentials were not included as features in the offline and online tests of the ERP detector.

5.1.1 Canonical Correlation Analysis Of EEG Time Series

Canonical correlation analysis (CCA) is a multivariate technique that finds linear relationships between two sets of observations. Given two matrices $P_{n\times m}$ and $Q_{n\times l}$ of paired data, whose rows correspond to observations and columns to variables, CCA seeks projections $p = [p_i]'_m$ and $q = [q_i]'_l$ that maximize the correlation between Pp and Qq. In other words, CCA finds a solution to the following optimization problem (Hardoon, Szedmak, & Shawe-Taylor, 2004):

$$\rho = \max_{p,q} \frac{p' C_{PQ} q}{\sqrt{p' C_{PP} p q' C_{QQ} q}}. \tag{5.3}$$

ρ is called the canonical correlation between P and Q for projections p and q respectively. Here, C_{XY} denotes the covariance matrix of X and Y ($\mathrm{cov}(X, Y)$); and

$$C = \mathrm{cov}\left(\begin{bmatrix} P & Q \end{bmatrix}\right) = \begin{bmatrix} C_{PP} & C_{PQ} \\ C_{QP} & C_{QQ} \end{bmatrix}. \tag{5.4}$$

The solution to the optimization problem defined in Equation 5.3 is obtained by solving a generalized eigenvalue problem (Hardoon et al., 2004), from which the projection p that maximizes ρ is an eigenvector of $C_{PP}^{-1} C_{PQ} C_{QQ}^{-1} C_{QP}$, meanwhile the projection q is given by the eigenvectors of $C_{QQ}^{-1} C_{QP} C_{PP}^{-1} C_{PQ}$. If one pair of projections cannot capture all the correlation structure of the multivariate observations, more than one pair can be selected in descending order according to the values of the corresponding eigenvalues. The greater are the eigenvalues; the greater is the correlation in the projected space.

CCA may be used to calculate linear combinations of a subset of features that maximize the correlation between the observations and the corresponding class (Fabien Lotte & Guan, 2009). This transformation computes new variables to highlight the differences between the target and non-target classes and reduce the dimensionality of the feature space. One possibility to build these subsets is to select characteristics for each electrode so that the feature vector $y(e)$ contains only the projected

timeseries of the *e-th* channel.

Let $l^{training} = [l_k]'_{n_k}$ be the column vector with the labels of the observations in the training set $Y_e^{training} = [y_e^k]'_{n_k}$, where y_e^k is the feature vector of the *k-th* epoch and *e-th* channel, $l_k \in \{1,2\}$, $l_k = 1$ for the target class, and $l_k = 2$ for the non-target class. The canonical correlation cancorr$(Y_e^{training}, l^{training})$ provides the projection w_e to compute the scalar value $\widetilde{y}(e) = w'_e y_e^k$. Only one projection for each electrode can be computed because of the dimensions of the matrix of labels. Thus, repeating this procedure for all the channels produces a new feature vector of projected data $\widetilde{y} = [\widetilde{y}(1), \widetilde{y}(2), ..., \widetilde{y}(n_e)]'$, which represents a compacted version of the original features.

A small or noisy training dataset may lead to a poorly estimation of the total covariance matrix. To improve the robustness of CCA, Fabien Lotte and Guan (2009) proposed to use regularized estimators that reduce the influence of computation errors. A general framework for regularizing the covariance matrix has the following form (F. Lotte & Guan, 2011):

$$\tilde{C} = \left(1-\gamma\right)\widehat{C} + \gamma I, \tag{5.5a}$$

$$\widehat{C} = \left(1-\beta\right)s_c C + \beta C_s, \tag{5.5b}$$

where $\tilde{C}$ is the regularized estimate, C is the original sample covariance matrix, C_s is another estimate of the covariance which incorporates a priori information of how it should be, I is the identity matrix of corresponding size, $s_c > 0$ is a scale factor, and $\beta \in [0,1]$ and $\theta \in [0,1]$ are the shrinkage parameters of the estimator.

The regularization parameter β shrinks the sample covariance matrix towards the other estimate with a priori information. It helps to reduce the risk of using the original matrix when it diverges from the expected form while the parameter γ contributes to shrinkage the estimate towards the identity, which represents the model of equal variances without covariances.

The parameters of the regularization model can be calculated by using resampling methods, cross-validations, and previous training sets. In this work, the BCI uses the method proposed by (Ledoit & Wolf, 2004) which assumes that $\gamma = 0$, $s_c = 1$ and C_s follows a constant correlation model. The advantage of this algorithm is that it finds the parameter β in a fully automatic way using only the original data.

5.1.2 Canonical Correlation Spatial Filters

Spatial filters are methods that combine information from several channels into one or more with the aim of highlighting specific properties of the original signals. They are built in such a way that they capture specific components of the EEG, improve the signal-to-noise ratio, or increase the separability between two or more conditions. The most basic form of a spatial filter is a linear combination of channels at each time point, that is to say

$$x^{sp}(t) = \sum_{e=1}^{n_e} w_e x_e(t), \tag{5.6}$$

where $x^{sp}(t)$ is the sample in the new projected space, and $w = [w_1, w_2, ..., w_{n_e}]'$ is the spatial filter (constant over time) for the n_e electrodes.

The common average reference and the surface Laplacian can be considered as examples of spatial filters since both compute new signals that are linear combinations of the original EEG channels. In spite the fact that they are tools that enhance the recorded signals significantly under certain conditions, many other options are more suitable for improving the classification stage in BCI systems. Most of these techniques require a training set to fit a model that finds discriminant information from the preprocessed data.

CCA can be used to compute spatial filters for ERP data. Let $Y(t)^{training} = [x_e^k(t)]_{n_k \times n_e}$ be the matrix with the observations at single time point $t \in T_{post}$. A spatial filter $w(t) = [w_1(t), w_2(t), ..., w_{n_e}(t)]'$ for the time point t may be obtained by computing cancorr$(Y(t)^{training}, l^{training})$. The final result is a set of time-dependent spatial filters $W = [w(1), w(2), ..., w(n_{post-1})]$ that transforms the original signals into a new time series that correlates with the corresponding class. This projection is given by

$$\widetilde{y}(t) = \sum_{e=1}^{n_e} w_e(t)\, x_e(t). \tag{5.7}$$

The previously described method (named in this document as *time-dependent canonical correlation spatial filter*) reduces the dimensionality of the original signals by obtaining only one time series. Additionally, if the data is downsampled before the computation of the spatial projections, the new features are easier to handle in the classification stage.

M. Spüler, Walter, Rosenstiel, and Bogdan (2014) proposed another way to use CCA to compute spatial filters for the detection of ERPs. Instead of finding new projected signals that correlate with the corresponding class labels, the canonical correlation is calculated between the trials of the target class and their *grand average*.

In the ERP analysis, event-related potentials are computed by averaging the preprocessed potentials across trails at each time point. This operation cancels out the noise that is randomly distributed around zero. In this way, the grand average $\overline{X}^{target}$ of the target class is given by

$$\overline{x}_e^{target}(t) = \frac{1}{n_{target}} \sum_{k \in K_{target}} \tilde{x}_e^k(t), \tag{5.8}$$

where n_{target} is the number of trials in the target class.

To compute spatial filters that increase the correlation between a new trial and the grand average $\overline{X}^{target}$, a matrix of preprocessed features X^{target} of size $(n_{target} n_t) \times n_e$ is built with data of the target class. The rows represent samples of the n_{target} trials, whereas the columns represent the n_e electrodes. The first n_t rows contain the data of the first epoch, the next n_t rows correspond to the second epoch, and so on. Then, this array is paired with a matrix Y^{target} that replicates n_{target} times the grand average values. The set of n_w spatial filters $W = [w^1, w^2, ..., w^{n_w}]$, $n_w \leqslant n_e$, is obtained by calculating cancorr$(Y^{target}, \overline{X}^{target})$. Each filter is used to obtain a new time series.

It is important to point out that in the previous method, only observations of the target class are used to find the spatial weights. The purpose of this algorithm is to increase the correlation in the projected space between epochs of the target class and the average target response. On the other hand, another alternative for obtaining spatial filters is to compute the grand average $\overline{X}^{non\text{-}target}$ for the non-target class, and build the matrices $\overline{X}^{non\text{-}target}$ and $Y^{non\text{-}target}$. Then, the filters are obtained by

calculating

$$\text{cancorr}\left(\begin{bmatrix} Y^{target} \\ Y^{non\text{-}target} \end{bmatrix}, \begin{bmatrix} \overline{X}^{target} \\ \overline{X}^{non\text{-}target} \end{bmatrix}\right). \tag{5.9}$$

These spatial filters increase at the same the correlation between epochs of the non-target class and its average response and the correlation between observations of the target class and the corresponding ERP. As will be shown later in this chapter, when the non-target class also elicits an ERP response, this method may be preferred than the spatial filter based only on target observations.

5.1.3 Fisher Spatial Filter

Given two sets X_1, X_2 with mean vectors m_1, m_2, and covariance matrices C_1, C_2, the *Fisher's criterion* is a separability measure that obtains the ratio of the between-class variance and the intra-class variance of the projected observations onto the vector w. This discriminant is defined by the following expression:

$$F(X_1, X_2) = \frac{w' C_B w}{w' C_W w}, \tag{5.10}$$

where C_B is the between classes scatter matrix, and C_w is the within classes scatter matrix. For a priori class probabilities p_1 and p_2 ($p_1 + p_2 = 1$)

$$m = p_1 m_1 + p_2 m_2, \tag{5.11a}$$

$$C_B = p_1 (m_1 - m)(m1 - m)' + p_2 (m_2 - m)(m2 - m)', \tag{5.11b}$$

$$C_w = p_1 C_1 + p_2 C_2. \tag{5.11c}$$

If $p_1 = p_2 = 1/2$, then

$$F(X_1, X_2) = \frac{\left(w'(m_1 - m_2)\right)^2}{w'(C_1 + C_2)\, w}. \tag{5.12}$$

If w is chosen so that the denominator of Expression 5.10 is $w' C_W w = 1$, the maximum separability is obtained by solving the following optimization problem:

$$F^*(X_1, X_2) = \min_w \left(-\frac{1}{2} w' C_B w\right), \tag{5.13a}$$

$$s.t. \quad w' C_W w = 1. \tag{5.13b}$$

The Lagrangian of Problem 5.13 has the form

$$L = -\frac{1}{2} w' C_B w + \frac{1}{2} \lambda \left(w' C_W w - 1\right), \tag{5.14}$$

which is reduced to a generalized eigenvalue problem

$$C_B w = \lambda C_W w, \tag{5.15}$$

whose solution is a set of eigenvectors $W = [w^k]$ and eigenvalues $\Lambda = [\lambda^k]$, whereas the maximum separability $F^*(X_1, X_2)$ corresponds to the eigenvector with the largest eigenvalue.

A set of spatial filters $W = [w^1, w^2, ..., w^{n_w}]$ can be obtained by finding the projections that maximize the Fisher's criterion for the target and non-target classes. These linear combinations increase the separability between the two sets. For this operation, the trials of both classes are concatenated in matrices X^{target} and $X^{non\text{-}target}$ of size $(n_{target} n_t) \times n_e$ and $(n_{non\text{-}target} n_t) \times n_e$ respectively. Then, the eigenvectors that optimize the Fisher's criterion $F(X^{target}, X^{non\text{-}target})$ are used as spatial filters. Additionally, the covariance matrix C_W can be estimated using the regularization method described previously.

5.1.4 xDAWN Algorithm

Another method that calculates spatial filters for the classification of ERP data is the xDAWN algorithm (Rivet, Souloumiac, Attina, & Gibert, 2009). This technique is used to enhance the ERP response to the target stimulus by estimating the channel components of the event-related potential and maximizing the signal to signal plus noise ratio. This option is very efficient in practice and may lead to a better recognition of ERP responses (Fabien Lotte, 2014). Additionally, this algorithm does not require the epoching and validation in the training stages because different sources of noise can be incorporated into the fitting process.

Given a matrix $X = [x_e(t)]_{n_t \times n_e}$ of preprocessed data with n_t sample points and n_e electrodes, if the evoked response is elicited by the stimulus presentation, the matrix of observations can be approximated by the additive model

$$X = DA + N, \tag{5.16}$$

where $A = [a_e(t_s)]_{n_{erp} \times n_e}$ represents the ERP response whose length in samples is denoted by n_{erp} (typically 600 ms or 1 second of time points); $t_s \in \{1, 2, ..., n_{erp}\}$; $N = [n_e(t)]_{n_t \times n_e}$ is a matrix with the on-going user's activity and artifacts; and $D = [d(t, t_e)]_{n_t \times n_e}$ is a Toeplitz matrix whose first column represents the time points when the target stimuli are presented (i.e $d(t,1) = 1 \quad \forall t \in T$, where T is the set of indices with the target stimulus onsets, and $d(t,1) = 0 \quad \forall t \notin T$).

The xDAWN algorithm finds a projection w that maximizes

$$J(x) = \frac{w' A' D' DAw}{w' X' X w}. \tag{5.17}$$

The numerator represents the variance of the ERP responses projected onto the new subspace, which is the signal with the relevant information that the method is enhancing. On the other hand, the numerator is the variance of the original observations (ERP signal plus noise) after the filtering. By maximizing Expression 5.17, the power of the projected ERP becomes larger whereas the power of the signal with noise decreases. The set of spatial filters $W = [w^1, w^2, ..., w^{n_w}]$ is obtained by solving the generalized eigenvalue problem associated with the optimization of $J(X)$.

One option to estimate the matrix A is to calculate the grand average of the target class. Another possibility is to find the solution of the following least squares problem (Rivet et al., 2009):

$$\widehat{A} = \arg\min_A \|X - DA\|_2^2, \tag{5.18}$$

whose solution is given by

$$\widehat{A} = \left(D'D\right)^{-1} D'X. \tag{5.19}$$

5.2 Feature Selection and Classification

The classification stage recognizes the user's intention based on the feature vector that characterizes the last epoch of EEG data. The result of this processing block is decoded in a label which represents one of the following classes: *target stimulus*, *non-target stimulus* and *not valid epoch*. If the last epoch is valid and free of artifacts, the hybrid BCI uses a binary classifier to identify whether the last stimulus elicited an evoked response (i.e. target stimulus vs. non-target stimulus). On the contrary, if the system detects that the last trail is not valid according to the criteria described in Chapter 4, the system does not decode the feature vector and assigns the third label to the last processed data.

This section describes the classification methods and feature selection algorithms included in the hybrid BCI to detect ERP responses. The last part of this chapter presents several tests for comparing the performance of these techniques in the proposed system.

5.2.1 Classification Methods for ERP Data

Linear classifiers have proved to be an efficient option for detecting ERP responses in BCI applications (Krusienski et al., 2006; Lotte, Congedo, Lécuyer, Lamarche, & Arnaldi, 2007; Nicolas-Alonso & Gomez-Gil, 2012; Fabien Lotte, 2014). They are relatively easy to train and evaluate, and require a low computational cost to classify new observations. In this category, *linear discriminant analysis* (LDA) and *support vector machines* (SVMs) are common choices for BCI researchers.

This type of classifiers makes a decision according to a linear combination of the feature vector. If x represents a real vector of n_c characteristics, the classification model evaluates the function

$$f(x) = g\left(\sum_{i=1}^{n_c} b_i x_i + c\right), \tag{5.20}$$

where $b = [b_1, b_2, ..., b_{n_c}]'$ and c are the coefficients of the linear model, and $g(a)$ is a scalar function. Then, the classification model assigns one label $l \in \{-1, 1\}$ to the given observation on the basis of the evaluation of $f(x)$. A typical approach is to use a threshold value such that values above it have class label $l = 1$. On the contrary, values below this threshold correspond to the other class ($l = -1$).

LDA finds the class l that maximizes the conditional probability $p(L = l|X = x)$ (Ng & Jordan, 2002). It assumes that the probability density functions $p(X = x|L = -1)$ and $p(X = x|L = 1)$ are both normally distributed with mean vectors m_{-1}, m_1 and covariance matrices C_{-1}, C_1. Under these assumptions, the decision rule $p(L = 1|X = x) > p(L = -1|X = x)$ is expressed as follows:

$$(x - m_{-1})' C_{-1}^{-1} (x - m_{-1}) + \ln\left(|C_{-1}|\, p(L = -1)\right) > (x - m_1)' C_1^{-1} (x - m_1) + \ln\left(|C_1|\, p(L = 1)\right), \tag{5.21}$$

where $|C|$ denotes the determinant of matrix C. The LDA model also considers homoscedasticity, which means that the covariance matrices are equal ($C_{-1} = C_1 = C$). From Equation 5.21, it can be shown that the decision rule is reduced to a dot product

$$b'x + c > 0, \tag{5.22}$$

where

$$b = 2C^{-1}(m_1 - m_{-1}), \tag{5.23a}$$

$$c = \ln\left(\frac{P(L=-1)}{P(L=1)}\right) + m'_{-1}C_{-1}^{-1}m_{-1} - m'_1C_1^{-1}m_1, \tag{5.23b}$$

and $P(L=l)$ is the probability of class l.

Without the assumption of homoscedasticity, Inequality 5.21 is reduced to the quadratic form

$$x'Ax + b'x + c > 0, \tag{5.24}$$

where:

$$A = C_{-1}^{-1} - C_1^{-1}, \tag{5.25a}$$

$$b = 2\left(m'_1C_1^{-1} - m'_{-1}C_{-1}^{-1}\right), \tag{5.25b}$$

$$c = \ln\left(\frac{|C_{-1}|P(L=-1)}{|C_1|P(L=1)}\right) + m_{-1}C_{-1}^{-1}m_{-1} - m_1C_1^{-1}m_1, \tag{5.25c}$$

Expression 5.24 is known as *quadratic discriminant analysis* (QDA). Both methods were included in the hybrid interface. Additionally, the regularized estimator of the sample covariance matrix was used to compute C_{-1} and C_1.

The *logistic classifier* is another option that has been used to identify ERP data (Seno, Matteucci, & Mainardi, 2008). This model approximates the conditional probabilities $p(L=l|X=x)$ with a logistic function

$$P(L=1|X=x) = \frac{1}{1+e^{b'x+c}}, \tag{5.26a}$$

$$P(L=-1|X=x) = \frac{e^{b'x+c}}{1+e^{b'x+c}}. \tag{5.26b}$$

The parameters of the classification model are estimated by maximizing the log-likelihood for n_k labelled observations. If $l=[l_1,l_2,...,l_{n_k}]$ represent the class labels of the training set $x=[x_1,x_2,...,x_{n_k}]$, the objective function is defined by

$$L(b,c) = \sum_{k=1}^{n_k} \ln\left[P(L=l_k|X=x_k,b,c)\right] - \lambda\|b\|_2^2, \tag{5.27}$$

where $\|a\|_2$ denotes the l^2-norm of the vector a. Here, the parameter λ penalizes large values of b.

As will be shown later in this chapter, training routines of P300-based BCIs use to produce highly unbalanced samples as a consequence of the way how the data is collected. In most cases, the number of observations available for the non-target condition is at least four or five times the number of samples of the target class. For this reason, optimizing Function 5.27 may produce a model biased towards the non-target class, since the number of terms in the summation corresponding to this category is greater than the number of terms of the target class. Under these conditions, the resulting model is useful in detecting one class but ineffective in discriminating the other one. One way to reduce the effect of an unbalanced sample is to add weights to the terms of the cost function to increase or lessen the cost of misclassifying observations of a particular group. The function to optimize is modified as follows:

$$L^*(b,c) = \sum_{k=1}^{n_k} w_{l_k}\ln\left[P(L=l_k|X=x_k,b,c)\right] - \lambda\|b\|_2^2, \tag{5.28}$$

where w_l is the weight for the observations of class l. In the BCI:

$$w_{target} = \frac{n_k}{n_{target}}, \tag{5.29a}$$

$$w_{non\text{-}target} = \frac{n_k}{n_{non\text{-}target}}, \tag{5.29b}$$

where n_{target} and $n_{non-target}$ are the number of observations of the target and non-target classes respectively.

Another linear classifier tested in the hybrid system is SVM. This method finds the optimal separating hyperplane with the largest distance (or maximum margin) to the nearest observations (Manyakov, Chumerin, Combaz, & Van Hulle, 2011). The optimal margin is the region bounded by the two parallel hyperplanes that are between the classes (see Figure 5.2). The hyperplanes can be described by the equations $b'x + c = 1$ and $b'x + c = -1$, whereas the hyperplane in the middle is defined by the expression $b'x + c = 0$. Geometrically, the margin size is given by $2/\|b\|_2$. If the training data is linearly separable, the optimal hyperplane is obtained by minimizing $\|b\|_2$ (which maximizes the margin size) and preventing that an observation falls into the margin. This method is known as *hard-margin*. The constraints of the optimization problem can be written as

$$l_k\left(b'x_k + c\right) \geq 1 \quad \forall k \in \{1, 2, ..., n_k\}. \tag{5.30}$$

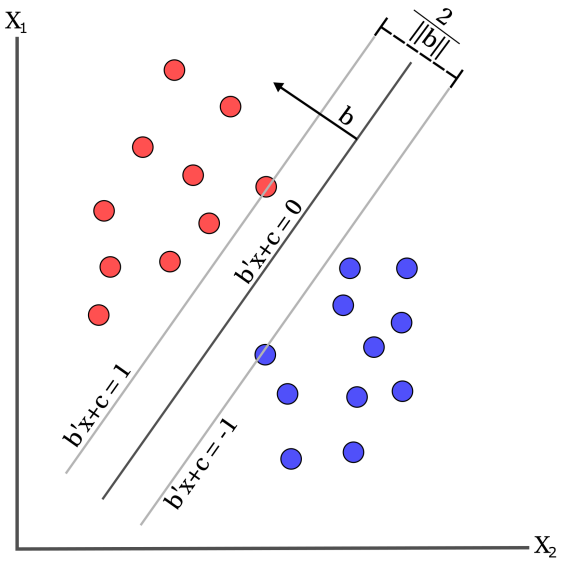


Figure 5.2. A 2-dimensional representation of the optimal separating hyperplane obtained after training a SVM classifier. The points on the margin are called support vectors.

The *soft-margin* version of this algorithm is an extension for data that is not linearly separable. It seeks a solution to the optimization problem

$$\arg\min_{(b,c)} \left[\frac{1}{n_t} \sum_{k=1}^{n_t} \max\left(0, 1 - l_k\left(b'x_k + c\right)\right) + \lambda \|b\|_2^2 \right]. \tag{5.31}$$

The regularization parameter λ is used to control the trade-off between increasing the margin and minimizing the training error. Large values of this parameter make the constraints of the optimization problem harder to ignore, whereas small values of λ tend to expand the margin allowing bigger training errors.

The SVM algorithm is prone to find biased solutions when one class has more observations than the other one. Thus, the cost function 5.31 was adapted to deal with the problem of unbalanced samples. The new function to optimize is

$$\arg\min_{(b,c)} \left[\frac{1}{n_t} \sum_{k=1}^{n_t} \max\left(0, 1 - w_{l_k} l_k \left(b' x_k + c\right)\right) + \lambda \left\| b \right\|_2^2 \right]. \tag{5.32}$$

With the aim of testing an algorithm that models non-linear relationships between features, the BCI also includes the kernel version of the SVM classifier. Kernel methods map the original observations into a higher dimensional space where linear properties are easier to generalize (Baudat & Anouar, 2000). They use a similarity function $k(x_i, x_j)$ (also known as kernel function) that can be expressed as the dot product $\langle \phi(x_i), \phi(x_j) \rangle$ of the mapped observations in the new feature space for a transformation function $\phi : \mathbb{R}^n \to \mathbb{R}^m$.

These classification methods do not require an explicit transformation of the original training set. The kernel function provides the proper mapping, which allows increasing the dimensionality of the data without computing the new feature vectors. In the projected feature space, it is possible to find a better separation rule based on a linear model. In all the experiments, the *radial basis function* (RBF) was used as the kernel function. The RBF kernel is defined by

$$k(x_i, x_j) = e^{-\gamma \left\| x_i - x_j \right\|_2^2}, \tag{5.33}$$

where γ is a parameter that controls the spread of the kernel.

5.2.2 Feature Selection and Ranking Methods

Selecting relevant variables is an important task in machine learning algorithms. A high number of features only increases the complexity of the classification method, resulting in a need of more computational resources in the offline operation and the training stages. Also, fitting a model with only a few observations may lead to overfitting and a poor generalization. In high-dimensional spaces, the available data becomes easily sparse, so that the number of required samples to estimate the parameters of the model increases exponentially.

By choosing only a small subset of attributes that contribute with discriminative information, it is possible to find models with a high predictive power that classify correctly new observations. At the same time, several features may contain noisy data or redundant information decreasing the performance of the classifier. Identifying these unnecessary features helps to increase the quality of the training set; simultaneously, it helps to understand the nature of the underlying process that generates the data.

Feature selection algorithms are usually classified in three main categories (Kumari & Swarnkar, 2011; Jović, Brkić, & Bogunović, 2015):

- *Filter methods.* Variables or subsets of features are evaluated according to a performance metric, regardless the classification or machine learning model. These metrics are developed to measure some statistical properties of the data such as correlation, similarity, information, separability, etc.
  Univariate methods of this category usually consist of ranking the features individually and selecting the best k. Then, the machine learning model is trained with the chosen variables. In this case, the scoring functions operate directly with the predictors assuming independence between variables. On the other hand, multivariate approaches include a search algorithm to generate the subsets that are evaluated in each step. In these methods, the inclusion of new features depends not only on the individual properties of each characteristic but also on the interactions between variables.
  Filters have the advantage of being fast and independent of the machine learning algorithm, but ignore the interactions between the data and the classifier (Kumari & Swarnkar, 2011). Even though they use to find a suboptimal solution, filters offer a good solution to high dimensional problems.
- *Wrapper methods.* Subsets of features are evaluated according to the performance of the machine learning algorithm. The final selection corresponds to the subset for the best model, i.e. the classifier with the lowest error rate.
  These methods demand more computation time, but they may achieve a better classification accuracy because they consider the interactions between variables and the classification model. When the number of features is considerably large, an intensive evaluation of subsets is not feasible. In this case, sequential selections and other heuristics help to reduce the number of feature subsets to evaluate.
- *Embedded methods.* The feature selection and model training are performed at the same time. Examples of this class of algorithms are classifiers that are based on the optimization of objective functions that include regularization terms, which forces some coefficients to be close to zero.

Wrappers and multivariate filters require search algorithms to generate the subsets tested in the selection procedure. These strategies avoid the exhaustive evaluation of all the possible combinations, whose number increases exponentially with the size of the feature space. The most common search strategies for feature selection algorithms are the following (Park, Koo, Kim, & Lee, 2008; Jović et al., 2015):

- *Forward selection.* The method starts with an empty feature set and then new variables are added one by one according to the improvement in the estimated model. The most promising variable is added to the feature set. The algorithm stops when the features that are not yet included do not improve the performance of the model.
- *Backward elimination.* The initial feature set contains all the variables. Features are eliminated one at the time until the model cannot be improved.
- *Stepwise selection.* The initial feature set (which may contain all the variables or no one) is modified by adding and removing features. In one step of the algorithm, the most promising variable is added into the model, and then the elements of the new set are evaluated in order to determine if it improves after removing one feature. The algorithm stops when there are no

reasons to add or remove more variables.

- *Stochastic search.* Optimization heuristics test and analyze a random sample of sets. These methods allow finding a solution close to the optimal without testing all the power set.

The proposed system includes a multivariate filter for feature selection based on the stepwise regression method. This algorithm combined with LDA has been extensively adopted by BCI researchers because of its simplicity and reliability (F. Lotte & Guan, 2011). In this method, each subset of features is used to fit a linear regression model in which the output variable is the class label. Then, the coefficient of each predictor (or member of the feature set) is tested against the hypothesis H_0 that its value is zero. The alternative hypothesis H_1 is that the coefficient is different than zero. Given a set of features $X = \{x_1, x_2, ..., x_{n_c}\}$ and two threshold values $p^{in} \in [0,1]$, $p^{out} \in [0,1]$, this stepwise selection procedure consists of the following steps:

1. Start with an empty feature set $Y_0 = \{\}$. Set counter $i = 0$.
2. For each feature x_k that is in X but not in Y_i, fit a linear regression model using the predictors $Y_k = Y_i \cup \{x_k\}$.
3. Compute the p-value p_k of each feature x_k.
4. Find the feature $\hat{x}$ with the lowest p-value $\hat{p}$. If $\hat{p} < p^{in}$, then $\hat{Y} = Y_i \cup \{\hat{x}\}$, else $\hat{Y} = Y_i$.
5. Find the predictor $\tilde{y}$ with the highest p-value in the model fitted with the feature set $\hat{Y}$. If this p-value is greater than p^{out}, then $Y_{i+1} = \hat{Y} - \{\tilde{y}\}$, else $Y_{i+1} = \hat{Y}$.
6. If $Y_{i+1} \neq Y_i$, then increase the counter i and repeat the steps 2-6.

Univariate filters were also tested in the hybrid system. These algorithms only require a metric function to score each variable and the number of features to include in the machine learning model. The selection of the number of variables may be done in advance or using a threshold value above which the best features are accepted. However, both approaches require several trials before finding a good classification model. For this reason, the BCI also includes a fully automated heuristic consisting of adding variables one by one and stopping when the classifier does not improve. This *forward selection* combines univariate filters with the wrapper strategy, which is restricted to work only with the ordered features instead of the complete set. With the aim of improving the robustness of this methodology, the system tests more than one variable to determine if they contribute to reducing the error rate. If after adding n features the model does not improve, the procedure ends and returns the last set. In the performance tests described in this document, $n = 4$.

Let $l = [l_1, l_2, ..., l_{n_k}]$ be the class labels of the training set, and $x^f = [x_1^f, x_2^f, ..., x_{n_k}^f]$ be the vector of observations of the feature f. The following metrics were implemented to evaluate the predictor f in the hybrid BCI:

- *The absolute value of the Pearson's correlation coefficient.* It measures the linear relationship between two variables. This metric is given by

$$R\left(x^f, l\right) = \left| \frac{\operatorname{cov}\left(x^f, l\right)}{\sqrt{\operatorname{var}\left(x^f\right)\operatorname{var}\left(l\right)}} \right|, \tag{5.34}$$

where $\operatorname{var}(x)$ is the variance of x, and $R(x^f, l)$ is the score of the feature f.

- *Fisher's separability.* The univariate version of the Fisher's criterion can be used to evaluate individual features. The following expression is used in this case:

$$R\left(x^f, l\right) = \frac{\left(E\left(x_{c_1}\right) - E\left(x_{c_{-1}}\right)\right)^2}{\text{var}\left(x_{c_1}\right) + \text{var}\left(x_{c_{-1}}\right)}, \tag{5.35}$$

  where $E(x)$ is the expected value of x and x_{c_l} is vector with the observations of the class l.
- *The inverse of the Davies-Bouldin index* (IBDI). This index quantifies the separability between classes by considering the mean distance of each observation to the centroid of the respective class, and the distance between the centroids of the classes (Carvalho et al., 2015). For two-class univariate problems, the Davies-Bouldin index is defined by

$$DB\left(x^f, l\right) = \frac{s_{c_1} + s_{c_{-1}}}{\left|E\left(x_{c_1}\right) - E\left(x_{c_{-1}}\right)\right|}, \tag{5.36}$$

  where s_{c_l} is the mean distance between each observation of the class l and $E(x_{c_1})$. In this way

$$R\left(x^f, l\right) = \frac{1}{DB\left(x^f, l\right)}. \tag{5.37}$$

Finally, another technique that was implemented and tested in the proposed system is the wrapper-based forward selection. In each iteration, the system tests every possible variable and keeps the one that increases the accuracy of the classifier. The stop criterion is the same as the used in the univariate filter.

5.3 Training Routines and Performance Evaluation

The P300-based BCI requires a binary classifier trained before the online operation. The estimation of this model depends on the collection of labelled data that represent the target and non-target classes. To this end, the system incorporates a calibration procedure that helps the user to understand the control task and at the same time obtains the data used to fit the machine learning model.

The training interface (shown in Figure 5.3) is a reduced version of the control interfaces described in Chapter 3. It contains only five options which represent basic commands of the wheelchair. During the calibration, the GUI indicates to the user a sequence of target commands chosen randomly. The subject is instructed to count how many times the specified target command flashes. The calibration routine (shown in Figure 5.4) is described as follows:

1. The interface highlights one of the options (chosen randomly) for five seconds.
2. The user waits five seconds until the flashing sequence starts.
3. The graphical items flash randomly for 30 seconds one at the time. Each flashing consists of highlighting one item for 75 milliseconds and turning off all the choices for 75 milliseconds. When an item is highlighted, the background changes to green, whereas the background of the non-highlighted elements is black. The same option is not highlighted twice consecutively. In this stage, the subject counts the number of times that the specified element flashes.
4. The user waits five seconds for another target chosen by the system.

This procedure is repeated eight times, giving a total training time of about 5 minutes and 20 seconds. The target elements flash around 280 times, and the non-target items are highlighted 1120 times.

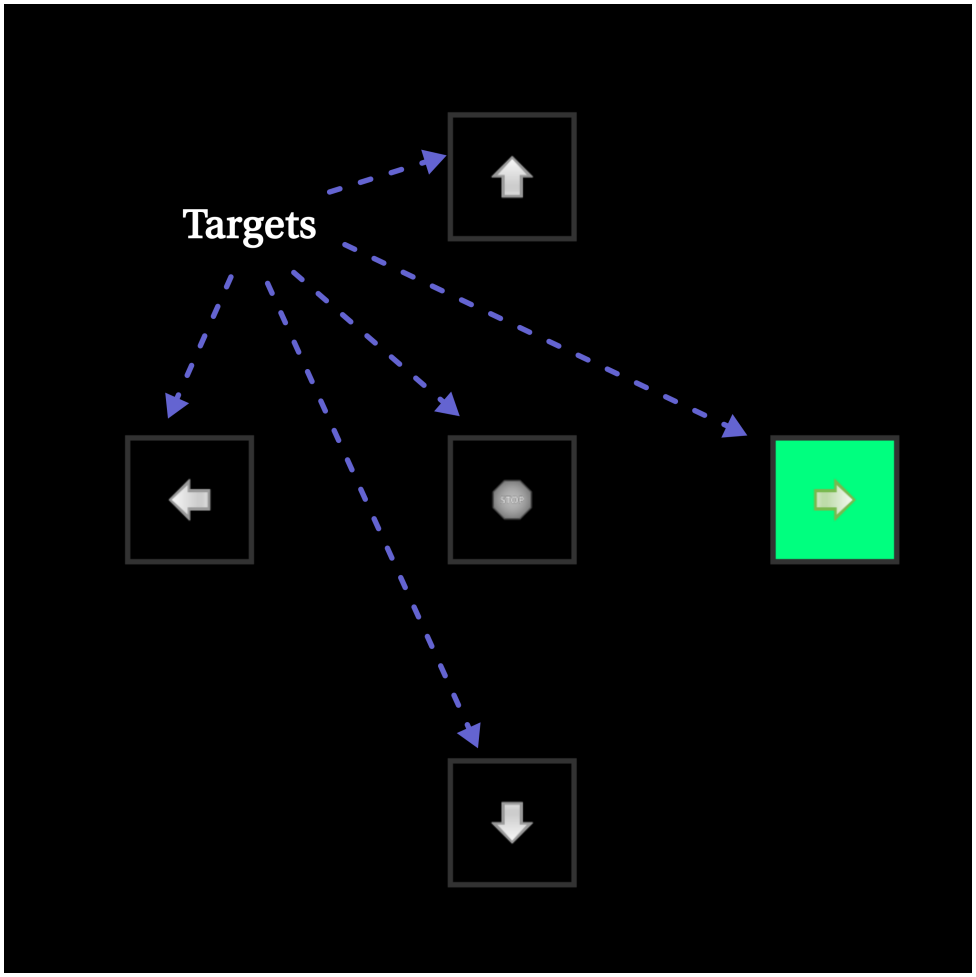


Figure 5.3. GUI used to train the P300-based BCI. The interface shows only five graphical items that represent basic commands of the autonomous system.

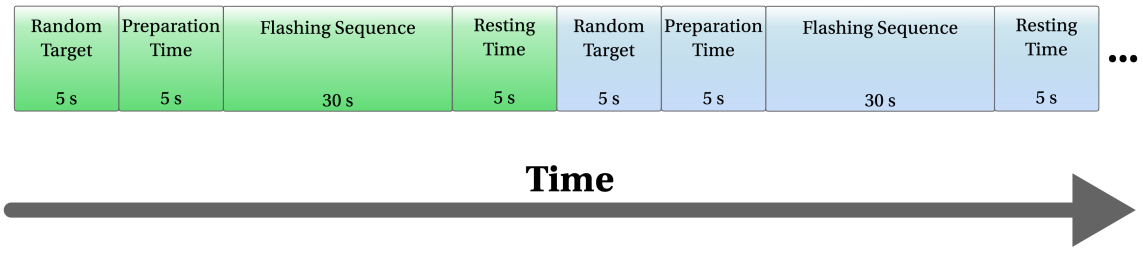


Figure 5.4. Training routine of the P300-based BCI.

After the training routine, the system processes and validates each epoch of EEG data, computes the chosen spatial filters, selects a subset of features with predictive power, and finds the parameters of the machine learning model. This calibration procedure also quantifies the classifier performance. If two training sets are available (i.e. the user performed twice the calibration routine), one of them is used to estimate the model and the other to measure the prediction accuracy. The classification rate

r_i of the *i-th* class ($i \in \{target, non\text{-}target\}$) is calculated as the fraction of epochs of this group in the training set that was classified correctly. Then, the classification error of the same class is

$$err_i = 1 - r_i. \tag{5.38}$$

Finally, the *accuracy* of the classifier is obtained as

$$acc = \left(p_{target} r_{target} + p_{non\text{-}target} r_{non\text{-}target}\right) \times 100\%, \tag{5.39}$$

where p_i is the probability of observing one epoch of the *i-th* class. Typically, p_i corresponds to the proportion of observations in the training set that belong to the category i. Unfortunately, the described training procedure provides around four times more observations of the non-target class than epochs of the target class. In this case, the accuracy is biased towards the most frequent class, giving optimistic estimates if the model classifies correctly in the direction of the non-target class. In order to balance the calculation of the accuracy, $p_{target} = 0.5$ and $p_{non\text{-}target} = 0.5$ in the evaluation of the ERP classifier.

When it is only possible to obtain one training set, the system evaluates the classification performance using *cross-validation*. This method estimates the expected prediction accuracy by calculating the mean accuracy of different partitions of the training set. The more common schemes of this technique are *k-fold cross-validation* and *leave-one-out cross-validation* (Hastie, Tibshirani, & Friedman, 2009). K-fold cross-validation splits the data randomly into K equal-sized parts. For each $k \in \{1, 2, ..., K\}$, the model is fitted excluding the observations of the *k-th* partition, which is used to test the classifier. Then, the accuracy estimates of the K folds are averaged to obtain the prediction accuracy. In contrast, leave-one-out cross-validation (LOOCV) uses the complete training set to estimate the model except for one observation that serves as the testing set. This procedure is repeated for each available observation.

If the training set is small, LOOCV may be preferred because removing a significant portion of the training set could lead to evaluate models that are quite different from the one that is obtained with all the trials. However, this procedure requires finding a classifier for each epoch in the set, demanding a considerable amount of time and computational resources. In most cases, five- or ten-fold cross-validation provide good results without the need of fitting the model too many times (Hastie et al., 2009).

It is important to point out that in order to avoid overfitting in the feature selection step, wrapper-based methods also require a second training set or a cross-validation scheme. The correct estimation of the performance of the individual classification models cannot be done using the training data. Therefore, the BCI uses cross-validation to evaluate the classification errors in the feature selection stage when required.

The system also provides an online evaluation of the P300-based BCI. This test allows quantifying the speed of the interface and the real performance of the ERP detector. Firstly, the interface selects a list of targets that the subject must complete using the BCI. Then, the application starts the flashing sequence and stops when it detects P300 components. The command that elicits ERP signals is highlighted for a short time, and then the graphical elements start to flash again. One by one, the subject selects a list of commands which is compared with the options specified before the test.

According to the offline tests shown in this document, the accuracy of the machine learning model is between 0.80 and 0.95. Hence the selection of one command based on the classification of single trials is risky and may lead to classification errors. In order to improve the performance of the system in online conditions, the interface makes a selection only when one option has elicited evoked responses n_p or more times in the last n_f flashes ($n_f \geqslant n_p$) of the corresponding visual stimulus. The proposed system incorporates an adaptive strategy to determine the optimal values of these parameters that maintain the error rate below the desired level. This strategy is explained in the following section.

One performance metric widely spread to evaluate brain-controlled applications is the *information transfer rate* (ITR). The calculation of this metric is based on the amount of information (expressed in bits per trial) that one selection represents. Let p^* be the probability of selecting correctly one option of the interface, and n_s be the number of available stimulus or options. The bits per trail are calculated as follows (Wolpaw, Ramoser, McFarland, & Pfurtscheller, 1998; Volosyak, 2011):

$$B_t = \log_2 n_s + p^* \log_2 p^* + \left(1 - p^*\right) \log_2 \left(\frac{1 - p^*}{n_s - 1}\right). \tag{5.40}$$

If the user is capable of selecting L_t options in one minute with the BCI, the ITR expressed in bits per minute is

$$ITR = L_t B_t. \tag{5.41}$$

In all the tests presented in this document, each epoch consists of 200 ms of pre-stimulus data, and 800 ms of post-stimulus samples. In the extreme case when $n_f = 1$, $n_s = 1$ and $n_p = 1$, the interface needs an average of 1675 milliseconds to process one target option (800 ms of post-stimulus, 375 milliseconds of average time to wait for the target stimulus, and 500 ms to deal with all the operations that delay the signals, including the FIR filtering). If the minimum resting time allowed in the interface between selections is one second, the maximum theoretical information transfer rate of the BCI would be $ITR_{max} = 52.081$ bits per minute (2.322 bits per trial and 22.423 commands per minute). However, a more realistic calculation of the maximum theoretical information transfer rate corresponds to the case when $p^* = 0.9$, which leads to $ITR_{max} = 37.0751$ bits per minute.

The training routine of the passive detector or error-related potentials uses the same interface shown in Figure 5.4. It also requires the P300-based BCI working in online conditions. In this procedure (illustrated in Figure 5.5), the interface performs the flashing sequence and waits until the BCI detects P300 responses. Then, the interface highlights the command selected by the P300 detector or chooses another option of the interface randomly. About 60 percent of the selections are made according to the results of the P300-based BCI, and the rest corresponds to incorrect choices. It is assumed that the classification rate of the P300 detector is close to 100 percent so that around 40 percent of the trials contain error-related potentials. The parameters n_p and n_f are chosen in such a way that the expected error rate of the P300-based BCI is below 0.01. After repeating this procedure 250 times, the passive detector of machine errors is trained with the recorded data and evaluated using a cross-validation scheme. The training routine is organized in blocks of 50 trials with 2 to 4 minutes of rest periods to reduce eyestrain.

In the online test of the ErrP-based BCI, the P300 detector operates as usual, but sometimes the system selects randomly another option different than the detected by the BCI. In this case, the system evaluates the user's response and determines if the wrong selection triggered an error-related

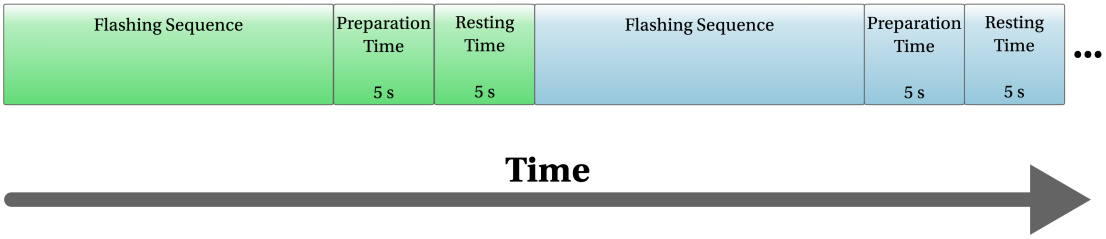


Figure 5.5. Training routine of the ErrP-based BCI.

potential. On the contrary, if the system selects the correct option, the expected result of the ErrP classifier should be non-error related activity. This routine is used to determine the number of times that the classifier detects the error-related potential successfully and the number of classification errors in online conditions.

5.4 Adaptive Control of Classification Errors in the P300-based BCI

As explained in the previous section, the P300-based BCI evaluates more than one epoch of each possible target to decide if one stimulus is eliciting evoked activity. Only when at least n_p trials have been classified in the target group in the last n_f flashes of one option, the interface executes the corresponding command and starts the flashing sequence again. This strategy reduces the classification errors but slows down the application. The choice of the parameters n_p and n_f is crucial to keep a balance between the selection speed and the classification accuracy of the interface.

Let X be number of times that the classifier has detected an ERP correctly. If independence between trials is assumed, the probability of detecting the target command after n_f flashes follows the binomial distribution

$$p\left(X \geqslant n_p\right) = \sum_{i=n_p}^{n_f} \binom{n_f}{i} p^i \left(1-p\right)^{n_f-i}, \tag{5.42}$$

where p is the probability of success of each trial. Likewise, the probability of classifying a non-target item incorrectly also follows the binomial distribution. Because there is more than one non-target command on the screen, if q is the probability of classifying a non-target item incorrectly, the probability of having a classification error is calculated as

$$q^* = p\left(Y \geqslant 1\right) = \sum_{i=1}^{n_s-1} \binom{n_s-1}{i} \tilde{q}^i \left(1-\tilde{q}\right)^{n_s-1-i}, \tag{5.43}$$

where n_s is the number of options, Y is the number of classification errors, and

$$\tilde{q} = \sum_{i=n_p}^{n_f} \binom{n_f}{i} q^i \left(1-q\right)^{n_f-i}. \tag{5.44}$$

Finally, the probability p^{target} of selecting the target stimulus is

$$p^{target} = p\left(X \geqslant n_p\right)\left(1-p\left(Y \geqslant 1\right)\right). \tag{5.45}$$

Table 5.1 illustrates the theoretical error rates and probabilities of success for different system configurations when $n_s = 5$. The probabilities p and q may be estimated in the training phase of the classifier. According to the rates shown in the table, a reasonable selection for n_f and n_p would be 5 and 4 respectively. With this configuration, the probability of selecting wrong commands is below 0.03. Moreover, another option may be more suitable to tighten the control of classification errors (for example $m = 10$ and $n = 7$).

BCI Configuration				$\mathbf{p}^{target}$	$\mathbf{q}^{*}$
$\mathbf{n_f}$	$\mathbf{n_p}$	**p**	**q**		
1	1	0.9	0.1	0.5905	0.3439
2	2	0.9	0.1	0.7781	0.0394
2	2	0.8	0.2	0.5436	0.1510
3	2	0.9	0.1	0.8676	0.1074
3	2	0.8	0.2	0.5775	0.3555
3	3	0.9	0.1	0.7261	0.0040
3	3	0.8	0.2	0.4958	0.0316
4	3	0.9	0.1	0.9338	0.0147
4	3	0.8	0.2	0.7336	0.1044
4	4	0.9	0.1	0.6558	0.0001
4	4	0.8	0.2	0.4070	0.0064
5	4	0.9	0.1	0.9169	0.0018
5	4	0.8	0.2	0.7177	0.0266
5	5	0.9	0.1	0.5905	0.0000
5	5	0.8	0.2	0.3273	0.0013
10	7	0.8	0.2	0.8761	0.0035

Table 5.1. Theoretical probabilities of success and misclassification for different configurations of the P300-based BCI. In all cases, $n_s = 5$.

Every time that the interface evaluates a new epoch, the values of p^{target} and q^* can be estimated using Expressions 5.45 and 5.43 respectively. The BCI only needs the classification rates obtained in the training phase of the interface, the number of previous epochs classified in the target class, and the total number of epochs already analyzed. This information can be used to choose the parameters n_f and n_p dynamically to achieve a desirable performance level.

Let q^{max} be the maximum allowed error rate. Let n_f^{min} and n_f^{max} be the limit values of the number of flashes to classify. In online conditions, the BCI analyzes the epochs of one target option according to the following algorithm:

1. Set $n_f = n_f^{min}$.
2. Wait until n_f epochs have been classified.
3. Calculate the number n_p of trials classified in the target class.
4. Obtain the probability q^* for the current n_f and n_p values.

5. If $q^* \leqslant q^{max}$, the corresponding option is selected and the flashing sequence starts again. Otherwise, set $n_f = \min(n_f + 1, n_f^{max})$ and repeat step 2.

This procedure is used in all the online tests of the P300-based BCI and the final tests on the wheelchair. In both cases, $q^{max} = 0.05$, $n_f^{min} = 2$ and, $n_f^{max} = 10$.

5.5 Offline and Online Tests of the P300 Detector

The proposed hybrid BCI consists of a set of real-time processing tools easily configurable to different users and scenarios. All the processing stages, including the signal enhancement, feature extraction, classification, and control interface, have several options that can be evaluated to reach an optimal performance for specific subjects and applications. The purpose of this design is to assess different approaches that have been reported in previously published works and determine what tools and algorithms work correctly in the implemented system. The non-stationary nature of the brain activity, combined with the high variability observed in the EEG among subjects, entails a revision of several techniques to identify a robust and reliable system configuration.

This section summarizes the tests that helped to characterize and calibrate the P300-based BCI. For this purpose, ten healthy subjects (M = 7, F = 3) participated in one study in which the users trained the interface with the calibration procedure described in this chapter. None of the participants had previous experience with BCI technology. The individual training sets obtained in the experiment were used to adjust and evaluate offline the machine learning model with different feature extraction methods and classification strategies. Finally, after training the interface, the subjects interacted with the BCI in an online session to measure the speed and the accuracy of the interface.

The first component of the BCI evaluated during the training sessions was the speed of the flashing sequence. The subjects tried to follow stimuli at different speeds, and most of them reported that they could detect all the flashes without problems when the icons flicker for 150 ms (75 ms highlighted, 75 ms with gray background). At faster rates, the participants missed some flashes, whereas at lower speeds, the users got bored and lost attention. However, two subjects (number 7 and number 9) indicated that this rate is too slow and it is easy to get distracted between flashes. For them, 100 ms of stimulus presentation was better (50 ms highlighted, 50 ms with gray background).

All the participants reported that the stimulus size is appropriate for the interface. They did not find problems with the colors and the background. However, two participants elicited low ERP responses with the flashing stimulus (number 4 and number 8). In these cases, the standard flashes were replaced by happy faces to highlight the interface options (see Figure 5.6). This alternative was suggested by L. Chen, Jin, Zhang, Wang, and Cichocki (2015), who proved that this type of stimulus is highly efficient to elicit ERP responses. The rest of the subjects did not have problems with the standard flash.

5.5.1 Pointwise Comparison of Target and Non-Target Responses

A simple way to determine if the training routine is useful to obtain valid observations for the target and non-target classes is to compare the preprocessed potentials of both types of responses at each channel position and time index in the post-stimulus segment. The BCI performs this operation using the Wilcoxon rank sum test (Noether, 1992), which is a univariate nonparametric test that determines

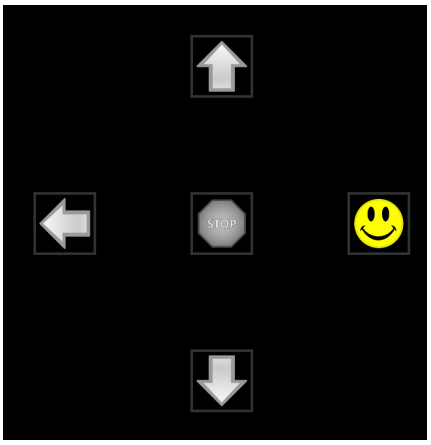


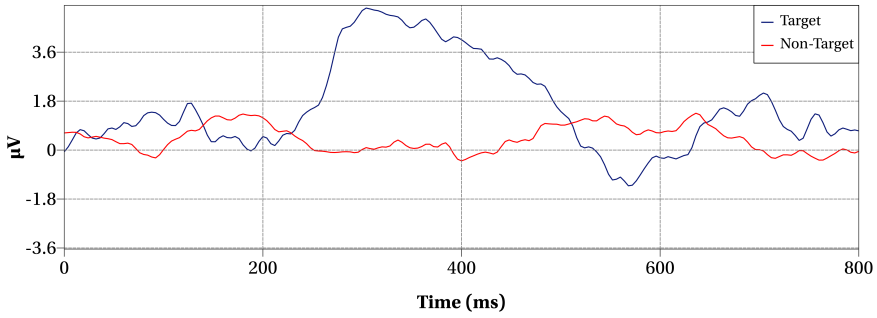
Figure 5.6. Dummy face stimulation method. The icons are replaced by happy faces when they are highlighted.

if two independent samples were selected from the same distribution. The alternative hypothesis indicates that both samples were drawn from different distributions. If the proportion of tests with a p-value below a pre-defined significance level (0.05 for instance) is considerably large, it is possible to say that the training set contains observations that are representative of both classes and the subject is able to produce the desired activity with the training routine.

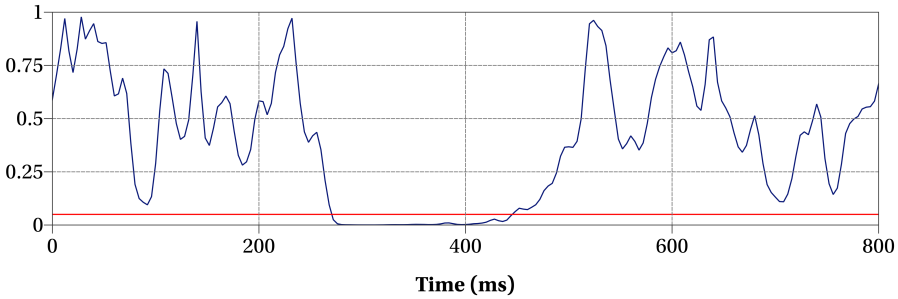
In this pointwise comparison, the training sets of the participants were processed as follows. First, a Butterworth band-pass filter (IIR) of 0.1 to 40 Hz was applied to the signals. Then, the data was separated in epochs of 200 ms of pre-stimulus activity and 800 ms of post-stimulus activity. All the epochs were validated, removing all the time windows with noise and artifacts. Finally, the epochs with clean signals were baseline corrected and used to compute the grand averages of each class and the p-values of the comparison.

Figure 5.7 illustrates the ERP responses and the corresponding p-values obtained for one subject at one electrode position (Cz). In this example, the p-values of the individual tests are below 0.05 mostly around the 300 ms, which is the expected result for P300 potentials. Clearly, the target stimuli elicited evoked activity, while the non-target class had a different behavior in the timeline, oscillating around zero in the post-stimulus segment.

The results obtained for the 16 channels may be illustrated in the timeline using *topographic* plots. Figure 5.8 presents the charts of the grand averages and the calculated p-values for the same subject using heads that represent scalar variables interpolated among the scalp. With the aim of reducing the size of the plots, each head depicts the average of several consecutive samples in the time domain (ten in this plots). These type of charts help to determine how the evoked potentials of the target class evolve in the time course. In this example, before 300 ms after the stimulus presentation, the stronger evoked responses are observed close to the occipital area. Likewise, around 400 ms after the stimulus onset, the frontocentral areas also present evoked activity. The lowest p-values of the performed tests were obtained mostly between 200 ms and 400 ms in the timeline. Additionally, Figure 5.9 presents the significant differences between the target and non-target observation for a significance level of 0.05. Both the uncorrected and corrected results for multiple comparisons are represented in the timeline. The two-stage *false discovery rate* (FDR) procedure for multiple testing described in (Benjamini, Krieger, & Yekutieli, 2006) was used to find the corrected maps ($FDR = 0.05$).

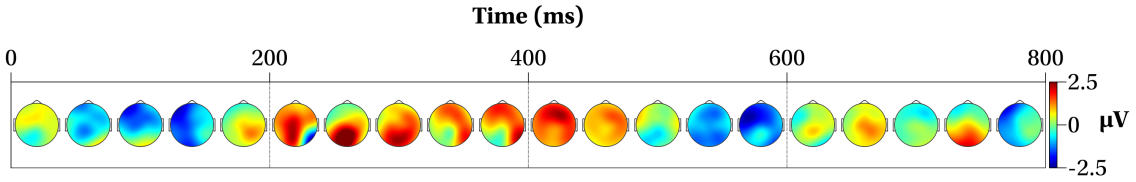


(a) Grand averages.

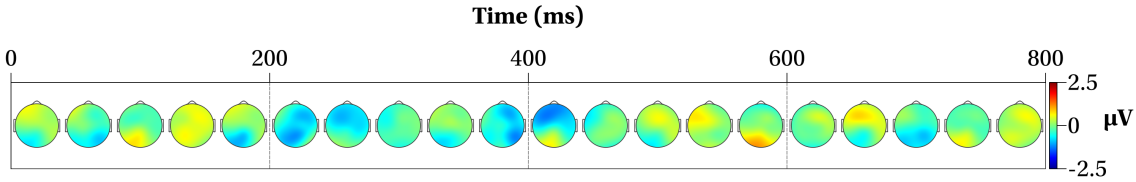


(b) Two-tailed p-values.

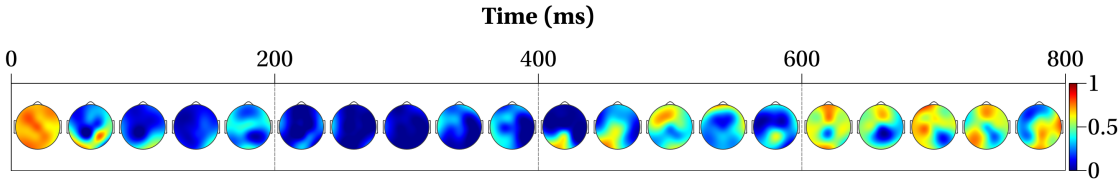
Figure 5.7. Comparison between the target and non-target responses for one subject at channel Cz. The grand averages of the target and non-target responses are shown in (a). The p-values obtained with the pointwise testing procedure at the same electrode position are illustrated in (b). In this chart, the blue line represents the p-values, and the red line indicates a significance level of 0.05.



(a) Grand averages of the target class.

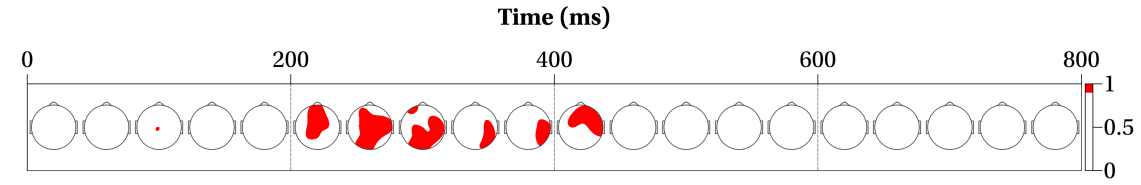


(b) Grand averages of the non-target class.

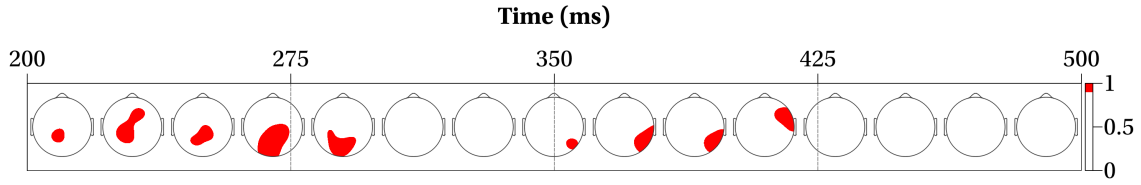


(c) Two-tailed p-values.

Figure 5.8. Grand averages of one subject for the target (a) and non-target (b) classes. The p-values of the pointwise hypothesis tests are shown in (c).



(a) Uncorrected significant differences.



(b) Corrected significant differences.

Figure 5.9. Significant differences between the target and non-target classes for one participant. The uncorrected map of differences for $\alpha = 0.05$ is shown in (a). The corrected map for $FDR = 0.05$ is illustrated in (b). The heads in (b) represent the average activity of five consecutive time points.

In most practical cases, the topographic representations are enough to determine how strong the P300 responses are. If the plots show clear ERPs between 200 ms and 400 ms for the target class, and at the same time the corresponding p-values around this time segment are relatively small, there is a high probability that the system will find a suitable classification model with a high accuracy to discriminate between the target and non-target classes. Likewise, the proportion of significant differences (corrected or uncorrected) also provides information about the characteristics of the training set. For example, in the previous case, around 25% of the calculated p-values are below a significance level of 0.05. This result indicates the existence of a high number of variables with discriminative information, which can be used to build a classification model.

Intended to formalize the pointwise testing procedure described in this subsection, the significance level of the proportion of uncorrected significant differences was calculated with a permutation test (Good, 2006). In this nonparametric methodology, the null hypothesis indicates that observations of both classes (target and non-target) are exchangeable so that under the null hypothesis, any random permutation of the class labels produces the same result as the value calculated with the original data. The alternative hypothesis is accepted when the tested variable is an extreme value in the empirical distribution built with n random permutations of the labels. In order to test the calculated proportion p of positive tests, all the observations of the two classes are randomly resampled without replacement and split into two new groups of the target and non-target classes. For each permutation $i = 1, 2, ..., n$, the proportion p_i is calculated using the permuted data. The p-value of the rate obtained with the original data corresponds to the fraction of sampled permutations with a greater value than the proportion p. That is:

$$p\text{-}value = \frac{1}{n}\sum_{i=1}^{n} I\left(p_i > p\right). \tag{5.46}$$

Table 5.2 summarizes the p-values for $n = 1000$ obtained with the training sets of the subjects that participated in the study. In all cases, the ERP responses correspond to P300 evoked potentials for the target class.

Subject	Proportion of Significant Differences (%)	P-Value
1	26.63	0.002
2	22.38	0.044
3	26.00	0.013
4	39.36	0.005
5	29.43	0.008
6	25.50	0.025
7	39.09	< 0.001
8	41.44	0.004
9	21.86	0.022
10	37.43	0.006

Table 5.2. Proportion of significant differences between target and non-target responses of each participant.

5.5.2 Comparison of Preprocessing and Feature Selection Methods

Figure 5.10 shows the classification rates (cross-validated, five folds) obtained with different feature extraction methods. The accuracies for each class (target and non-target) and the average performance are plotted individually in the chart. The boxplots represent the quartiles of the results obtained for the ten participants. The bottom and top of the boxes correspond to the first and third quartiles, and the midline is the median of the data. The end of the whiskers represent the minimum and maximum of all the calculated accuracies.

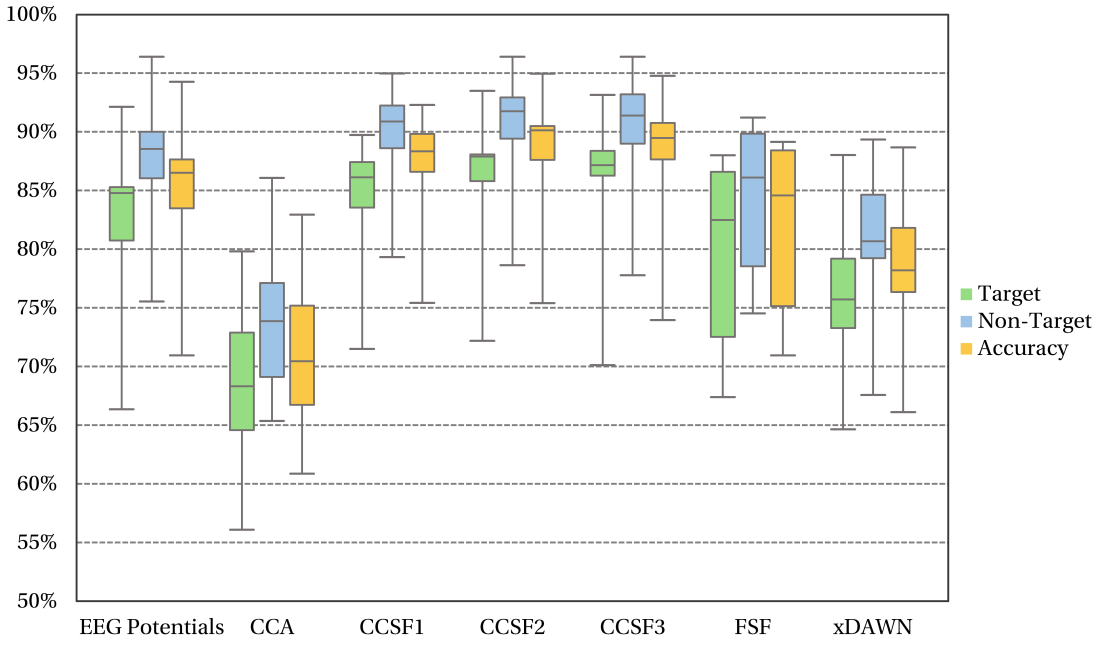


Figure 5.10. Classification rates obtained with different feature extraction methods for P300 responses.

The methods shown in the chart are: filtered EEG potentials, canonical correlation analysis of EEG

time-series (CCA), time-dependent canonical correlation spatial filters (CCSF1), canonical correlation spatial filters for target responses (CCSF2), canonical correlation spatial filters for target and non-target responses (CCSF3), Fisher spatial filters (FSF), and the xDAWN algorithm. The system only calculated features for the best three spatial filters obtained with the methods CCSF2, CCSF3 and xDAWN. In this comparison, the raw EEG potentials were bandpass filtered in the preprocessing stage using a FIR filter with a bandpass from 4 to 14 Hz. Re-referencing and artifact suppression were not considered in these results. The data were separated in epochs of 200 ms of pre-stimulus activity and 800 ms of post-stimulus activity. All the processed trials were validated, baseline corrected and downsampled at a rate of 4. In the classification stage, LDA combined with the univariate filter approach for feature selection were used to obtain the machine learning models. Features were ranked with the Fisher's separability criterion, and the classifiers did not include more than 200 features.

The BCI reached the best classification rates with CCSF2 (87.86%), followed by CCSF3 (87.52%) and CCSF1 (86.70%). These algorithms compute spatial filters based on the canonical correlation analysis. Then, the EEG potentials obtained an average accuracy of 84.86%, which means that the rest of the feature extraction methods could not improve the performance of the BCI. The accuracy ranges represented in the boxplots also indicate that CCSF1, CCSF2 and CCSF3 have lower variabilities than the other methods. These results show that the three spatial filters based on the canonical correlation analysis are a viable option for classifying ERP data.

The BCI also performs a permutation test to obtain the significance levels of the classification accuracies. In this case, for each permutation of the class labels, a new machine learning model is estimated using the permuted observations. Then, the new classifier is evaluated using the cross-validation procedure. After obtaining the classification rates for n random permutations, the p-value of the classification accuracy corresponds to the proportion of sampled permutations with a greater value than the rate calculated with the original data. The calculated p-values of the classification rates shown in Figure 5.10 are below 0.001 for $n = 1000$, which prove that the results were not obtained by chance and the classification models have real predictive power.

One alternative to avoid the problem of evaluating one by one the feature extraction methods is to compute all the variables and allow the feature selection stage to discover a subset that provides an optimal performance. Using the same preprocessing parameters described above, the average performance of the system obtained with LDA and the univariate filter is close to 89%, which is higher than the average performance obtained with CCSF2. However, this operation increases the computational time to find the classification rule, and the growth in performance is indiscernible. The feature extraction methods tested in this study are not complementary, so using more than one of them should not represent a better classification rate.

The re-referencing methods described in Chapter 4 may modify the overall performance of the BCI because they change the nature of the signals that are processed and classified by the system. Figures 5.11 and 5.12 show the classification rates of each subject obtained after applying the CAR method and the surface Laplacian respectively to the training sets.

The best accuracies after applying the CAR method were obtained with CCSF2 (87.42%), CCSF3 (87.17%) and CCSF1 (87.15%). In general, the BCI achieved similar accuracies than the observed without CAR with all the feature extraction algorithms. Only the classification rates obtained with CCA (77.84%) improved more than five percentage points. A similar pattern was observed after applying the

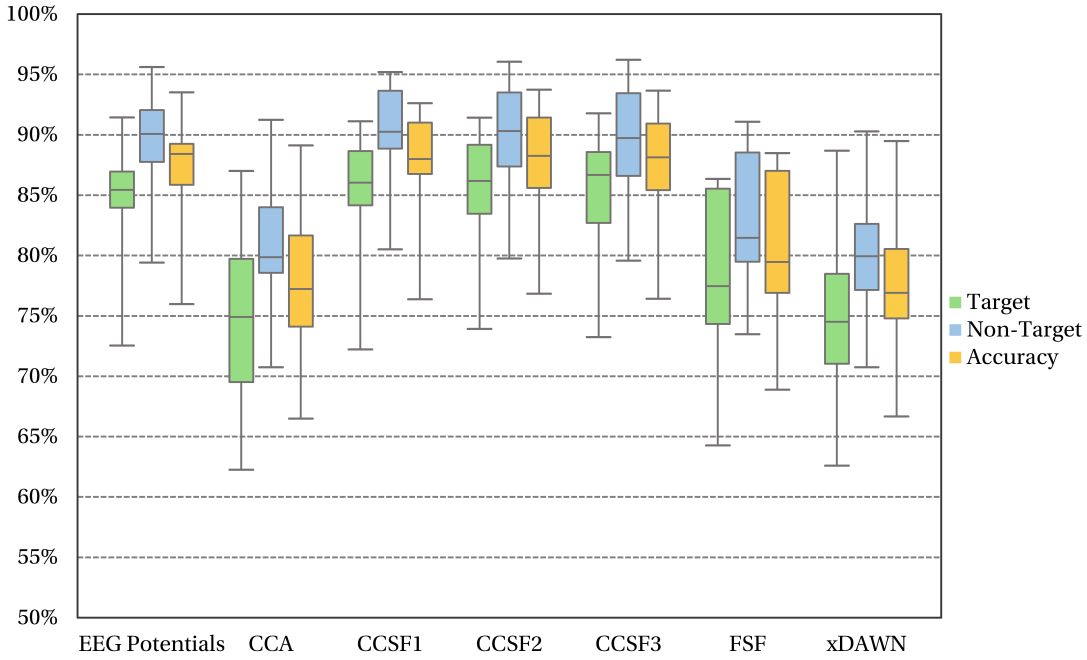


Figure 5.11. Classification rates obtained when the common average reference was applied in the preprocessing stage.

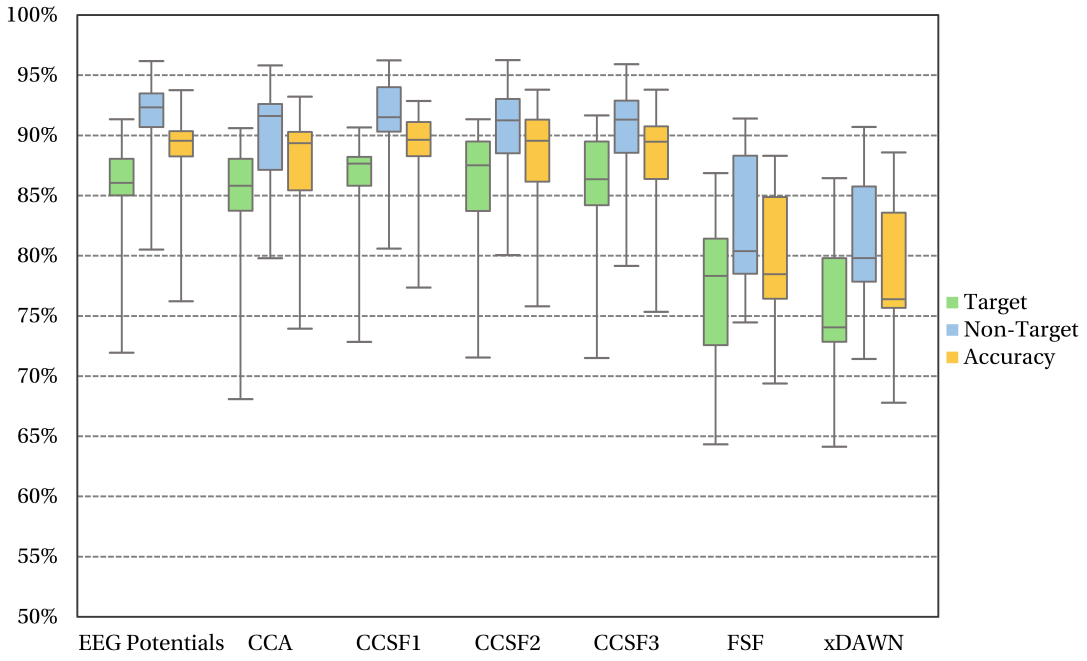


Figure 5.12. Classification rates obtained when the surface Laplacian was applied in the preprocessing stage.

surface Laplacian. The best results in this case were obtained with CCSF1 (87.87%), CCSF2 (87.62%) and CCSF3 (87.59%). However, it is important to note that the accuracies with CCA (86.85%) after applying the surface Laplacian represents an improvement over eleven percentage points.

Another interesting test consists on investigating the effect of applying the ASR method to suppress artifacts before the computation of the classification models. In this case, the ASR method helps to

reject fewer epochs from the training set, but the new clean signals may have weaker ERP responses as a consequence of the combination of resting state data and the original training set. Figure 5.13 presents the performance of the P300-based BCI after removing artifacts. As was expected, CCSF2 (87.94%), CCSF3 (87.51%) and CCSF1 (87.10%) obtained the best classification rates, which are similar to the presented previously.

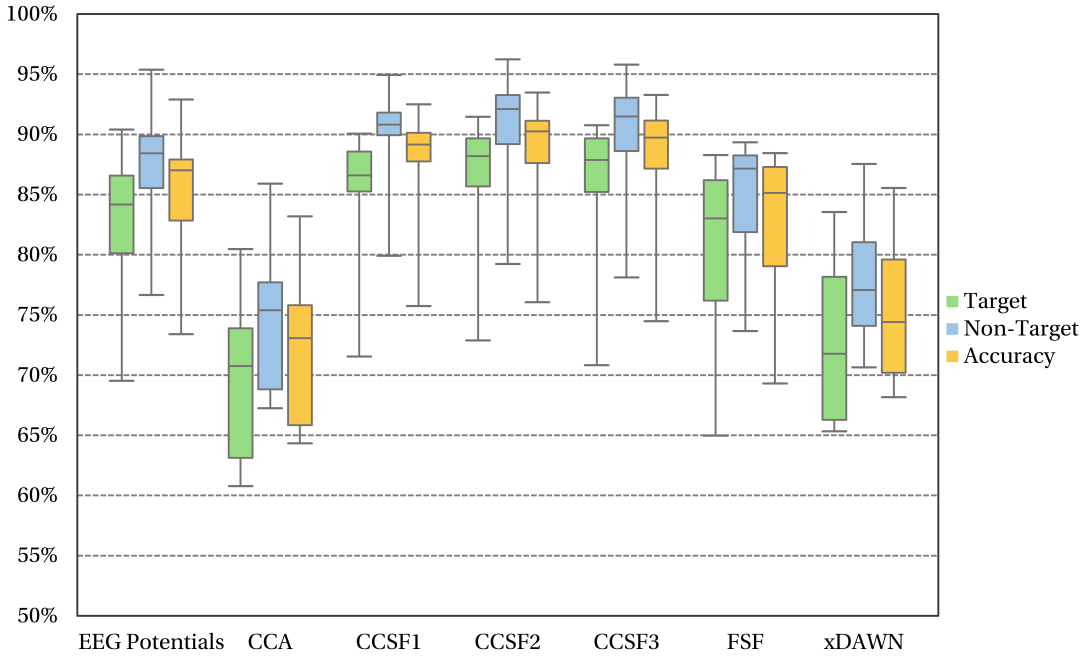


Figure 5.13. Classification rates obtained when artifacts were rejected with the ASR method.

These results indicate that the ASR method does not decrease the overall performance of the BCI, even though it uses data from another condition (resting state) to estimate the removed components. In consequence, the artifact subspace reconstruction algorithm can improve the online performance of the BCI because it reduces the time expended in waiting for clean signals. However, the ASR method is only useful when one particular subject produces a high number of artifacts. In the offline tests, more than 95% of the epochs were free of artifacts, so that the ASR method did not increase the size of the training sets significantly. However, if features are also computed from frontal channels, around 10% of the epochs are excluded because these electrode positions are more prone to collect noise and artifacts.

Figure 5.14 illustrates the performance of the classifier with different numbers of features included to train the machine learning model. As stated above, the results in this subsection were obtained using a univariate filter which ranks the characteristics according to the individual separability between classes. The chart suggests that only classifiers based on EEG potentials require a significant number of features to achieve the best possible performance. On the contrary, the methods based on spatial filtering produce high classification rates with all the number of features tested in this study. Finally, the BCI obtains a low performance in all cases with the CCA method, which generates only eight characteristics. These results indicate that spatial filtering methods are more efficient to process the available information than the other algorithms and they can reach a good performance with a reduced number of features.

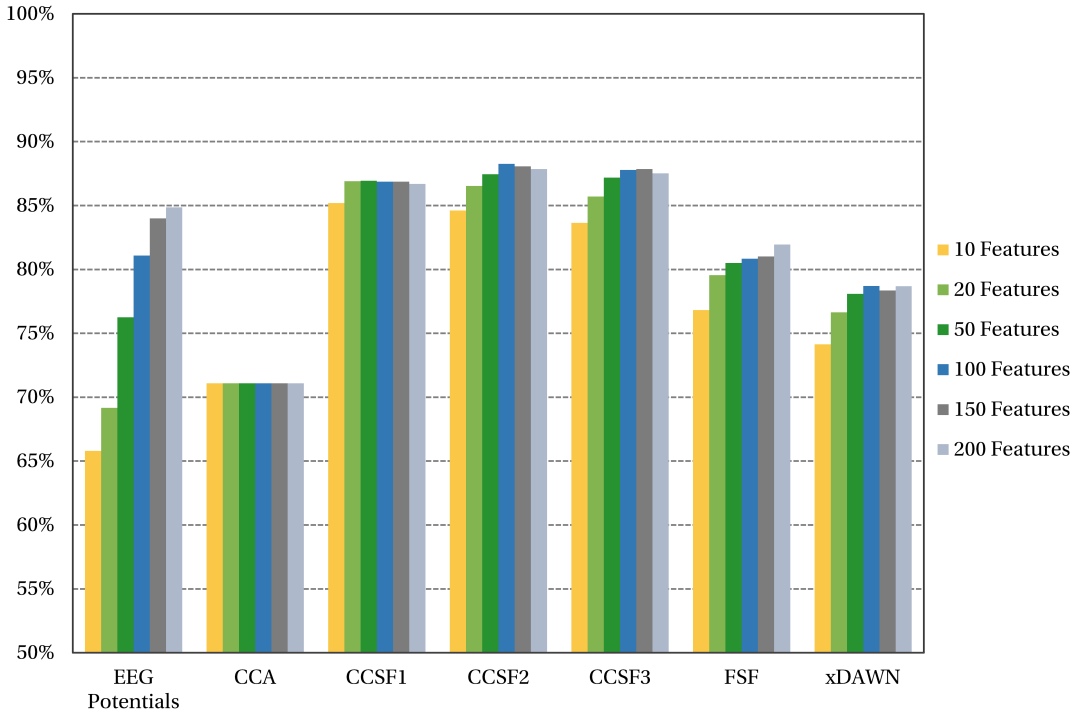


Figure 5.14. Average classification rates of the P300-based BCI obtained with different number of features.

It is important to point out that the previous results were calculated using only the selected EEG sensors of the P300-based BCI. In contrast, Figure 5.15 shows the classification rates achieved with features of all the available electrodes. According to this chart, using all the sensors hurt the performance obtained with the EEG potentials (81.74%) and the xDAWN algorithm (70.70%). For the rest of the methods, there is not an important difference in performance. The observed classification rates can be explained by the fact that some of the excluded channels are noisier than the rest of the signals, especially around the frontal positions. At the same time, the evoked responses are weaker in these electrodes. The excluded sensors do not provide more information than the eight positions chosen in advance, so there is no need to compute features using all the signals because the classification rates will not improve.

It is also interesting to visualize the spatial weights calculated with data of all the EEG sensors. In the case of the CCSF1 method, the BCI finds one filter for each time point after the stimulus onset. The final result is a sequence of spatial weights that can be represented in the timeline. Figure 5.16 shows the filters calculated for subject 1 at seven time lags. In this example, the filter at 320 ms presents the pattern with the highest coefficients in absolute value in the timeline. Around this point, it is expected to observe the P300 response.

Figure 5.17 illustrates the best three spatial filters calculated for subject 1 with the CCSF2, CCSF3 and xDAWN methods. As may be observed, the filters based on the canonical correlation are similar in this example. This figure may indicate that including epochs of the non-target class (which is the case of CCSF3) to built the spatial filter is not relevant, and the weights can be calculated using only the target observations just as it is done in the CCSF2 method. In contrast, the xDAWN algorithm provides a different distribution of weights, which is not unusual because the optimization problem solved in

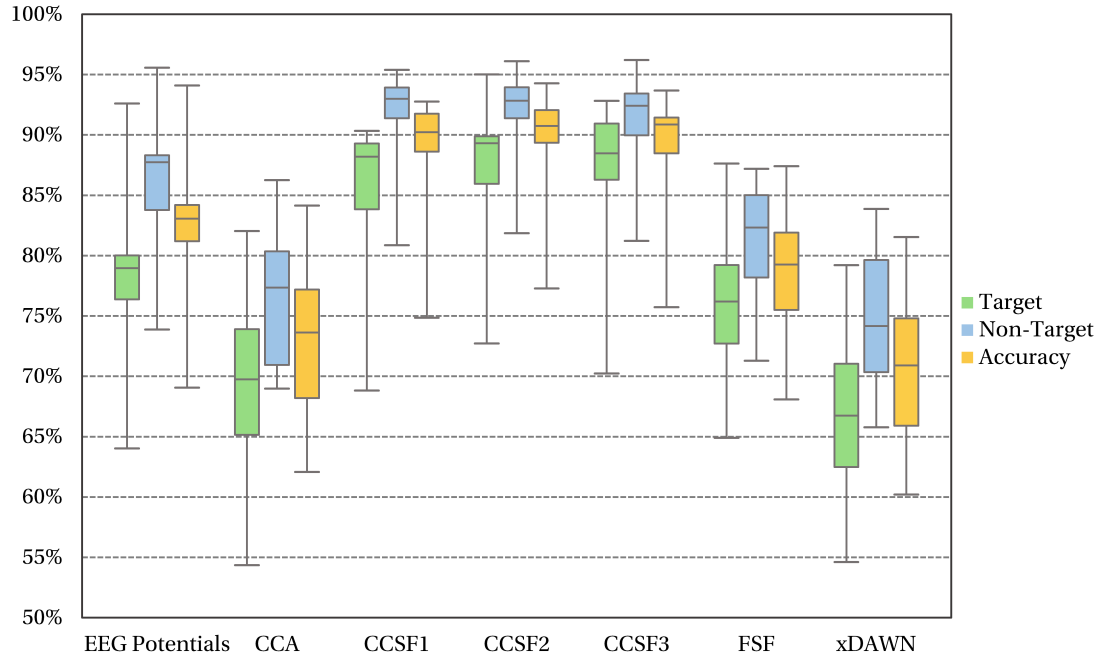


Figure 5.15. Comparison of feature extraction methods of the P300-based BCI using all the available EEG sensors.

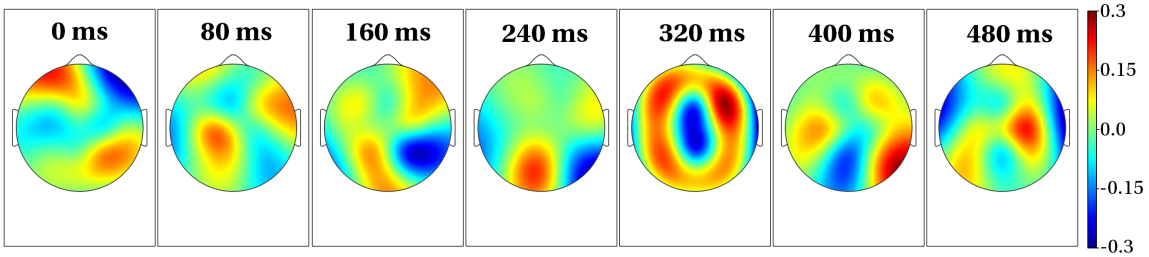


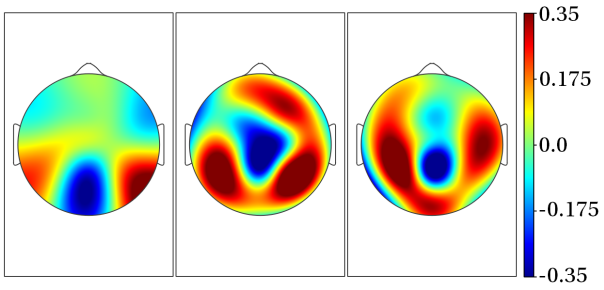
Figure 5.16. Spatial filters at different time lags using the CSSF1 method.

this case to find the optimal projections is different from the one used in the rest of the algorithms.

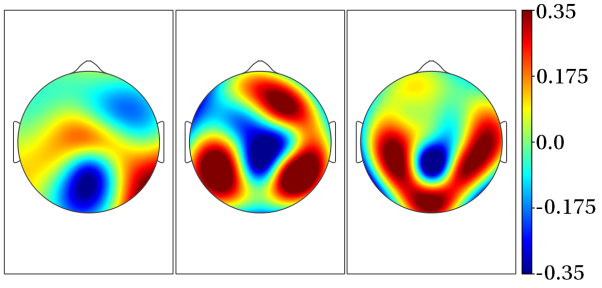
Finally, Figure 5.18 summarizes the results presented in the course of this subsection. One can conclude that the methods based on spatial filters combined with the canonical correlation analysis are a robust and reliable option for classifying ERP data. In most cases, CCSF1, CCSF2 and CCSF3 provided the best classification rates close to 90% independently of the selected preprocessing strategy. For this reason, CCSF2, without re-referencing and artifact suppression will be used in the rest of the tests presented in this section.

5.5.3 Comparison of Classification Methods and Feature Selection Algorithms

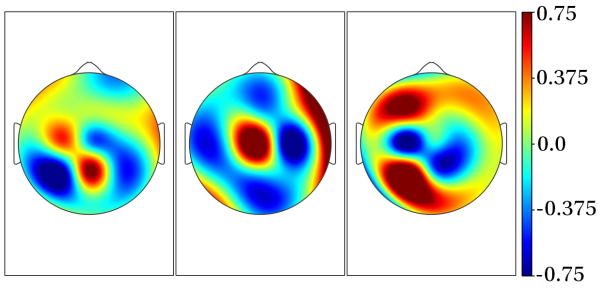
The decision of selecting a particular classification model in any machine learning application highly depends on the nature of the extracted characteristics that represent the classes and how the data can be modeled. Perhaps, one of the most common questions is whether a linear function can separate the classes or it is necessary to model the features with non-linear relations. In this work, five different models were tested to answer this question for the features used in the online tests: three linear classifiers (LDA, logistic and SVM), one quadratic model (QDA) and one non-linear function (radial



(a) CCSF2



(b) CCSF3



(c) xDAWN

Figure 5.17. Spatial filters obtained with different methods for ERP-based BCIs.

basis function SVM or RBF-SVM).

Figure 5.19 shows the classification accuracies obtained with the five described models. The CCSF2 method and the univariate filter combined with the Fisher's separability criterion were used for feature extraction and selection in these tests. The number of selected characteristics was 50. From the chart, it is clear that there are not advantages in using the non-linear classifiers. The five models obtained similar results between 86% and 88% of average accuracy, so that the assumption of linear separability works well in this particular case. LDA with regularized covariance estimates (87.70%) and the logistic classifier (87.56%) gave the best accuracies in discriminating the target and non-target classes. Because of LDA is relatively easy to train and evaluate, the following tests were conducted with this classifier.

Another important question regarding classification models is how many features the system must include to achieve an optimal performance and how to select them. Previously, this chapter described some approaches that were incorporated into the BCI. The methods implemented are: univariate filters combined with different performance metrics (the Fisher's separability, the Pearson's correlation and the inverse of the Davies-Bouldin index), a filter-wrapper approach (using the Fisher's separability to rank features), stepwise regression, and a wrapper-based forward selection. The classification

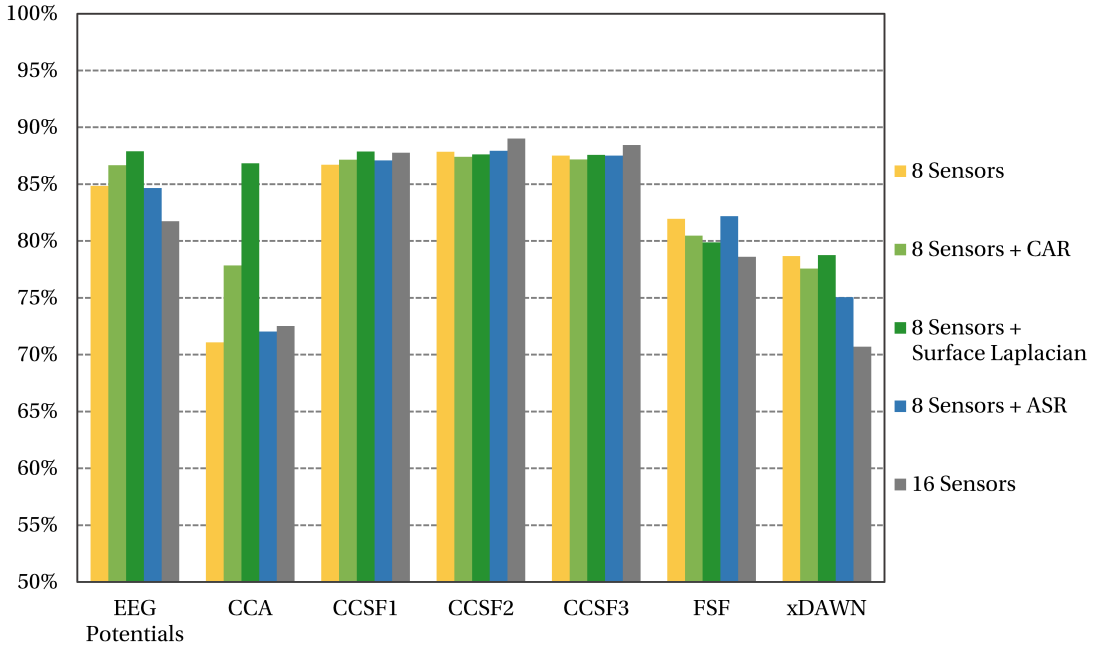


Figure 5.18. Comparison of electrode arrangements and preprocessing algorithms in the P300-based BCI.

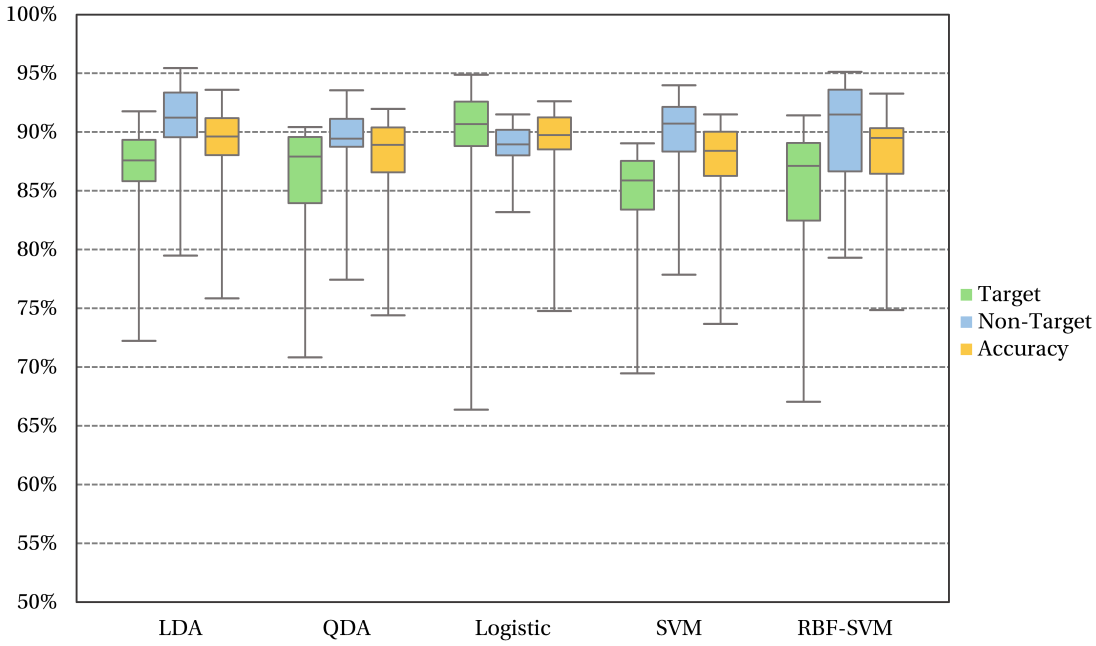


Figure 5.19. Comparison of the classification methods implemented in the P300-based BCI. The univariate filter approach for feature selection combined with the Fisher's separability criterion was used to rank and select the best 50 features.

accuracies obtained with the feature selection methods are shown in Figure 5.20. Additionally, Figure 5.21 illustrates the number of features selected by each algorithm. Practically, the BCI has similar accuracies with all the tested methods, but the filter-wrapper approach is the one that selects the smallest feature sets (19.3 features). In contrast, step-wise regression obtained better accuracies (88.16%), and only includes on average another five features in the model. According to this test, the

combination of the CCSF2 method with LDA and step-wise regression is a good choice.

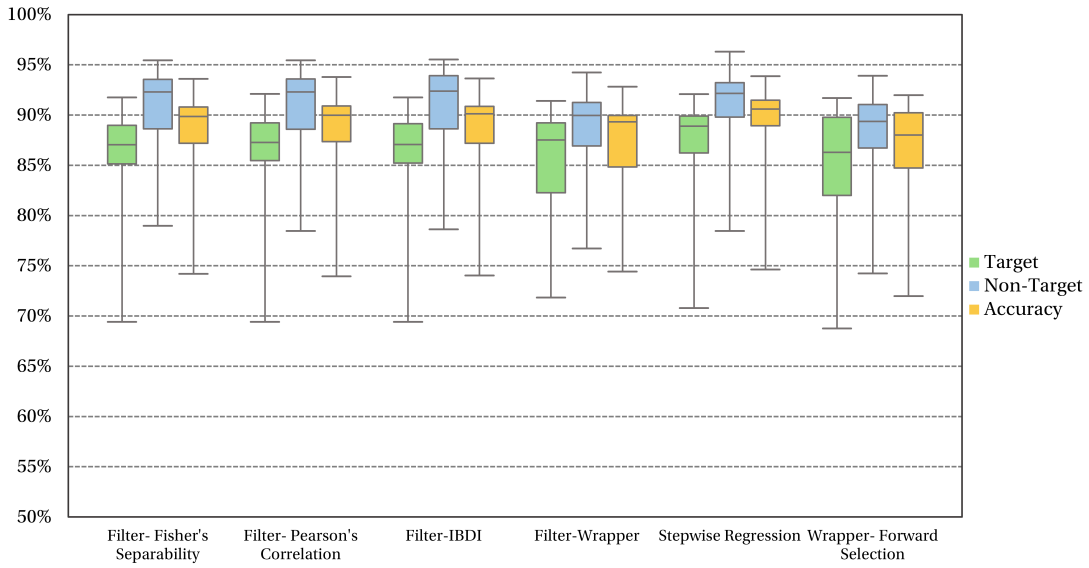


Figure 5.20. Classification rates obtained with different feature selection methods implemented in the P300-based BCI. The LDA classifier was used in this comparison. No more than 50 features were included in the classification models.

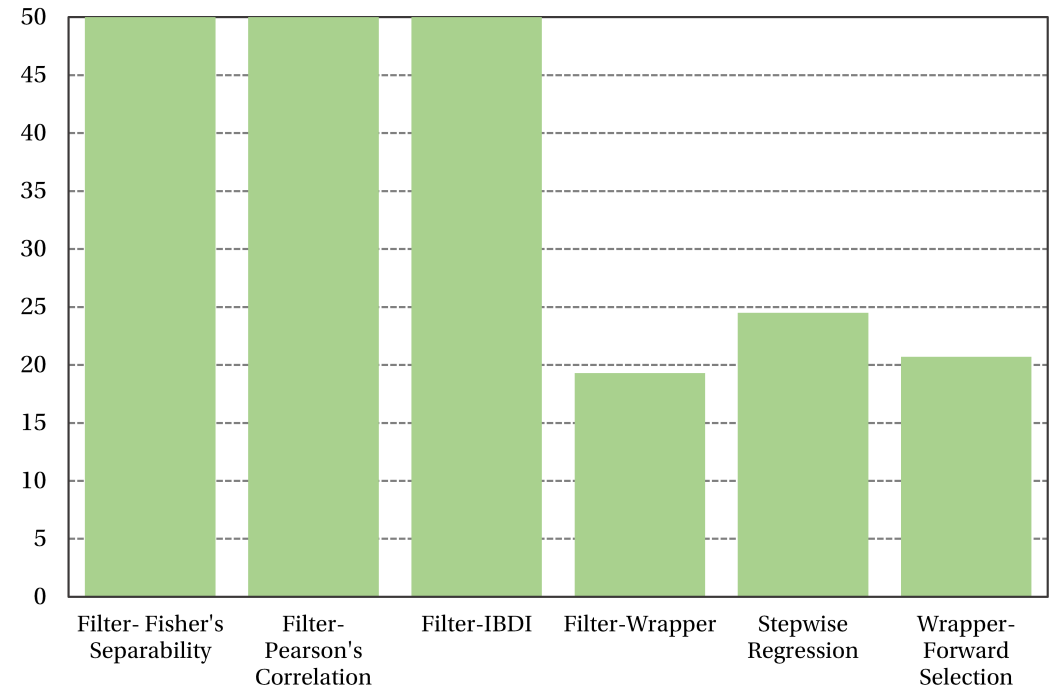


Figure 5.21. Comparison of the number of features obtained by different feature selection methods in the P300-based BCI.

The training set size is another important variable to take into consideration in the design of the calibration routines. On the one hand, small data sets may lead to a poor system performance and a high probability of misclassification. On the other hand, prolonged training routines are exhaustive for the users and discourage the long-term use of the BCI. With the aim of investigating the relation

between the training set size and the accuracy of the system, different portions of the data sets were used to fit and evaluate the classification rule. In this test, the machine learning model was estimated using a percentage of the total number of epochs of each class. The trials of the new training set were selected without replacement randomly. Then, the new classifier was tested with the rest of the data. This procedure was repeated 100 times to calculate the average accuracy.

Figure 5.22 shows the classification rates achieved with different percentages of epochs of the original training data. As can be seen from the chart, the performance stabilizes above 0.85 around the 30% of the data, which corresponds to 84 epochs of the target class and 336 observations of the non-target class. However, it is interesting to note that even with only 20% of the epochs, it is possible to fit a classifier with an accuracy above 0.8. These results suggest that it is feasible to reduce around 70% the training time of the BCI without sacrificing the accuracy of the classifier.

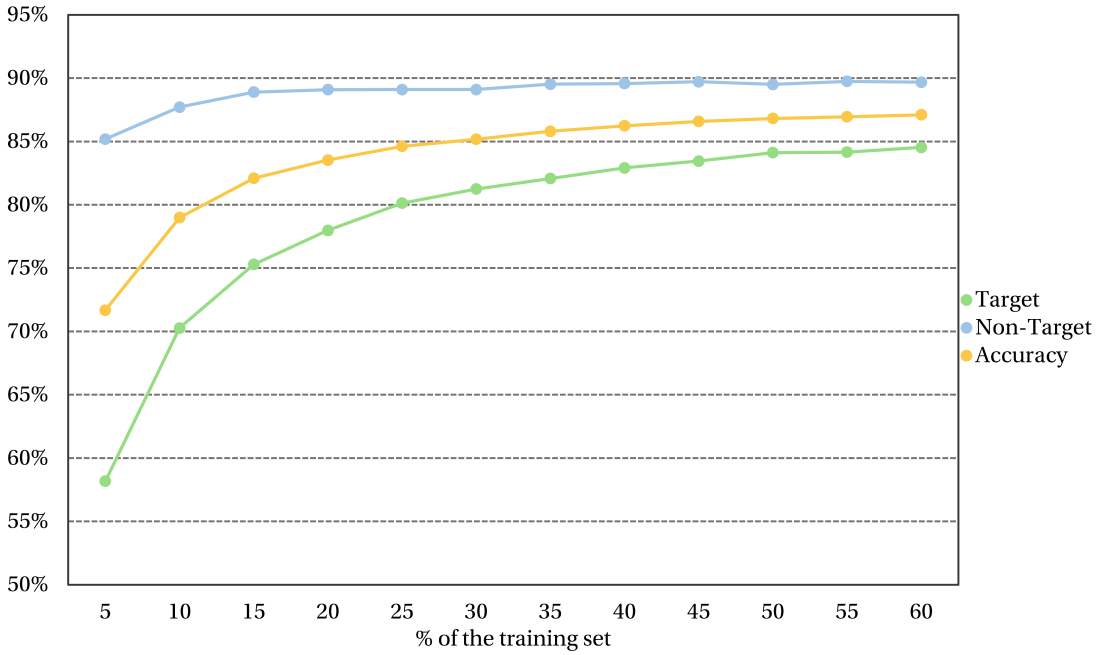


Figure 5.22. Classification accuracies obtained with different training set sizes in the P300-based BCI. The horizontal axis represents the percentage of epochs selected randomly from the original sets. The training data of each participant consists of 280 epochs of the target condition and 1120 trials of the non-target class.

Another offline test of the P300-based BCI consists of determining the performance of the system when a data set is evaluated with a classifier trained with epochs collected in a different experimental session. To this end, three subjects participated four times in the experiment with a difference of two months between sessions. Both the cross-validated accuracy of the model of each data set and the respective classification rates of the model trained with epochs from a different session were calculated and compared. Table 5.3 summarizes the results of this experiment. As can be seen from the table, the accuracies worsen consistently when the classifier is fitted with data recorded in another session. Sometimes, the differences are negligible, but in other cases, the performances drop below 0.7. These results give clear evidence that the classification model requires epochs of the target and non-target classes measured close to the moment of the online operation, so that with the aim of reducing the risk of a significant decrease in the system performance, the classifier is updated constantly with new

recorded data. This operation is implemented in an adaptation step included in the BCI, which is described in the next chapter.

Subject	Class/ Accuracy	Session 1		2		3		4	
1	Target	83.98	**76.92**	85.59	**78.77**	87.21	**85.12**	89.31	**82.41**
	Non-Target	86.93	**84.33**	87.49	**88.36**	87.16	**88.02**	88.85	**89.88**
	Accuracy	85.45	**80.62**	86.54	**83.56**	87.18	**86.57**	89.08	**86.15**
2	Target	77.90	**43.40**	81.99	**63.43**	83.16	**51.06**	89.02	**63.57**
	Non-Target	84.43	**81.40**	86.03	**78.54**	83.51	**79.47**	89.79	**76.91**
	Accuracy	81.36	**62.40**	84.01	**70.99**	83.34	**65.27**	89.40	**70.24**
3	Target	81.32	**78.01**	80.48	**77.27**	83.31	**50.40**	82.75	**74.22**
	Non-Target	84.92	**84.09**	86.92	**85.98**	84.44	**81.56**	85.13	**83.89**
	Accuracy	83.12	**81.05**	83.70	**81.63**	83.88	**65.98**	83.94	**79.05**

Table 5.3. Classification rates of three participants obtained with data sets recorded in different sessions. Values in bold represent the accuracies calculated with training data of each experimental session using a classification model trained with epochs collected in another session. The other values indicate the cross-validated results of each experimental session.

5.5.4 Online Tests of the P300-Based BCI

In the online test of the P300-based BCI, the participants were instructed to select in order 40 items indicated by the application. If one option is chosen incorrectly, the participant must continue with the next one. There is not limit time to complete the task. At the end of the experiment, the interface measures the average speed to make one selection with the BCI and the accuracy reached with the trained model. These values help to compare the proposed system with similar approaches reported in other studies. For all the participants, the parameters used in the adaptive control of classification errors are $q^{max} = 0.05$, $n_f^{min} = 2$ and, $n_f^{max} = 10$.

Table 5.4 shows the classification accuracies and the information transfer rates obtained in the online tests of the P300-based BCI. The selection times (without considering resting times) of each participant are shown in Table 5.5. Nine out of the ten participants reached accuracies of 100%. Only one subject had one error. The average selection time was 7.63 seconds, which means that if the stimulation time is 150 ms, the subjects have to count around ten times to select one option. However, fifty percent of the participants only need to count seven flashes to select one command. The average minimum selection time is around 3.887 seconds, which corresponds to 5 flashes, while the maximum selection time is 20.30 seconds (27 flashes). The peak performance obtained by the system was 27.24 bit/min, which corresponds to 73% of the maximum theoretical information transfer rate that this system can achieve (37.08 bit/min). The average performance was 19.38 bit/min, but six participants obtained ITRs above 20 bit/min.

Comparing these results with other BCIs, a similar performance has been reported in recent P300 spellers with healthy participants. In (Käthner, Kübler, & Halder, 2015), the authors obtained an average accuracy of 94% and transfer rates of 15.5 bits/min. Here, faster rates were only possible in single trail classification ($m = 1$ and $n = 1$), but in this case, the accuracies were below 65%. In another

Subject	Accuracy (%)	B_t	L_t	ITR
1	97.50	2.1033	9.8284 ± 2.0783	20.678 ± 4.3713
2	100.0	2.3219	8.7440 ± 5.2966	20.303 ± 12.299
3	100.0	2.3219	9.9661 ± 4.7553	23.141 ± 11.041
4	100.0	2.3219	9.9595 ± 6.1691	23.125 ± 14.324
5	100.0	2.3219	11.407 ± 4.7115	26.486 ± 10.940
6	100.0	2.3219	3.2734 ± 2.1660	7.6005 ± 5.0293
7	100.0	2.3219	6.2578 ± 4.6655	19.513 ± 10.833
8	100.0	2.3219	11.732 ± 4.7802	27.240 ± 11.099
9	100.0	2.3219	3.1517 ± 2.0622	7.3180 ± 4.7882
10	100.0	2.3219	7.9262 ± 5.3689	18.404 ± 12.466
Mean	99.75	2.3000	8.2245 ± 4.2053	19.3809 ± 9.7191

Table 5.4. Classification accuracies and information transfer rates obtained in the online test of the P300-based BCI. The selection speed L_t was calculated as the average time expended to detect an ERP plus the resting time.

Subject	Minimum	Maximum	Mean	SD
1	4.5250	7.1770	5.1048	0.5457
2	3.1930	17.605	5.8619	3.5143
3	3.1610	9.1720	5.0204	1.8314
4	3.0960	19.255	5.0244	3.2684
5	3.1600	6.4110	4.2601	1.2339
6	5.8190	45.478	17.340	11.951
7	4.0780	19.456	6.1397	2.9703
8	2.9650	7.6200	4.1144	1.1723
9	5.9910	58.228	18.038	12.011
10	2.8810	12.679	5.3987	2.9077
Mean	3.8870	20.3081	7.6302	4.1406

Table 5.5. Selection times in seconds of each participant in the online test of the P300-based BCI.

published work, the implemented BCI obtained classification rates of 88% after completing 15 flashing cycles, i.e. $m = 15$ (Takano, Hata, & Kansaku, 2011). In contrast, high transfer rates with accuracies above 95% were reported by L. Chen et al. (2015); however, they did not include the resting times between selections in the calculations, so their results are not comparable with the presented in this work. The same occurs in (Lenhardt, Kaper, & Ritter, 2008), where it is shown an interface that selects 4.41 letters correctly per minute. The BCI described by (Käthner et al., 2015) could select 9.93 letters per minute correctly using single trial classification (around 35% of errors), whereas the proposed system can select without mistakes on average 6.95 symbols per minute. However, the number of

options used in the online test is only 5.

As a conclusion, the information transfer rates and selection times presented in this subsection are highly competitive even though the system was configured to obtain the best possible accuracies. The adaptive strategy to determine the number of flashes worked as expected, providing an easy way to control the misclassification rate.

5.6 Tests of the Passive Detection of Machine Errors

The automatic detection of machine errors has the potential of solving the problem of how to alert the system when the BCI has selected a wrong option. With such a tool, the user would not need to carry out any action to correct the application, and the costs of misclassifying EEG signals would be reduced significantly. Ideally, the subject would only need to select another command when the system identifies ErrP components elicited by the perception of an incorrect selection. With the aim of investigating if it is possible to detect error-related activity with the proposed system, five healthy participants (M = 3, F = 2) trained the ErrP-based BCI with the routine described in this Chapter. In total, each training set consisted of 150 epochs of the correct class and 100 trials of the incorrect class. This section summarizes the results of the offline tests performed with the calibration data and the accuracies reached in online conditions.

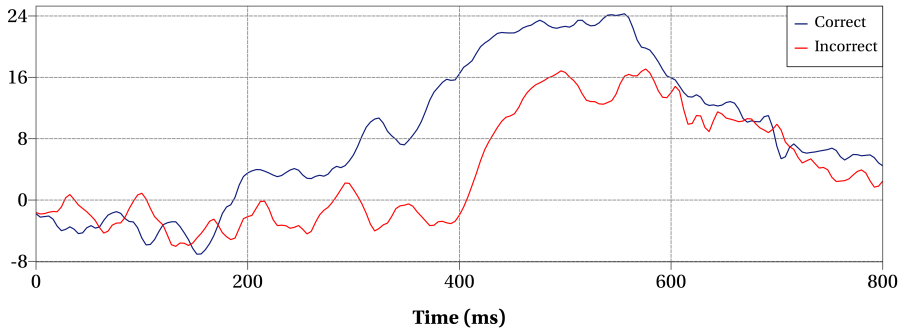
5.6.1 Pointwise Comparison of Correct and Incorrect ERPs

Unlike P300-based BCIs, the two conditions classified in an ErrP detector are defined by time-locked activity to the stimulus presentation, so that the BCI must distinguish between two types of event-related potentials that differ in amplitude and phase. This is because the perception of any machine selection regardless of whether it is correct or incorrect elicits ERP components that can be observed in the EEG. The characteristics of the ERPs of the correct and incorrect conditions can be determined by the standard EEG analysis and the pointwise comparison between both classes.

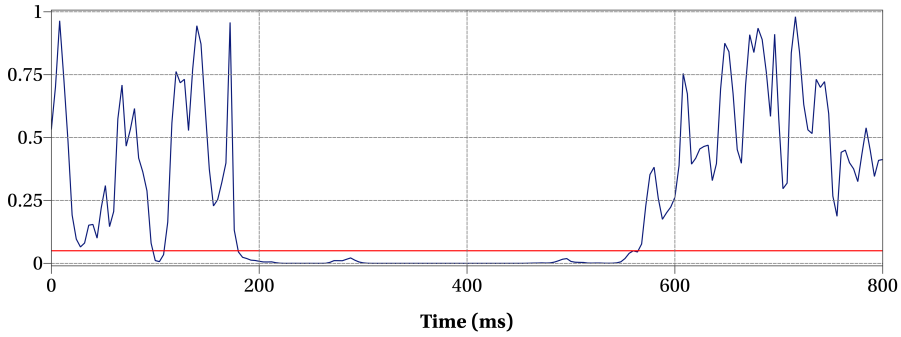
Figure 5.23 illustrates the grand averages of the correct and incorrect conditions at C3 for one participant. The p-values of the pointwise comparison at the same electrode position are also shown in the same figure. In this example, the peak of the grand average of the incorrect class appears 400 ms after the stimulus presentation. In contrast, the event-related potential of the correct class has a stronger positive peak that starts to rise close to the 200 ms. The lowest p-values are observed between 200 and 500 ms after the stimulus presentation. A similar pattern can be observed in other frontocentral channels, as it is illustrated in the topographic plots shown in Figure 5.24.

The maps of significant differences are shown in Figure 5.25. Both the uncorrected and corrected differences are observed predominantly in the frontocentral area after the 200 ms. However, it is interesting to point out that there are some differences in the uncorrected map close to the 100 ms, which may correspond to the early processing of the incorrect stimulus. Unfortunately, the corrected map does not show these significant differences, probably because the sample size is not enough to reach a significant level below the corrected p-value.

Finally, the significance level of the proportion of significant differences was calculated by a permutation test. The results are shown in Table 5.6. All the participants exhibited significant differences

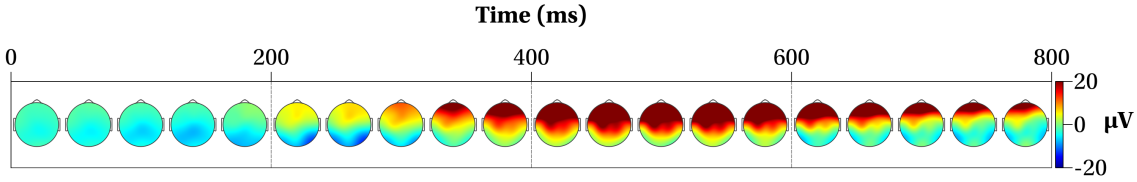


(a) Grand averages.

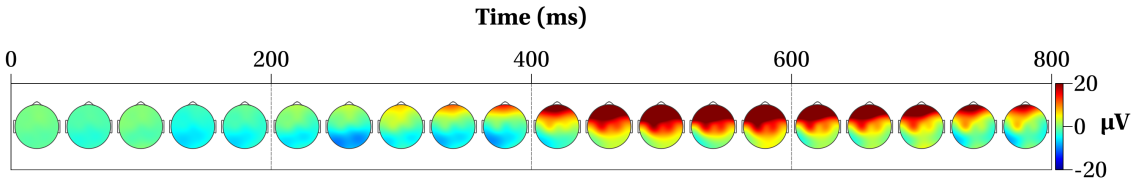


(b) Two-tailed p-values.

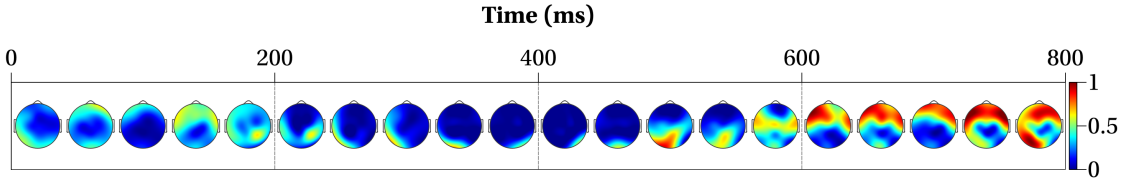
Figure 5.23. Comparison between the correct and incorrect ERPs for one subject at channel C3. The grand averages of both conditions are shown in (a). The p-values obtained with the pointwise testing procedure at the same electrode position are illustrated in (b). In this chart, the blue line represents the p-values, and the red line indicates a significance level of 0.05.



(a) Grand averages of the correct class.

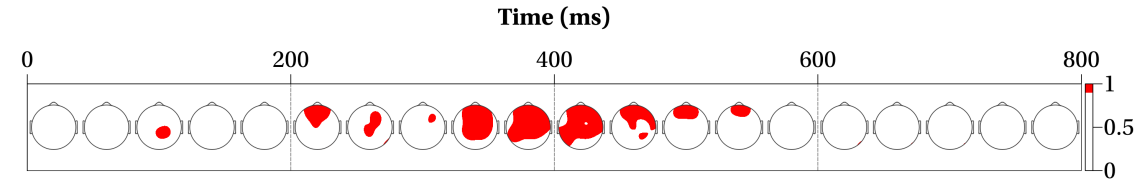


(b) Grand averages of the incorrect class.

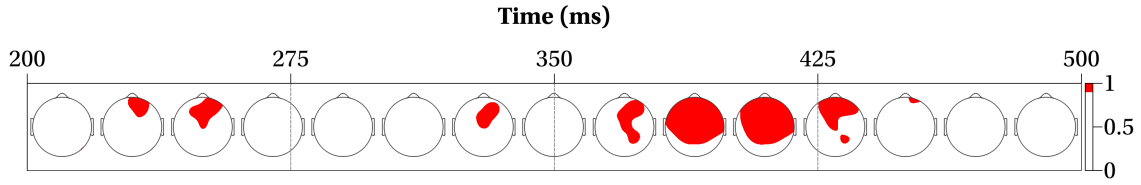


(c) Two-tailed p-values.

Figure 5.24. Grand averages of one subject for the correct (a) and incorrect (b) classes. The p-values of the pointwise hypothesis tests are shown in (c).



(a) Uncorrected significant differences.



(b) Corrected significant differences.

Figure 5.25. Significant differences between the correct and incorrect classes for one participant. The uncorrected map of differences for $\alpha = 0.05$ is shown in (a). The corrected map for $FDR = 0.05$ is illustrated in (b). The heads in (b) represent the average activity of five consecutive time points.

between the correct and incorrect conditions, which means that the system can measure error-related activity.

Subject	Proportion of Significant Differences (%)	P-Value
1	34.69	< 0.001
2	49.44	< 0.001
3	61.28	< 0.001
4	44.00	< 0.001
5	43.39	< 0.001

Table 5.6. Proportion of significant differences between correct and incorrect responses of each participant.

5.6.2 Comparison of Feature Selection Methods

For the tests presented in this subsection, a FIR filter with a bandpass from 4 to 14 Hz was applied to the raw potentials. After this operation, the data was split in epochs of 200 ms of pre-stimulus activity and 800 ms of post-stimulus data. All the contaminated trials were rejected before training and evaluating the classification models. LDA and the univariate filter (combined with the Fisher's separability criterion for feature scoring) were used to obtain the classification models. No more than 200 features were included in the feature sets. All the computed characteristics were obtained only from the nine sensors selected for this BCI (see figure 5.1b at the beginning of this chapter). This electrode arrangement does not include the frontal positions FP1 and FP2, because they exhibit a considerable amount of high amplitude artifacts.

Figure 5.26 shows the classification rates obtained with different feature extraction methods. The xDAWN algorithm is not included in this comparison because it is used when the EEG contains only

one type of ERP response. In contrast, the training data contains a mix of three types of ERPs: P300 responses, ERPs for correct selections and error-related activity.

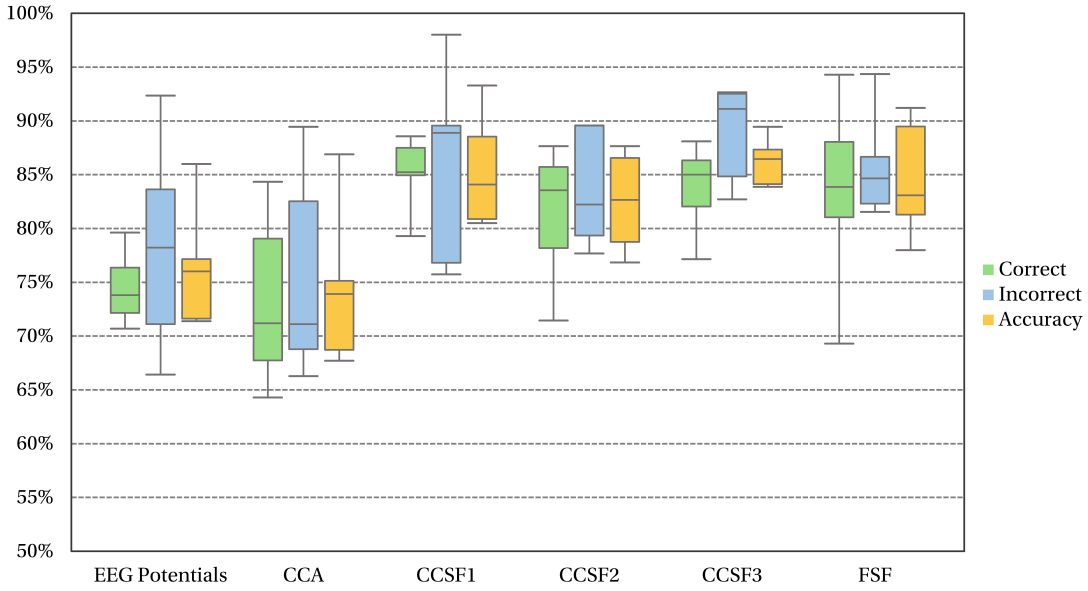


Figure 5.26. Classification rates obtained with different feature extraction methods for ErrP-based BCI.

As can be seen from the chart, the best results were achieved with spatial filtering techniques. CCSF3 obtained the best average accuracy (86.24%), followed by CCSF1 (85.80%) and FSF (84.60%). At the same time, the corresponding p-values of all the classification accuracies calculated in this test are below 0.001 for 1000 permutations, which demonstrates that the system can classify ErrP data successfully.

It is interesting to notice that in this case, the classifier obtains a better performance with the method CCSF3 than with the method CCSF2. This may be explained by the fact that in this BCI, both conditions have ERP responses, so that spatial filters built with information of both classes may produce better characteristics than filters that only consider one class. On the contrary, in the P300-based BCI, only one condition has associated time-locked activity, so that the method CCSF2 may be preferred in the latter case.

The effect of using re-reference techniques in the preprocessing stage is illustrated in Figures 5.27 and 5.28. The methods that achieved the best accuracies after applying the common-average reference are CCSF3 (88.58%), CCSF1 (86.77%) and CCSF2 (83.38%). Likewise, the same techniques produced the best accuracies with the surface Laplacian (88.77%, 86.33%, and 84.83%). The BCI obtained better classification rates with all the methods after applying the re-referencing techniques, except FSF. In the latter case, the system lost around three percentage points with CAR and one percentage point with the surface Laplacian. This penalty in performance is not notable for FSF so that it can be concluded that the common average reference and the surface Laplacian can be used in the preprocessing stage to improve the quality of the computed characteristics.

The artifact subspace reconstruction method was also tested to investigate if this strategy is suitable to reject artifacts without sacrificing accuracy. Figure 5.29 shows the accuracies calculated in this test. The methods that obtained the best classification rates are CCSF3 (86.41%), CCSF2 (82.31%)

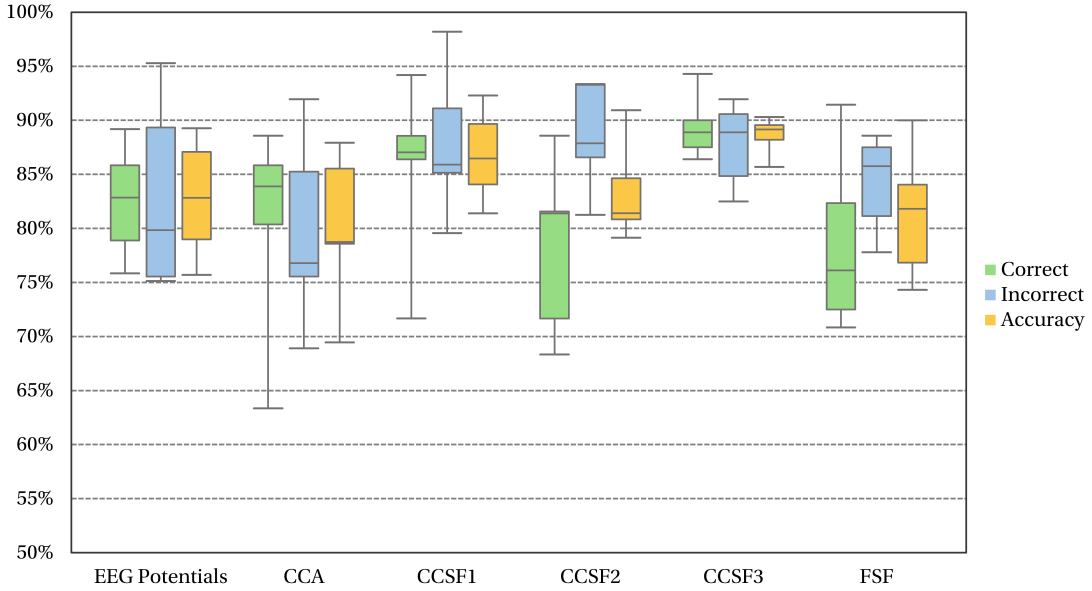


Figure 5.27. Classification rates obtained when the common average reference was applied in the preprocessing stage.

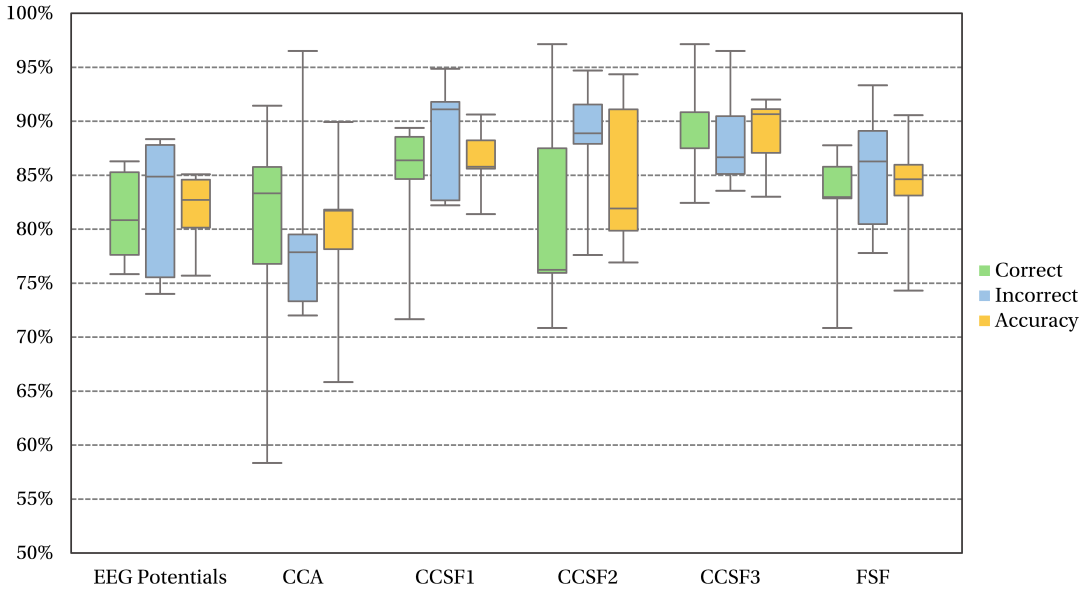


Figure 5.28. Classification rates obtained when the surface Laplacian was applied in the preprocessing stage.

and CCSF1 (80.56%). In general, all the techniques achieved a lower performance, which indicates that the ASR method had a negative effect in classifying ErrP data. Perhaps only in those cases when the subject is producing high amplitude artifacts very often, it may be recommended to include the artifact subspace reconstruction method to clean the signals.

Figure 5.29 summarizes the results of the tests described in this subsection. From this chart, it can be concluded that the methods CCSF3 and CCSF1 are good choices for classifying error-related potentials. At the same time, re-referencing can be included in the preprocessing stage, even though the gain in performance is not notable for this two feature extraction methods.

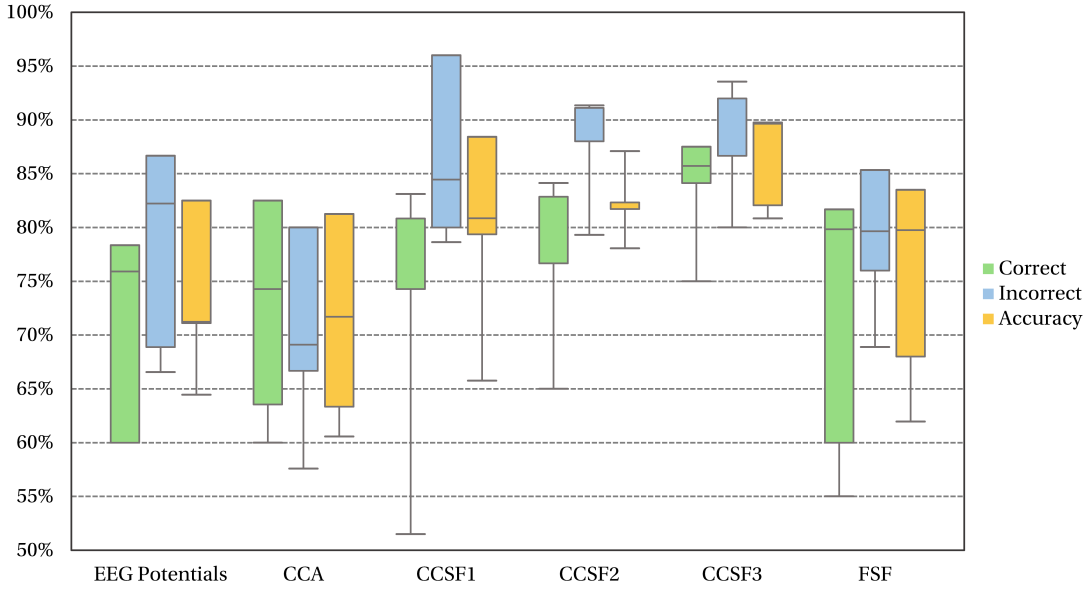


Figure 5.29. Classification rates of the ErrP-based BCI when artifacts were rejected with the ASR method.

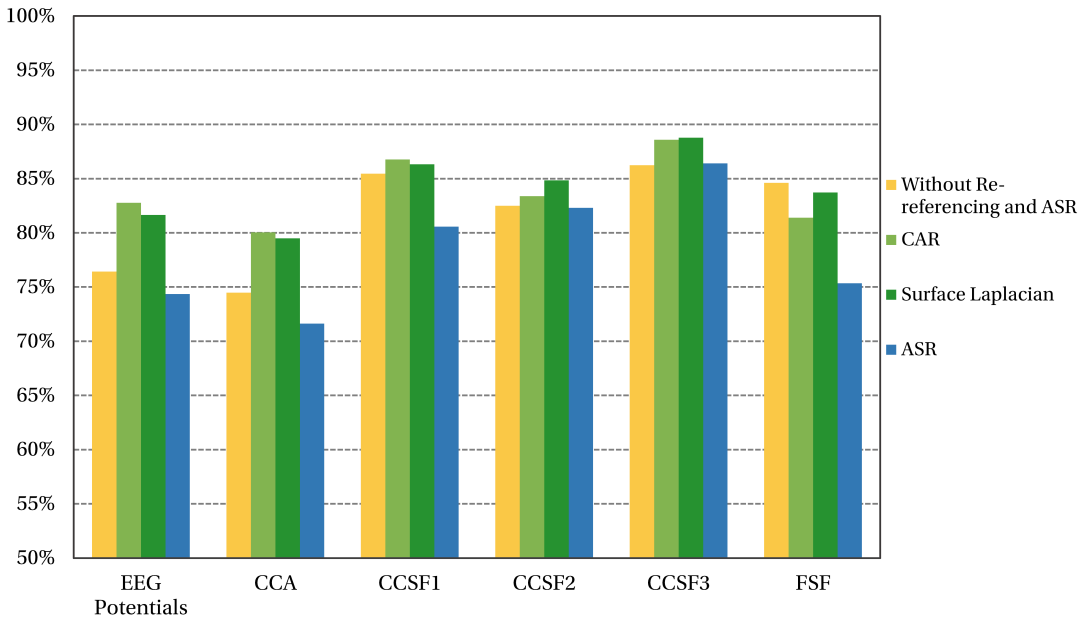


Figure 5.30. Comparison of preprocessing algorithms in the ErrP-based BCI.

5.6.3 Online Tests of the ErrP Detector

The online test of the ErrP detector is similar to the training routine of this BCI. The P300 interface operates in normal conditions, but the system selects randomly another option to elicit error-related activity. The purpose of this experiment is to measure how many times the interface classifies the correct and incorrect conditions successfully in an online run. The complete test consists of 50 epochs, 30 of the correct class and 20 of the incorrect class. The classification rates of the five participants (shown in table 5.7) were obtained with the CCSF3 method for feature extraction, combined with LDA and the univariate filter described in the previous section.

Subject	Correct (%)	Incorrect (%)	Accuracy (%)
1	80.00	80.00	80.00
2	86.67	85.00	85.84
3	83.34	75.00	79.17
4	83.34	80.00	81.67
5	73.34	80.00	76.66
Mean	81.33	80.00	80.67

Table 5.7. Classification accuracies obtained with the ErrP detector.

The classification accuracies obtained in the online tests are similar to the values calculated in the cross-validation of the classification models. These results demonstrate that the system can classify error-related activity during normal operation of the BCI with an accuracy close to 80% with healthy people. Unfortunately, this classification rate is not enough to improve the P300 interface, because there is a high chance to observe classification errors that interrupt the BCI. For all the participants, the ErrP detector became annoying at some point because they were accustomed to operating the P300 interface without problems. They reported that they prefer not to use this passive system because they do not consider that the interface works better with this BCI.

It is important to point out that the ErrP detector may be useful when the accuracy of other BCIs is low so that they require constant supervision to correct mistakes. In these cases, it is better to select several times the same option rather than performing more actions to correct the errors of the interface. Schmidt, Blankertz, and Treder (2012) reported that users with a low performance to operate a P300 speller obtained better results when errors are identified with a passive BCI.

5.7 Conclusion

This chapter described the ERP-based BCI included in the hybrid system to detect P300 evoked potentials and error-related activity. An integral evaluation of all the processing stages of this subsystem was presented to determine the optimal configuration of this BCI.

The LDA classifier combined with stepwise regression proved to be a good choice when spatial filtering is used to extract ERP features. Meanwhile, according to the results obtained in the offline tests, spatial filtering based on the canonical correlation analysis represents an efficient method to characterize EEG data in ERP experiments. Re-referencing methods can also be included in the preprocessing stages, but the ASR method for artifact suppression may cause a loss in performance in the ErrP detector.

One of the essential components incorporated into the P300 detector is the adaptive mechanism that estimates the number of flashes needed to detect one evoked response. The proposed method only requires the accuracy estimated in the training phase of the machine learning model to find the stopping criterion of the flashing sequence. The online tests of the P300-based BCI proved that the proposed algorithm has an excellent control of classification errors, and allows achieving good selection times. Even users that obtained accuracies below 75% in single-trial classification can operate

the interface with low error rates.

It was also demonstrated in this chapter that the implemented BCI could be used to detect error-related activity. On average, the BCI can discriminate with an accuracy of 80% between correct and incorrect machine selections. The differences between both evoked responses were also measured and tested so that there is enough evidence that the perception of errors is feasible in the implemented experimental setup. However, the participants reported that they prefer not to use this passive system to detect errors in the P300-based BCI because in normal condition, the interface has high accuracies and the false positives of the ErrP detector interrupt some correct selections.

It can be concluded that the implemented ERP-based BCI is an efficient system to detect different types of evoked components. In the wheelchair control, this BCI is used to select target places and basic orders, whereas the passive detection of machine errors has the intention to boost the performance of the P300-based interface.

Chapter 6

Auto-Calibration and Adaptation of the P300-Based BCI

Brain signals are in essence non-stationary. The patterns observed in training sessions under controlled conditions may be different from those measured outside the laboratory, and even changes in concentration and motivation alter the brain activity. In consequence, machine learning models obtained in calibration routines may not be optimal in every possible situation. The system needs to learn more about the subject during the online sessions and adapt the classification rule with the aim of avoiding a significant decrease in performance.

The proposed BCI incorporates a mechanism for the continuous adaptation of the P300-based BCI. The machine learning model is periodically retrained without user's intervention using the last recorded observations. This chapter describes this adaptation procedure and includes several tests that validate this method for online operation.

At the same time, if the system has not been trained before the online run, the BCI uses the available data to find the classification model automatically without the need for a calibration routine. This chapter also details the method implemented for auto-calibrating the machine learning model of the P300-based BCI. This algorithm is based on quantifying the significance level of event-related responses after the stimulus onset. This procedure allows detecting a set of observations that on average have strong ERP components, which are classified in the target group. On the contrary, if the analyzed epochs do not exhibit event-related activity, they are classified as non-target epochs.

6.1 Automatic Detection of Event-Related Activity

The core functionality of the proposed auto-calibration procedure consists of identifying samples that correspond to the target and non-target classes without user's intervention. This task is achieved by testing the average response of each possible option against the hypothesis that the event (the flashing) does not elicit changes in the EEG with respect to the pre-stimulus segment. The assumption in this method is that the target stimulus produces a greater proportion of significant changes than any non-target option.

Let $X = \{X_1, X_2, ..., X_{n_k}\}$ be a set of n_k trials free of artifacts for one particular option of the user interface. Let $X_k = [x_e^k(t)]_{n_e \times n_t}$ be the matrix with the time series of the *k-th* epoch. The following test

is used to determine the class of X:

1. The trials in X are baseline corrected using Expression 5.1 to obtain the set of epochs $\tilde{X} = \{\tilde{X}_1, \tilde{X}_2, ..., \tilde{X}_{n_k}\}$.
2. The normalized average response $Y = [y_e(t)]_{n_e \times n_t}$ is calculated as

$$y_e(t) = \frac{\mu_e(t)\sqrt{n_k}}{\sigma_e(t)}, \tag{6.1}$$

where

$$\mu_e(t) = \frac{1}{n_k}\sum_{k=1}^{n_k} \tilde{x}_e^k(t), \tag{6.2a}$$

$$\sigma_e^2(t) = \frac{1}{n_k - 1}\sum_{k=1}^{n_k}\left(\tilde{x}_e^k(t) - \mu_e(t)\right)^2. \tag{6.2b}$$

3. The level of significance of each normalized response $y_e(t)$ is estimated by using the non-parametric kernel density estimation method (Bowman & Azzalini, 1997). The probability values $p_0(y_e(t))$ under the null distribution H_0 are calculated as

$$p_0\left(y_e(t)\right) = \int_{-\infty}^{y_e(t)} \frac{1}{n_{pre}} \sum_{l \in T_{pre}} w\left(x - y_e(l), h_e(l)\right) dx\ , \tag{6.3}$$

where $w(x,h)$ is a kernel function and h is the bandwidth or smoothing parameter. The Gaussian kernel (used in this test) is defined by

$$w(x,h) = \frac{1}{\sqrt{2\pi}h} e^{-\frac{x^2}{2h^2}}. \tag{6.4}$$

In the BCI:

$$h_e(l_1) = h_e(l_2) = h_e. \tag{6.5}$$

The smoothing parameter is selected according to the *mean integrated squared error* (MISE) estimator for Gaussian kernels and Gaussian distributions (Bowman & Azzalini, 1997):

$$h_e = \left(\frac{4}{3n_{pre}}\right)^{0.2} \sigma_e^{pre}, \tag{6.6}$$

where σ_e^{pre} is an estimate of the sample standard deviation of the pre-stimulus segment for the *e-th* channel. A robust estimator of this parameter is given by

$$\sigma_e^{pre} = \frac{\text{median}\left\{\left|y_e(t) - \text{median}\left\{y_e(t), \forall\, t \in T_{pre}\right\}\right|\right\}}{0.6745}. \tag{6.7}$$

4. The p-values of the normalized variables are calculated as

$$p_v(y_e(t)) = 2\begin{cases} p_0\left(y_e(t)\right) & \text{if} \quad p_0\left(y_e(t)\right) <= 0.5, \\ 1 - p_0\left(y_e(t)\right) & \text{otherwise.} \end{cases} \tag{6.8}$$

5. The proportion q of accepted alternative hypotheses is quantified and compared against threshold values q_{target} and $q_{non\text{-}target}$ ($q_{target} \geqslant q_{non\text{-}target}$). Given a significance level α, the class

$C(X)$ of the set X is determined by

$$C(X) = \begin{cases} target & \text{if} \quad q > q_{target}, \\ non\text{-}target & \text{if} \quad q < q_{non-target}, \\ unknown & \text{otherwhise}, \end{cases} \tag{6.9}$$

where:

$$q = \frac{1}{n_e n_{pos}} \sum_{e=1}^{n_e} \sum_{t \in T_{pos}} I\left(p_v\left(y_e(t)\right) \leq \alpha\right), \tag{6.10a}$$

$$I\left(p_v\left(y_e(t)\right) \leq \alpha\right) = \begin{cases} 1 & \text{if} \quad p_v\left(y_e(t)\right) \leq \alpha, \\ 0 & \text{otherwhise.} \end{cases} \tag{6.10b}$$

If the system does not have a priori information about the user, the parameters q_{target} and $q_{non\text{-}target}$ are set in advance according to previously estimated values using different training sets. However, if the system was trained before with labeled data, the BCI uses a *subsampling* method (Politis, Romano, & Wolf, 1999) to estimate the distributions of the proportions of positive tests for the target and non-target conditions. In this algorithm, new sets of trials smaller than the original group of target epochs are drawn from this class without replacement. Then, for each new set $X^*(i) \in \{X^*(1), X^*(2), ..., X^*(n)\}$, where n is the number of random samples and $i \in \{1, 2, ..., n\}$, the proportion q_i^* is calculated to estimate the distribution of this variable for the target class by using the kernel density method. At the same time, the described procedure is applied to the observations of the non-target class to obtain the distribution for the proportions of the non-target group. Finally, the threshold limits are set to satisfy that the probability of misclassifying a proportion q is lower than q_r. That is to say

$$q_{target} = \inf\left\{q : p\left(Q \geqslant q | H_{non\text{-}target}\right) < q_r\right\}, \tag{6.11a}$$

$$q_{non\text{-}target} = \sup\left\{q : p\left(Q \leqslant q | H_{target}\right) < q_r\right\}, \tag{6.11b}$$

where H_{target} and $H_{non\text{-}target}$ denote the distributions of the proportions for the target and non-target classes.

This methodology was also adapted to test other representations of the original time series. For instance, the epochs can be band-pass filtered with a bank of complex filters tuned at different frequencies to obtain their time-frequency decomposition. Let $Z_k = [z_{e,b}^k(t)]_{n_e \times n_b \times n_t}$ be the time-frequency representation of the *k-th* epoch of the training set, where n_b is the number of frequency bands and $b \in \{1, 2, ..., n_b\}$. The local log-amplitude $s_{e,b}^k(t)$ and phase $\phi_{e,b}^k(t)$ of each channel, frequency component and time point are

$$s_{e,b}^k(t) = \ln\left(\sqrt{\mathrm{Re}\left(z_{e,b}^k(t)\right)^2 + \mathrm{Im}\left(z_{e,b}^k(t)\right)^2}\right), \tag{6.12}$$

$$\phi_{e,b}^k(t) = \tan^{-1}\frac{\mathrm{Im}\left(z_{e,b}^k(t)\right)}{\mathrm{Re}\left(z_{e,b}^k(t)\right)}, \tag{6.13}$$

where $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ represents the real and imaginary parts of a complex number z respectively.

The base-line corrected epochs $\tilde{S} = [\tilde{S}_1, \tilde{S}_2, ..., \tilde{S}_{n_k}]$ of the log-amplitude values and the normalized average response $\tilde{Y} = [\tilde{y}_{e,b}(t)]_{n_e \times n_b \times n_t}$ of these observations are calculated in a similar way as in the case of the EEG potentials. Likewise, the significance level of each $\tilde{y}_{e,b}(t)$ is obtained by using the kernel density method. The time series of each channel and band are compared against their respective pre-stimulus distribution to obtain the probability values $p_0(y_{e,b}(t))$ and the p-values $p_v(y_{e,b}(t))$. Finally, the proportion of significant increments and decrements of log-amplitudes after the stimulus onset is tested against threshold values q_{target} and $q_{non\text{-}target}$ to determine the class of the original set of observations.

A significant increment of log-amplitude with respect to the pre-stimulus segment is also known as *event-related synchronization* (ERS) (G. Pfurtscheller & Lopes da Silva, 1999). Similarly, a significant decrement of log-amplitude is called *event-related desynchronization* (ERD). The presented methodology (introduced originally by Marroquin et al. (2004)) allows analyzing two components in the oscillatory activity that explain event-related power changes: the *evoked response* and the *induced activity*. The evoked response is phase-locked to the stimulus presentation so that it can be obtained by averaging the baseline corrected potentials of multiple trials. In contrast, the induced activity is not phase-locked to the stimulus onset, but the power of the different oscillatory components are time-locked. Since both types of activities produce changes in the power content of the EEG signals, the time-frequency representation is a suitable technique to analyze simultaneously evoked and induced components of event-related activity.

As explained above, the evoked responses are in essence the result of phase-locked oscillatory components, so that if there is a stimulus that elicits an ERP, a similar phase must be observed in multiple trials of the same type. On the contrary, the absence of an ERP means that the phase of each oscillatory component may take any value along trials, following a uniform distribution around the circle without any preference. The characteristics of the distribution of the phase values of each oscillatory component may provide information about the type of brain activity observed by the EEG equipment.

The phase values $\phi^k_{e,b}(t)$ are analyzed using a different methodology than the presented here for significant changes of log-amplitudes. The baseline correction is not needed, and the test against the pre-stimulus segment does not apply in this case. Instead, the phases after the stimulus onset are tested against the hypothesis of uniformity around the circle. One method to check the randomness of circular data is the Hodges-Ajne test (Mardia & Jupp, 2000), which consists of counting the maximum number of samples that fall in the arc of one of the possible semicircles. If this count is large, the null hypothesis of uniformity is rejected. For each channel, band and time point, the corresponding p-value is obtained and then the proportion of positive tests is compared against pre-defined threshold values.

The BCI uses a bank of quadrature filters to calculate the time-frequency decomposition of the EEG time series. This approach allows centering the filters at arbitrary frequencies and defining the bandwidth of each extracted component freely. A simple way to find a filter bank is to compute a sampled version of the Gabor filters (Gabor, 1946), which were defined originally in the continuous time-domain. Given a sample rate f_s in samples/s and a tuning frequency f_0 expressed in Hz, the sampled version of a Gabor filter is given by

$$g_{f_0,\sigma_t}(t) = \frac{1}{M_t} e^{-\frac{t^2}{2\sigma_t^2}} \left(\cos\left(\frac{2\pi f_0}{f_s} t\right) + i \sin\left(\frac{2\pi f_0}{f_s} t\right) \right), \tag{6.14}$$

where σ_t is the standard deviation in samples of the Gaussian kernel, and M_t is a normalization parameter. In the frequency domain, the filter defined by Equation 6.14 approximates to another sampled Gaussian function

$$G_{f_0,\sigma_f}(f) \approx \frac{1}{M_f} e^{-\frac{(f-f_0)^2}{2\sigma_f^2}}, \tag{6.15}$$

where f represents the frequency in Hz, M_f is a normalization parameter, and σ_f is the standard deviation in Hz of the sampled Gaussian function. The parameters of the filter in the frequency domain are obtained by

$$\sigma_f = \frac{f_s}{2\pi\sigma_t}, \tag{6.16a}$$

$$M_f = \frac{1}{M_t} \sum_{t=-\infty}^{\infty} e^{-\frac{t^2}{2\sigma_t^2}}. \tag{6.16b}$$

Each filter can be specified in terms of its gain (the inverse of M_f) and bandwidth (the length of the range of frequencies for which the magnitude of the frequency response is above $0.5M_f$). Given a bandwidth f_w expressed in Hz, the standard deviation is obtained by

$$\sigma_f = f_w \sqrt{\frac{1}{\ln(4)}}. \tag{6.17}$$

When f_0 is close to zero, the frequency response of a sampled Gabor filter is not zero at negative frequencies, which violates the constraint of a quadrature filter. One alternative is to use *sinusoidal quadrature filters* (Guerrero, Marroquin, Rivera, & Quiroga, 2005), which are defined in the frequency domain for the n-point Fourier transform by the following expression:

$$S_{f_0,f_w}(f) = \begin{cases} \frac{M}{2}\left(1+\sin\left(\frac{f_l+2(f-f_0)}{2f_l}\pi\right)\right) & \text{if} \quad f \in [f_0-f_l, f_0], \\ \frac{M}{2}\left(1+\sin\left(\frac{f_u+2(f-f_0)}{2f_u}\pi\right)\right) & \text{if} \quad f \in (f_0, f_w+f_0], \\ 0 & \text{otherwise}, \end{cases} \tag{6.18}$$

where M is the filter gain, and

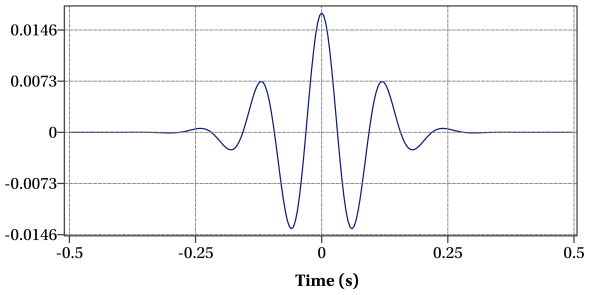
$$f_l = \min\left(\frac{n}{2}f_w, f_0\right), \tag{6.19a}$$

$$f_u = \min\left(\frac{n}{2}f_w, \frac{n}{2}-f_0\right). \tag{6.19b}$$

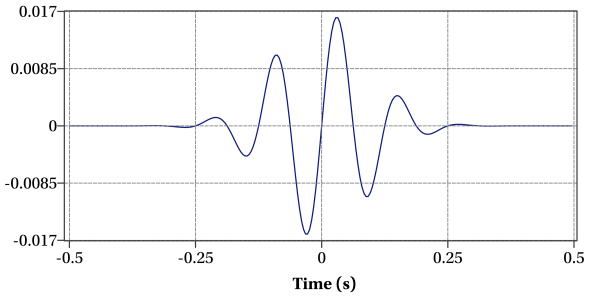
The time-frequency decomposition of the EEG time series is performed using sinusoidal quadrature filters of unitary gain. The bandwidths are set to 4 Hz, and the tuning frequencies are $4, 6, 8, \ldots, 30$ Hz. Figure 6.1 shows the filter centered at 8 Hz.

6.1.1 Comparison of Methods for Automatic ERP Detection

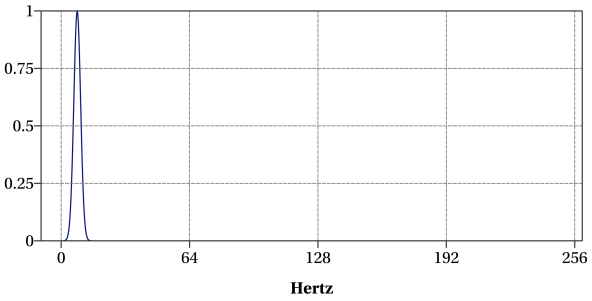
Up to this point, three different alternatives have been described to identify the class of a set of observations obtained from a P300 experiment. In one of these strategies, the grand averages are tested against the hypothesis that the stimulus presentation does not elicit event-related potentials. In the second approach, the significance levels of log-power changes are quantified to identify event-related synchronizations and desynchronizations after the stimulus onset. Finally, in the third method, the



(a) Real part in the time-domain.



(b) Imaginary part in the time-domain.



(c) Frequency response.

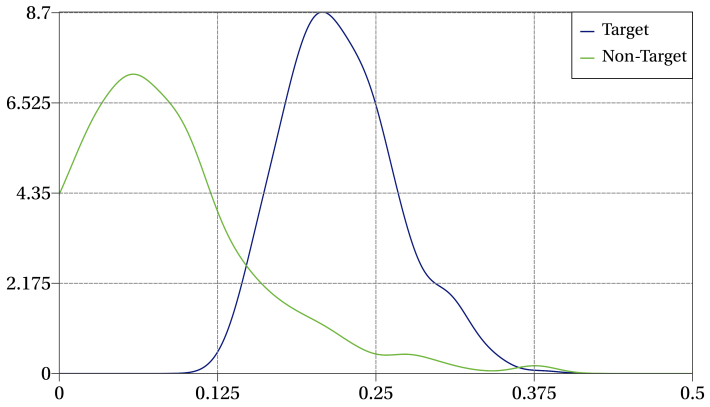
Figure 6.1. Sinusoidal quadrature filter centered at 8 Hz of unitary gain and a bandwidth of 4 Hz. (a) and (b) illustrate the filter in the time domain. (c) represents the frequency response of the filter.

phase values of the oscillatory components extracted by any time-frequency decomposition method are tested against the hypothesis that they are distributed uniformly around the circle. These three methodologies require threshold values to determine if the calculated proportions of positive tests for a given significance level α correspond to the target or non-target groups.

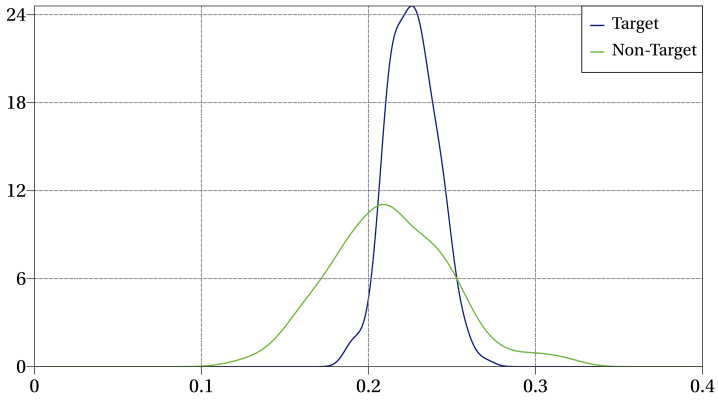
The training sets collected in the tests of the P300-based BCI were used to compare the performance of the methods implemented for the automatic classification of ERP responses. The training data of each participant were processed to estimate the distributions of the proportions of positive tests for different levels of subsampling. In this experiment, the effect of having small samples was also quantified to determine the minimum number of epochs of each class that is needed to classify groups of trials correctly.

Figure 6.2 illustrates the distributions obtained with 1000 random subsamples of the training set of one subject. In this example, the target and non-target groups consisted of 250 epochs selected without replacement. The charts show that the distributions corresponding to the test of uniform phase values do not overlap so that in this case, it is easy to find threshold values that allow classifying

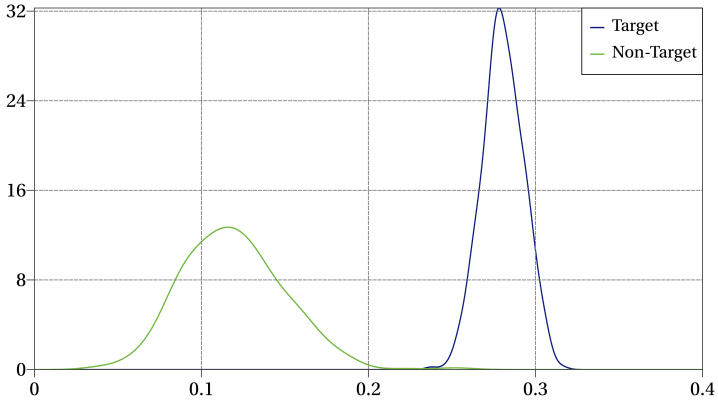
with a high accuracy target and non-target sets of observations. In contrast, the distributions obtained in the pointwise testing of event-related potentials are separated but overlap because of the high variance of both distributions. Finally, in the case of ERSs and ERDs, the distribution of the target class is more concentrated than the other one, but they overlap in all the domain. The modes of both distributions are very close, and the non-target distribution exhibits a high variability.



(a) Event-related potentials.



(b) Event-related synchronizations and desynchronizations.



(c) Phase values.

Figure 6.2. Distributions of the proportions of positive tests calculated with different variables that characterize the post-stimulus activity. The plots were obtained with 1000 subsamples of 250 epochs for each class.

The same pattern described above was observed in almost all the training sets tested in this

experiment. In general, the distributions obtained with the instantaneous phases show a better separation between classes than the other alternatives. These results can be verified by the *overlapping coefficients* (OVL) shown in Table 6.1. This metric is used to measure the simultaneous area under two probability distributions $h_1(x)$ and $h_2(x)$, that is to say

$$OVL = \int_{-\infty}^{\infty} \min\left(h_1(x), h_2(x)\right) dx. \tag{6.20}$$

The table presents the coefficients estimated with 1000 random subsamples of 250 observations of each class. The lowest overlapping coefficients were obtained with the proportions of positive tests calculated with the phase values. Moreover, the boxplots presented in figure 6.3 indicate that the same technique exhibits the lowest variability among subjects. According to the results shown in the table, 250 trials are enough to obtain distributions with low overlapping levels independently of the subject.

Method	**Subject**										
	1	2	3	4	5	6	7	8	9	10	Mean
ERPs	0.5409	0.5022	0.1238	0.3332	0.4611	0.5283	0.0269	0.6117	0.7448	0.1443	0.3721
ERSs and ERDs	0.8121	0.3203	0.6893	0.1253	0.4283	0.5196	0.1688	0.0079	0.6941	0.8042	0.5174
Phase Values	0.1081	0.0021	0.0000	0.0123	0.0147	0.2380	0.0433	0.0079	0.2715	0.0000	0.0698

Table 6.1. Overlapping coefficients obtained with the distributions of the proportions of positive tests in the automatic detection of ERPs.

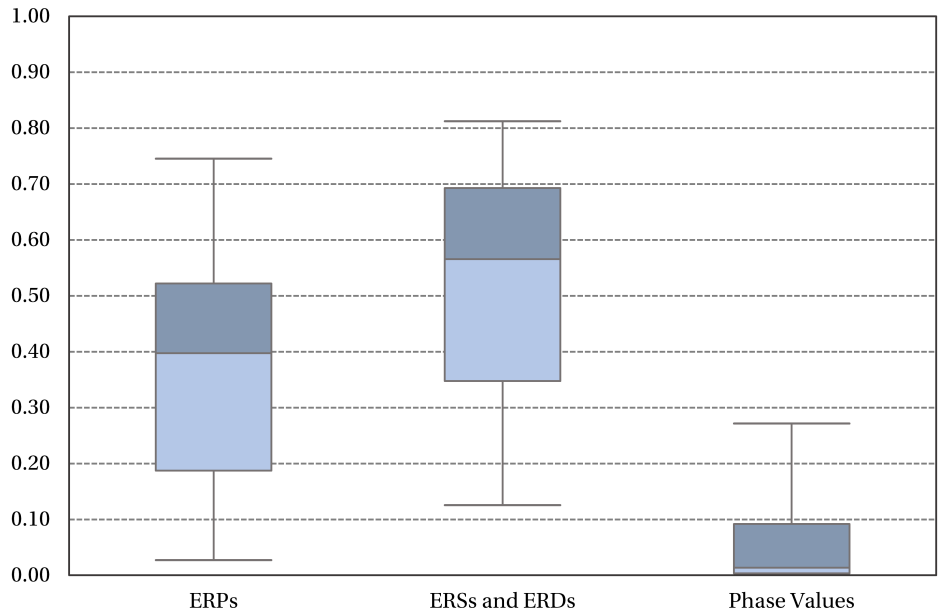


Figure 6.3. Boxplots of the overlapping coefficients calculated with different techniques that identify ERP responses from unlabeled data.

The levels of significance of the comparative study between methods were determined by a Wilcoxon signed-rank test for paired samples (Gibbons & Chakraborti, 2011). The overlapping coefficients obtained with the phase values are significantly lower than the obtained with ERPs ($p = 0.0019$).

The same was observed when the overlapping coefficients calculated with phase values are compared with the results achieved with ERSs and ERDs ($p = 0.00097$). Finally, no significant differences are observed when the method based on ERPs is compared with the method based on power changes ($p = 0.3222$).

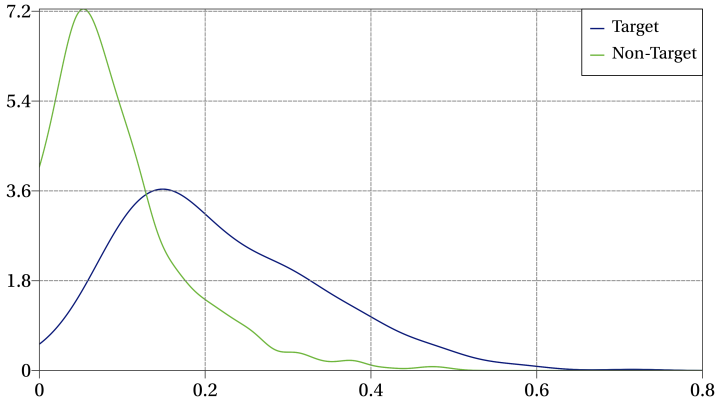
The main advantage of using phase values is that they do not need to be corrected or normalized with respect to the pre-stimulus distribution. The instantaneous phases are independent of the activity registered before the stimulus onset and only noise and artifacts after the stimulus presentation may contaminate the results. On the contrary, the results obtained with the other approaches depend on the pre-stimulus activity. The number of available observations and the length of the pre-stimulus segment determine the characteristics of the test distributions and have a direct influence in the calculated p-values.

Another problem with the methods that are based on significant changes is how the pre-stimulus is defined in the context of the experiment. In a typical ERP-based study, the preparation time and the pre-stimulus segment are long enough to ensure that the subject reaches a particular state of alertness before the stimulus presentation. In this type of experiments, it is assumed that the baseline condition is common for all the trials and exhibits minor changes among epochs. For instance, in the case of event-related potentials, the EEG in the pre-stimulus is represented by oscillations of the baseline state which are canceled out after averaging several epochs. In contrast, in the analysis of ERSs and ERDs, changes with respect to the baseline power are used to characterize the brain mechanisms that process the stimulus. Consecutive epochs overlap in a P300-based BCI so that it is not possible to provide a unique state in the pre-stimulus for all the trials. Sometimes, the pre-stimulus of the non-target condition is contaminated by P300 responses of the target option. Moreover, the post-stimulus of a target epoch may reach the pre-stimulus of another target trial.

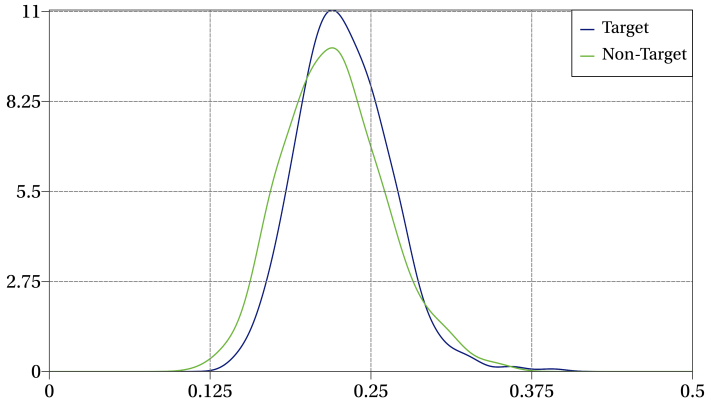
The overlapping between epochs may explain the low performance of the methods based on significant changes with respect to the pre-stimulus activity. As stated before, the post-stimulus of the target option contaminates other epochs, so that it is possible to observe spurious activity that is detected in the test. Also, the ERP responses to target stimuli modify the probability distributions of the pre-stimulus of non-target options, which contributes to the variance of the calculated proportions. However, it is important to say that the test based on ERPs can be used to classify sets of epochs, but the pointwise testing of phase values is a better option. On the contrary, the test based on ERSs and ERDs is not an option for this operation. This technique may be more useful in another experimental setup in which the pre-stimulus and post-stimulus segments are not perturbed by other kinds of brain activity.

The test based on phase values use to provide better results than the other techniques, even when the sample sizes are smaller. Figure 6.4 presents the distributions obtained with 150 epochs. The plots show tat the overlapping between distributions is larger in all the cases, but it is still feasible to use the test based on phase values to classify sets of epochs.

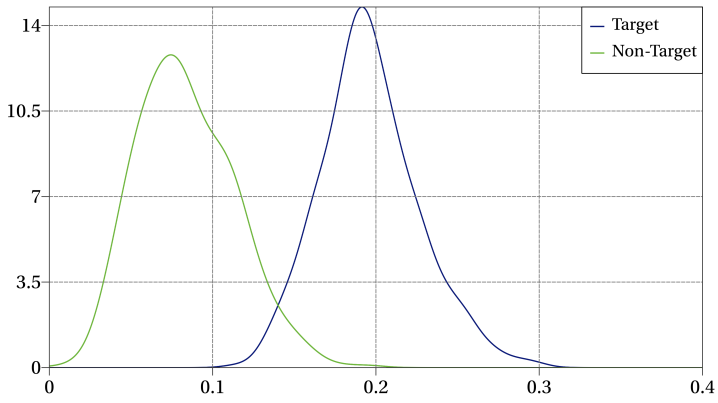
Generally speaking, the smaller is the sample size, the larger is the overlapping between distributions. This fact can be observed in figure 6.5, which presents the distributions of proportions of non-uniform phase values for 50 and 100 epochs. In the case of 100 epochs, it is possible to set $q_r = 0.05$ to have a low error rate without losing the sensitivity to detect the correct class. However, for 50 epochs, the same parameter leads to a low recognition rate, because the majority of the proportions



(a) Event-related potentials.



(b) Event-related synchronizations and desynchronizations.

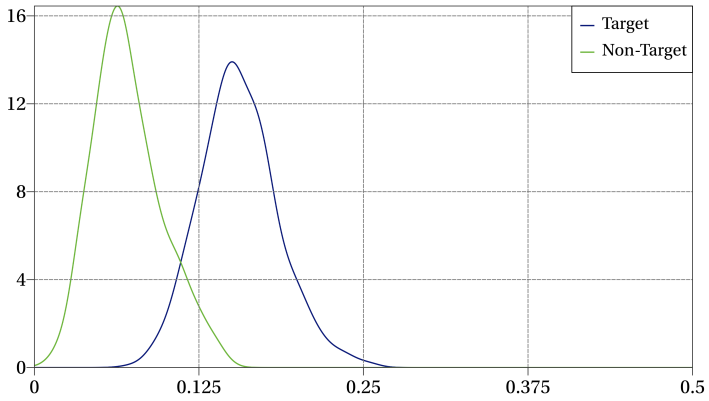


(c) Phase values.

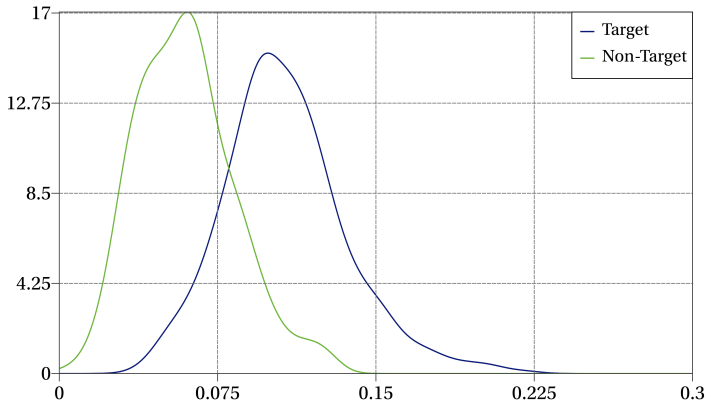
Figure 6.4. Distributions of the proportions of positive tests calculated with 1000 subsamples of 150 epochs for each class.

would be classified in the *unknown* class.

The overlapping coefficients heavily depend on the subject. This relation can be seen in Figure 6.6, which presents the coefficients calculated for three different participants in the test of non-uniform phase values. In this example, one subject has an overlapping below 0.1 for 150 samples, whereas another participant requires more than 300 epochs to reach a similar level. This chart illustrates the importance of having subject-specific data to train the different systems of the BCI. Even the auto-calibration routine requires a certain knowledge about the subject to identify sets of epochs



(a) 100 epochs.



(b) 50 epochs.

Figure 6.5. Distributions of the proportions of non-uniform phase variables obtained with different sample sizes.

correctly.

6.2 Online Training of the P300-Based BCI

In the online operation of the P300-based BCI, there are three different situations in which the machine learning model is adjusted or retrained with new observations. In one case, the classifier works properly, and the subject can generate control signals without problems. However, after several hours of operation, there is a chance to observe a decrease in performance because of changes in attention or motivation levels that perturb the neural activity. Here, the interface uses the classified data of the online run to accumulate new epochs of the target and non-target classes. Then, after getting a minimum of 100 to 300 observations of each class, the system automatically estimates and evaluates a new model. If the accuracies calculated in the cross-validation of the new estimated model for each class are similar or higher than the accuracies of the previous classifier, the new model replaces the old one. On the contrary, if this condition is not met, the new classifier is discarded, and the system continues operating without modifying the P300-based BCI.

The second case is when the subject struggles to select one option of the interface. Typically, if the performance of the machine learning model is low, the system cannot stop the flashing sequence as a consequence of the heuristic used to control false positives. As explained in Chapter 5, the BCI

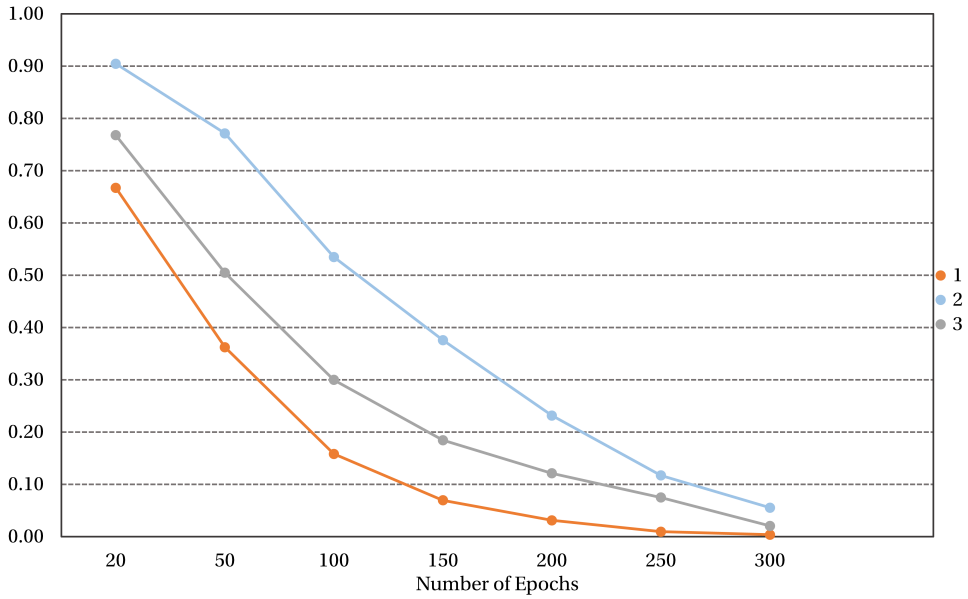


Figure 6.6. Overlapping coefficients for the distributions obtained with different sample sizes in the test of non-uniform phase variables. Each line represents the results for one participant.

must detect a certain number of times a target stimulus before executing the corresponding command. Under these conditions, after 100 flashes of each visual stimulus, the BCI uses the procedure explained in this chapter to identify the class of each option. It is expected that only one option elicits ERP data, whereas the rest of the stimuli should be classified in the non-target class. If the results of this operation indicate that one set of epochs corresponds to the target class, and there is at least one set assigned to the other category, the system trains the BCI with this data and replaces the old model. On the contrary, if no set has been classified in the target group, or there are more than one stimulus assigned to this label, the system waits for other 50 epochs, and tries to build the training set again but using 150 trials for each option. The same procedure is repeated until the BCI identifies training data to fit the classifier. However, if more than 300 epochs have been processed without succeeding, the system discards all the epochs and starts the routine again.

In the last case, the system has to wait for 100 epochs of each possible option because the subject has not calibrated the BCI with the routine described in Chapter 5. Then, each set of epochs is processed and classified according to the proportion of phase variables that are not distributed uniformly around the circle. Similar to the previous case, if the test fails in identifying one set for the target group and another one for the non-target options, the system accumulates other 50 epochs and performs the test again. This procedure is repeated until the BCI is able to train the machine learning model successfully.

The first scenario describes the most typical operating conditions in which the BCI is adapted continuously using the last classified data. The system adjusts the machine learning model according to the newly registered activity so that small changes in amplitude or latency observed in the ERP responses do not alter the BCI performance significantly. Unless the subject cannot generate evoked potentials temporally because of an exceptional event, the continuous adaptation provides an optimal performance all the time. The question that arises here concerns how often the model must be updated

in the online operation.

If one subject selects twenty options in one hour, and each selection requires around five flashes of the target stimulus, no more than 100 epochs of the target class can be obtained in one hour. Then, the total time required to register 300 trials for training the BCI is around three hours. Alternatively, it is possible to reduce this time by choosing some epochs from old data sets. For instance, after one hour of online operation, 200 epochs can be selected randomly from the original training set. In this example, after two hours of operation, the system would have around 200 new epochs of the target class and would only require 100 observations from the old data set. This strategy allows adapting the system after short periods of continuous operation and avoids abrupt changes in the classification rule.

Another important consideration is the duration that one classifier can discriminate P300 responses without a decrease in performance. In the evaluation of the P300-based BCI, a second classification model was estimated after two hours of operation. Then, a new data set was collected and evaluated with the new model and the classifier obtained at the beginning of the experiment. The purpose of this test is to determine if the new classifier outperforms the model estimated two hours before if new observations are evaluated. At the same time, both classifiers were tested with old data collected after the training phase of the first model. The results of this experiment are shown in Table 6.2.

As can be seen from the table, the BCI achieved higher accuracies most of the time with observations recorded after their respective training routines. A two-way repeated measures ANOVA test indicates that there is a significant difference between the accuracies calculated with data collected after the training routines of each classifier and the accuracies measured with data recorded at another moment of time ($F(1,9) = 12.2530$, $p < 0.01$). This fact demonstrates that the adaptation of the classification rule is necessary to reach optimal results in online conditions, which is in agreement with the results presented by Schettini et al. (2014). However, it is also possible to say that the drop in performance is small after two hours (no more than five percentage points in most cases), so that the same classifier can be used for a couple of hours without an important loss in accuracy. A reasonable selection for adapting the classifier would be to retrain the model once or twice every hour.

Another interested offline test performed with the data collected in the previous experiment consisted of determining if the classifier trained at the beginning of the session can label correctly in online conditions the data recorded two hours later. The training data obtained at the end of the experimental session was prepared to simulate an online run for this test. Then, the system classified and collected new observations for each class as it would do it in normal conditions. In the end, the data obtained for each category was compared with the original labels.

For all the participants, 100% of the collected observations were classified correctly. The BCI was configured to have a false positive rate below 1%, so that the multi-trial classification approach worked as expected, allowing the correct labeling of all the new samples. Some observations of the online simulation were not assigned to any class because the system was waiting for more observations to perform a selection. However, in real conditions, this is not a problem because the interface interrupts the flashing sequence when one option is selected.

Finally, in another experimental session, three participants tried to fit the BCI with the autocalibration procedure. In order to avoid counting big numbers, the subjects were instructed to restart

Subject	Class/ Accuracy	Old Data		New Data	
		Old Classifier	New Classifier	Old Classifier	New Classifier
1	Target	90.31	83.79	88.13	90.33
	Non-Target	90.25	88.61	88.40	92.54
	Accuracy	**90.28**	86.20	88.26	**91.43**
2	Target	87.28	84.45	85.71	89.95
	Non-Target	92.62	90.98	92.99	91.86
	Accuracy	**89.95**	87.72	89.35	**89.41**
3	Target	88.65	85.91	80.48	85.56
	Non-Target	93.47	89.85	92.27	86.28
	Accuracy	**91.06**	87.88	**89.28**	87.42
4	Target	79.20	67.92	70.82	78.58
	Non-Target	84.87	82.20	79.75	81.37
	Accuracy	**82.03**	75.06	75.28	**79.97**
5	Target	85.69	77.47	73.08	77.60
	Non-Target	89.34	83.75	86.87	83.60
	Accuracy	**87.51**	80.61	79.97	**80.60**
6	Target	70.44	72.85	61.94	69.92
	Non-Target	78.54	76.74	76.97	76.60
	Accuracy	74.49	**74.80**	69.45	**73.26**
7	Target	89.59	85.71	85.91	86.58
	Non-Target	94.53	93.23	93.22	92.02
	Accuracy	**92.06**	89.47	**89.56**	89.30
8	Target	91.76	89.73	90.83	91.26
	Non-Target	96.31	95.70	94.45	95.11
	Accuracy	**94.03**	92.72	92.64	**93.18**
9	Target	81.28	64.73	78.14	87.76
	Non-Target	78.38	83.27	80.09	89.50
	Accuracy	**79.83**	74.00	79.11	**88.63**
10	Target	88.84	86.31	79.56	76.62
	Non-Target	91.63	91.21	88.45	86.59
	Accuracy	**90.23**	88.76	**84.00**	81.60

Table 6.2. Classification rates obtained with two different models. The first classifier was fit at the beginning of the evaluation of the P300-based BCI, and the second classifier was computed two hours later. Both models were evaluated with data collected after the training routines of each classifier.

the mental sequence after 100 flashes. In this test, the parameter q_r was set to 0.05. The three subjects achieved classification accuracies around 85% after finishing the auto-calibration. One of them required 150 flashes of the target stimulus to end the routine, whereas two of them needed 200 flashes. These results indicate that the proposed method can identify successfully a calibration set without assistance.

6.3 Comparison of Adaptation Methods for P300-based BCIs

As stated in the previous sections, the proposed method for auto-calibrating and adapting the P300-based BCI addressed two problems at the same time:

- The continuous adaptation of the classification rule by incorporating new observations of the target and non-target classes into the model. The new data is acquired as the subject performs

new selections with the interface. In this case, the system requires an initial classifier with an acceptable performance which is modified in the online operation of the interface.

- The complete calibration of the BCI by identifying a new training set from unlabeled data. The system can be trained without the need of a calibration routine and at the same time, it is possible to find a new model when the BCI struggles to generate control commands.

Similar systems that provide a continuous adaptation for P300-based BCIs also incorporate new observations to retrain or adjust the classification model. The main difference between approaches is how the new data recorded in the online run is labeled before the adaptation. In (Panicker, Puthusserypady, & Sun, 2010), the new observations are labeled by two different classifiers which are trained before the online session. Then, the results of each model are used to find the labels of the new observations that are incorporated into the training set of the other classifier. This co-training approach helps to increase the robustness of the adaptation step by giving two different points of view of the same data.

In contrast, the proposed system uses a self-training approach in which the original classifier is used to evaluate the new observations that are used to train the same model. A similar approach was described in (Schettini et al., 2014). However, the proposed system also incorporates another mechanism to identify a training set in the extreme case when the BCI cannot classify new epochs correctly so that the machine learning model can be retrained with a new data set.

One of the most important features of the proposed P300-based BCI is the strict control of false positives. As explained in Chapter 5, even if the accuracy of the single trail classifier of ERP responses is around 70%, the implemented multi-trial classification approach allows detecting with accuracies around 100% target options. For this reason, most of the time the new data is correctly labeled, and in the worst case scenario, the training set contains at least more than 90% of observations identified in the correct class. In this system, the self-training approach is suitable to adapt the BCI because of the high accuracy of the classification model, whereas the method proposed by Panicker et al. (2010) may be useful in spellers and other applications in which errors are allowed to increase the selection speed.

Other strategies implemented to improve the adaptation step involves the use of error-related potentials to identify misclassified data (Martin Spüler, Rosenstiel, & Bogdan, 2012). As explained in Chapter 5, ErrPs have the potential to boost the performance of BCI systems, but this is only possible if the classification accuracies are low and the ability of the system to detect ErrPs is high. In the hybrid system, the BCI operates with accuracies close to 100%, while the classification rates for detecting ErrPs are around 80%. Under these conditions, the ErrP detection does not represent a viable option in the adaptation of the P300-based BCI.

Finally, Zeyl, Yin, Keightley, and Chau (2016) proposed a Bayesian probabilistic model that scores each new trial according to the probability that the BCI is classifying correctly. These scores determine the moment when the interface selects one option of the interface and what trials are used to retrain the classifier. The BCI presented in this thesis also uses a probabilistic model to determine if one option is a target element, but the stopping criterion is defined regarding the expected false positive rate. Both approaches work similarly. However, the proposed BCI also provides the other mechanism for a complete retraining, which increases the autonomy of the system by covering other scenarios in which the model is not working properly.

6.4 Conclusion

This chapter presented a novel method for auto-calibrating P300-based BCIs. This algorithm identifies a new calibration set without assistance by analysing the ERP responses of each option of the interface. The user only needs to start the mental count for one of the flashing icons, and after 150 or 200 trials of each stimulus, the BCI estimates a classification model that can be used in online conditions.

The P300-based BCI also has a self-adaptation mechanism that updates the classification model with new data collected in the online run. In the previous sections, it was demonstrated that the BCI could identify with a high accuracy new observations of each class. These new trials are combined with other observations to create a training set which is used to adapt the classification rule. Since the typical classification rates are close to 100%, the adaptation does not struggle to fit a new model with representative data of each class.

From all the experiments presented in this chapter, it can be concluded that the adaptation and auto-calibration strategies incorporated into the hybrid architecture are powerful tools that allow operating the BCI all the time in optimal conditions and improve the autonomy of the system. The user does not need assistance to calibrate and retrain the system, which is an important feature for subjects that cannot use the interface with other input devices.

Chapter 7

Detection of Steady-State Visual Evoked Potentials

The analysis and discrimination of event-related activity rely on techniques that extract from the brain signals time-locked information to the stimulus onset. In ERP-based BCIs, the EEG is synchronized with the external events (i.e., the stimulus presentations) so that the calculated features can be adequately indexed in the time-domain in all the processing and classification stages. This type of systems only can decode the user's intentions when the application localizes the events in the timeline accurately.

On the contrary, other classes of BCIs are designed to measure the modulation of rhythms or oscillatory components closely related to certain neurophysiological processes. In these systems, the time-domain information is not relevant since the BCI only analyzes local properties of the EEG without considering the external events. The most typical applications of BCIs that are based on the measurement of oscillatory activity are the passive detection of cognitive states and control interfaces that exploit the modulation of sensorimotor rhythms or steady-state visual evoked potentials.

This chapter presents the implementation of the SSVEP-based BCI incorporated into the proposed hybrid system. It also describes the different classification strategies and the training routines used to build the machine learning model of this BCI. Finally, the last section shows several tests to validate the classification model and the online operation of the BCI.

7.1 Feature Extraction Methods for SSVEP Responses

The perception of flashing patterns and flickering stimuli induces oscillatory activity in the EEG which can be observed on the scalp on areas located close to the visual cortex. This brain region participates in the processing of visual information and is found in the occipital area back of the skull. The electrodes that are analyzed in the implemented SSVEP-based BCI are shown in Figure 7.1.

Even thought the response to a single flash is in principle an evoked potential, the continuous presentation of a flashing stimulus produces oscillatory signals whose amplitudes can be distinguished from the background activity and the environmental noise. These oscillations are generated without the need of performing a mental task, but the subject must look at the visual stimulus until the system interprets the desired command.

The spectral properties of SSVEP signals depend on the intensity, frequency rate and visual

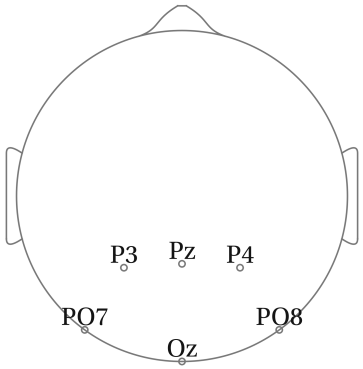


Figure 7.1. Distribution of electrodes of the SSVEP-based BCI.

structure of the presented stimuli (Volosyak, 2011). Typically, the frequencies of the induced oscillations are integer multiples of the flickering rates. For this reason, SSVEP-based BCIs use to obtain features from the components of the EEG that correspond to the flashing frequencies and some of their harmonics. The following additive model approximates the EEG response to a single flashing light of frequency f (Volosyak, 2011):

$$x_e(t) = \sum_{h=1}^{n_h} \left(a_{e,h} \sin\left(2\pi h f t\right) + b_{e,h} \cos\left(2\pi h f t\right) \right) + \eta_{e,f}(t), \tag{7.1}$$

where n_h is the number of harmonics of the model; $a_{e,h}$ and $b_{e,h}$ are multiplicative constants for each channel e and harmonic h; and $\eta_{e,f}(t)$ corresponds to the measured activity that is not related to the SSVEP response, i.e. the environmental noise plus other brain signals.

Let $F = \{f_1, f_2, ..., f_{n_s}\}$ be the set of flickering frequencies of the n_s visual stimuli. The most straight forward approach to obtain features for an epoch $X = [x_e(t)]_{n_e \times n_t}$ of n_e channels and n_t samples consists of computing the power of each frequency band $f_i \in F$ and their corresponding n_h harmonics in each sensor. One option to estimate the spectral content of each channel is the Welch's method (Welch, 1967). This algorithm splits the data in n overlapped segments and computes the *power spectral density* (PSD) function of each sub-window using the *fast Fourier transform* (FFT) and modified periodograms. The power $p_{e,f}$ of each channel e and frequency band f is obtained by averaging the values of the periodograms. Only the frequencies of interests are considered as features ignoring the rest of the calculated values. In the implemented system, the data was windowed using the Hamming function (Thede, 2004b) and divided into eight segments with 50% of overlapping.

Bandpass filtering is another option to calculate the power values of each channel and frequency band. The preprocessed data is bandpass filtered using a bank of real filters centered at the flickering frequencies and their harmonics. Then, the power values are calculated from the filtered signals $Y = [y_{e,f}(t)]_{n_e \times n_f \times n_t}$ as

$$p_{e,f} = \frac{1}{n_t} \sum_{t=1}^{n_t} \left(y_{e,f}(t) \right)^2. \tag{7.2}$$

All the filters in the bank are designed in the frequency domain using a Gaussian kernel with unitary gain and a bandwidth of 4 Hz. At the same time, the flickering frequencies are chosen in such a way that the frequency responses of the filters do not overlap.

The voltage ranges of the EEG may vary on time because of changes in the scalp conductivity and the contact impedance. Besides, other external factors can modify the electrical characteristics of the recorded signals. Unfortunately, these variations alter the power values, so that it is preferred to use normalized values instead. If the Welch's method is used to calculate the power features, the normalized values $\tilde{p}_{e,f}$ are obtained by dividing $p_{e,f}$ by the complete PSD of the respective channel. On the other hand, if the filtering approach is used, the normalization is computed as

$$\tilde{p}_{e,f} = \frac{\sum_{t=1}^{n_t} \left(y_{e,f}(t)\right)^2}{\sum_{t=1}^{n_t} \left(x_e(t)\right)^2}. \tag{7.3}$$

Voltage offsets also perturb the total energy of the signals and the power values at the lowest frequencies. These changes are avoided by rejecting the zero-Hertz component from the raw signals. In the preprocessing stage, the voltage offset is filtered out using an IIR filter with a bandpass between 0.1 Hz and 60 Hz. In this way, the normalized features are not sensitive to small changes in the recorded signals.

The following subsections present other features that improve the discrimination rate of SSVEP responses. All of them are based on the previous ideas, but they incorporate more sophisticated techniques such as spatial filtering and the canonical correlation analysis to increase the separability between classes and reduce the background noise.

7.1.1 Common Spatial Pattern of Bandpass Filtered Data

One of the must known and studied algorithms in BCI research for feature extraction is the *common spatial pattern* (CSP). This method finds spatial filters that project the EEG data in a new space in which the variances of the signals that correspond to one class are maximized whereas that the variances of the signals of a second class are minimized (F. Lotte & Guan, 2011). Even thought the CSP algorithm was introduced initially to improve the classification rate of BCIs that predict movement intentions using the motor imagery paradigm, SSVEP-based BCIs can use this approach to obtain discriminative features because both BCIs are based on control tasks that produce oscillations modulated by the user's intentions.

Let $X^{training} = [X_1, X_2, ..., X_{n_k}]$ be a training set of n_k trials previously validated and free of artefacts; and let $K_i \subseteq \{1,2,...,n_k\}$ be the subset of indices of the trials that belong to the class $i \in \{1,2\}$. The CSP algorithm obtains the projection $w = [w_1, w_2, ..., w_{n_e}]'$ for the *i-th* class that solves the following optimization problem (Dornhege et al., 2006):

$$w^* = \arg\max_{w} \sum_{k \in K_i} \text{var}\left(w' X_k\right) \tag{7.4a}$$

$$s.t. \quad \sum_{k=1}^{n_k} \text{var}\left(w' X_k\right) = 1. \tag{7.4b}$$

If C_i represents the covariance matrix obtained after concatenating the time points of all the trials of the *i-th* class, Problem 7.4 becomes

$$w^* = \arg\max_{w} \left(w' C_i w\right) \tag{7.5a}$$

$$s.t. \quad w'\left(C_1 + C_2\right) w = 1. \tag{7.5b}$$

The optimal projection w^* is obtained by solving the associated generalized eigenvalue problem of the previous expressions. Let Q be an orthogonal matrix of eigenvectors of C_i and let D be a diagonal matrix with the respective eigenvalues of C_i. If $y = Qw$, Problem 7.5 is simplified as

$$y^* = \arg\max_y \left(y' D y\right) \tag{7.6a}$$

$$s.t. \quad y' y = 1, \tag{7.6b}$$

whose solution is given by the vector y with all entries equal to zero except for the *j-th* element that corresponds to the largest eigenvalue of C_i, which is equal to one. It follows that the projection w^* is the eigenvector of C_i paired with the largest eigenvalue. The matrices Q and D are calculated by solving the expression

$$C_i Q = \left(C_1 + C_2\right) QD. \tag{7.7}$$

Typically, more than one projection for both classes is retained in the final solution. In this way, the n_e time-series are transformed into n_p new signals. In the projected space, the calculated features are the log-variances of the new time-series.

Other versions of the CSP algorithm improve the separation rate between the two classes by incorporating spectral information into the optimization problem. In this way, the spatial filters are also designed to highlight the frequency components that best separate the classes. Examples following this approach are the *common spatio-spectral pattern* (CSSP) (Lemm, Blankertz, Curio, & Muller, 2005), the *common sparse spectral-spatial pattern* (CSSSP) (Dornhege et al., 2006) and the *spectrally-weighted common spatial pattern* (SWCSP) (Tomioka et al., 2006).

In the CSSP algorithm, the classical CSP method is used to find the optimal projections for the $2n_e$ channels that are built by delaying the EEG signals τ samples in the time domain, and concatenating in the channel dimension these new time-series with the original EEG. If $w = [w^0, w^\tau]'$ represents a CSSP spatial filter, where $w^0 = [w_1^0, w_2^0, ..., w_{n_e}^0]$ is the vector of coefficients that correspond to the non delayed signals, and $w^\tau = [w_1^\tau, w_2^\tau, ..., w_{n_e}^\tau]$ are the coefficients for the delayed channels, the projected EEG can be expressed as follows (Lemm et al., 2005):

$$x^{cssp}(t) = \sum_{e=1}^{n_e} \left(w_e^0 x_e(t) + w_e^\tau x_e(t-\tau)\right) \tag{7.8}$$

$$= \sum_{e=1}^{n_e} u_e \left(\frac{w_e^0}{u_e} x_e(t) + \frac{w_e^\tau}{u_e} x_e(t-\tau)\right), \tag{7.9}$$

where $u = [u_1, u_2, ..., u_{n_e}]'$ is a pure spatial filter and the coefficients w_e^0/u_e and w_e^τ/u_e work as a FIR filter for the channel e. The spectral properties of the FIR filter depend on the value of τ, which can be found using cross-validation to obtain the parameter that brings the optimal performance. Even though the time-domain filtering has a simple form, it helps to enhance the frequency response of the oscillations that separate the classes.

In contrast, the CSSSP algorithm learns a complete spatiotemporal model without creating artificial signals. It finds the spatial weights and the coefficients of a FIR filter of order n simultaneously by solving two nested optimization problems (Dornhege et al., 2006). One problem is used to find

the spatial filter for a fixed FIR filter, and the other one helps to find the best time-domain filter for the optimal spatial weights that optimizes the separability between classes. In order to enforce a sparse solution for the time-domain filter, the cost function also incorporates a regularization term that penalizes the non-zero terms of the FIR filter.

Similarly, the spectrally-weighted CSP follows the strategy of enhancing the frequency response and the separability rate between classes simultaneously, but the power spectrum of the projected signals is manipulated directly in the Fourier domain. This algorithm calculates the spatial filter and the spectral weights that maximize the separation between the classes. The CSSSP method and the spectrally-weighted CSP may lead to classification models with similar accuracies, the latter being more computationally efficient (Tomioka et al., 2006).

These enhanced versions of the original CSP algorithm are useful when the frequency components of the modulated signals may vary among subjects. For instance, in the motor imagery paradigm, the particular frequency of the sensorimotor rhythm is not the same for all the users, so that the system needs to learn what components are being manipulated in the control task. However, in SSVEP-based BCIs, the frequencies of the modulations are well known in advance, and there are minor differences between subjects. For this reason, a good approach is to use a filter bank of real filters centered at the flickering frequencies and their harmonics, and apply for each band the standard CSP algorithm.

In the training routine, the BCI collects observations of each possible flickering stimulus. Then, for each pair of flashing frequencies, the hybrid system calculates CSP filters for all the extracted bands. If n_m represents the number of all the retained spatial filters for all the flashing frequencies, the trial $Y = [y_{e,f}(t)]_{n_e \times n_f \times n_t}$ is transformed into $Y^{CSP} = [y^{CSP}_{e,f}(t)]_{n_m \times n_f \times n_t}$. Finally, for each frequency band and projected channel, the BCI calculates the log-variance. In the training stage, a variable selection algorithm finds subsets of features with high predictive power that are included in the classification models.

In the present work, the CSP method was modified following the regularized framework described by F. Lotte and Guan (2011). Equation 7.5 becomes

$$w = \arg\max_{w} \left(w' C_i^* w \right) \tag{7.10a}$$

$$s.t. \quad w' \left(C_1^* + C_2^* + \alpha K \right) w = 1, \tag{7.10b}$$

whose associated generalized eigenvalue problem is expressed as follows:

$$C_i^* Q = \left(C_1^* + C_2^* + \alpha K \right) QD. \tag{7.11}$$

Here, C_i^* represents the regularized covariance matrix of the *i-th* class. In chapter 5 was described how to obtain this estimator. The matrix K is used to incorporate a priori knowledge about the form of the expected spatial filter w. Large values of α ensure that the final solution satisfies the a priori information. This strategy is effective when there are available other data sets from which a general form of the spatial filters is estimated. Since our system does not have available other databases for the particular EEG equipment used in this thesis, only the regularization of the covariance matrix was included in the tests presented in this document.

7.1.2 Canonical Correlation Analysis of SSVEP Oscillations

The model expressed in Equation 7.1 provides an easy way to define a set of reference signals that describes the behavior of a typical SSVEP response. If f represents the flickering frequency of a visual stimulus, and n_h is the number of harmonics in the SSVEP model, the multidimensional reference signal $X_f^{ref} = [x_{j,f}^{ref}(t)]_{2n_e \times n_t}$, $j = 1, 2, ..., 2n_e$, is defined as follows (Zhang et al., 2014):

$$X_f^{ref}(t) = \begin{bmatrix} \sin\left(2\pi f t\right) \\ \cos\left(2\pi f t\right) \\ \vdots \\ \sin\left(2\pi \left(n_h f\right) t\right) \\ \cos\left(2\pi \left(n_h f\right) t\right) \end{bmatrix}. \tag{7.12}$$

When the subject observes the stimulus that flashes at the frequency f, it is expected that the canonical correlation between the reference signal X_f^{ref} and one epoch X of EEG data is significantly greater than the obtained when the subject is not gazing at that particular flickering element. Therefore, a natural feature that can be used to detect SSVEP responses is given by the canonical correlation analysis of these signals. In the implemented BCI, for each flashing frequency $f \in F$, the correlation $\rho_f = \text{cancorr}(X', (X_f^{ref})')$ is calculated and concatenated into the feature vector. This method provides an efficient way to reduce the dimensionality of the analyzed data. For instance, if the interface has n_s stimuli, only n_s features are calculated for each epoch.

7.1.3 Likelihood Ratio Test

The *likelihood ratio test* (LRT) represents another possibility to use the reference signal defined in Equation 7.12 to extract features for SSVEP responses (Zhang et al., 2014). Let $Y = [Y_1, Y_2]'$ be a p-dimensional vector of real random variables following a multidimensional normal distribution with mean vector μ and covariance matrix Σ

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix}. \tag{7.13}$$

One method to verify the statistical independence between Y_1 and Y_2 is the likelihood ratio test. For N random samples drawn from Y, the likelihood ratio statistic is defined as

$$\lambda = \left(\frac{|C|}{|C_{1,1}||C_{2,2}|} \right)^{N/2} = v^{N/2}, \tag{7.14}$$

where $|C|$, $|C_{1,1}|$ and $|C_{2,2}|$ are the determinants of the sample covariance matrices of Y, Y_1 and Y_2 respectively.

The quotient v expresses the alienation between Y_1 and Y_2. The greater is v, the more is the certainty of independence between the sets of random variables. From here, a metric measuring the

association between the variables (adjusted according to the dimension of the problem) is given as follows (Zhang et al., 2014):

$$\rho = 1 - \left(\frac{|C|}{|C_{1,1}||C_{2,2}|} \right)^{1/p_2}, \tag{7.15}$$

where p_2 is the number of variables in Y_2, and $\rho \in [0,1]$. Values close to zero of this coefficient are obtained when both sets are independent, whereas values close to one represent the case when both sets of variables are almost identical.

As in the case of canonical correlation analysis, for each reference signal, the coefficient of association ρ_f between X and X_f^{ref} is calculated. In the end, the BCI computes n_s characteristics to detect SSVEP-related activity.

7.1.4 Minimum Energy Combination

As in the case of the CSP algorithm, the *minimum energy combination* (MEC) finds spatial filters to improve the EEG responses of the oscillatory components modulated in one particular control task. However, in this method, the BCI does not calculate a unique set of filters that projects all the processed epochs. Instead, the MEC algorithm obtains projections for each individual trial.

From Equation 7.1, one epoch X can be expressed as follows (Volosyak, 2011):

$$X = GX_f^{ref} + \eta_f, \tag{7.16}$$

where

$$G = \begin{bmatrix} a_{1,1} & b_{1,1} & \cdots & a_{1,n_h} & b_{1,n_h} \\ a_{2,1} & b_{2,1} & \cdots & a_{2,n_h} & b_{2,n_h} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n_e,1} & b_{n_e,1} & \cdots & a_{n_e,n_h} & b_{n_e,n_h} \end{bmatrix}_{n_e \times 2n_h} \tag{7.17}$$

and $\eta_f = [\eta_{e,f}(t)]_{n_e \times n_t}$ is the matrix with the activity that cannot be explained by the SSVEP response. If $\eta_f = \mathbf{0}_{n_e,n_t}$, where $\mathbf{0}_{n_e,n_t}$ is a matrix with all entries equal to zero, the least squares solution for the matrix G is

$$G^* = X\left(X^{ref}\right)'\left(X^{ref}\left(X^{ref}\right)'\right)^{-1}. \tag{7.18}$$

From Equation 7.16, the matrix η_f can be approximated as

$$\eta_f^* = X - G^* X_f^{ref}. \tag{7.19}$$

With the aim of reducing the effect of the background activity and other nuisance signals (artifacts, noise, etc.), the MEC method calculates projections w_i that minimize the energy of η_f^*. In this way, the new signals contain only relevant information for the flickering frequency f. As shown in (Friman, Volosyak, & Graser, 2007), the spatial filters are determined by the eigenvectors of the matrix $M = \eta_f^*(\eta_f^*)'$ paired with the smallest eigenvalues. If v_i is an eigenvector of M and λ_i is the corresponding eigenvalue, the number n_m of spatial filters that are retained is the minimum value that satisfies the

inequality

$$\frac{\sum_{i=1}^{n_w}(\lambda_i)}{\sum_{i=1}^{n_e}(\lambda_i)} \gtrless T, \tag{7.20}$$

where

$$\lambda_1 \leqslant \lambda_2 \leqslant ...\lambda_{n_m} \tag{7.21}$$

and T is a threshold value controlling the energy of the tolerated nuisance activity. The matrix W of spatial weights, therefore, is calculated by

$$W = \left[\frac{v_1}{\sqrt{\lambda_1}} \quad \frac{v_2}{\sqrt{\lambda_2}} \quad \cdots \quad \frac{v_{n_m}}{\sqrt{\lambda_{n_m}}} \right] = \left[w^1 \quad w^2 \quad \cdots \quad w^m \right]. \tag{7.22}$$

Here, the eigenvectors are rescaled so that the projected signals have the same power ranges. Finally, for each flickering frequency, the average power is obtained as

$$p_f = \frac{1}{n_m n_h} \sum_{i=1}^{n_m} \sum_{j=1}^{2n_h} \sum_{t=1}^{n_t} \left(x_{j,f}^{ref}(t) \sum_{e=1}^{n_e} w_e^i x_e(t) \right)^2. \tag{7.23}$$

7.2 Classification Methods and Training Strategies

Essentially, all the previously described features represent the strength of the modulations that theoretically each flickering frequency elicits in the brain activity. These oscillatory components are separated from the background activity and quantified using different techniques that improve the sensitivity of the system to detect SSVEP responses. These characteristics provide reliable information that represents the SSVEP activity so that they can be used to build classification models with high predictive power that are relatively easy to train and adapt.

The hybrid BCI provides two different classification methods to detect SSVEP responses. One strategy consists of finding binary classifiers to discriminate between pairs of stimuli. In this approach, a voting scheme indicates what graphical element represents the command that elicits the stronger SSVEP activity. In contrast, the second method computes one of the described features to measure the particular strength of each SSVEP response. Then, the system compares the obtained values with the estimated null distribution that corresponds to no significant activity. The following subsections details both classification strategies and the corresponding training routines used to find the parameters of the models.

7.2.1 Voting Scheme for Binary Classifiers

All the classification methods presented in Chapter 5 are machine learning models that assign to one observation a label between two possibilities. In the implemented ERP-based BCI, the system only needs to discriminate the EEG responses between target and non-target stimuli. Another class (artifact) is also decoded in the preprocessing stage to reduce the possibility of misclassification because of the presence of contaminated observations. Only when one epoch is free of artifacts, the BCI evaluates the extracted features and finds the corresponding label using the trained classification rule.

In SSVEP-based BCIs there may be more than two graphical stimuli of which the system must find the one that produces the highest modulations on the visual cortex. This *multiclass classification*

problem is addressed by the *one vs. one* reduction method (Bishop, 2006). If n_s represents the number of possible options, this approach consists of evaluating the new instances of the feature vector in the $n_s(n_s-1)/2$ binary classification functions that are built for all the possible pairs of classes. One epoch is classified according to the number of votes that each graphical option receives after evaluating the classifiers.

The system requires a set of observations with SSVEP modulations for each flickering frequency to fit the classification models. This data is collected using a training routine that consists of a sequence of target flashing stimuli that the user must look at. The size, frequency, position and color of each option must be the same in the training stage and the online operation of the BCI because the characteristics of the SSVEP modulations depend on the graphical properties of the stimuli.

In the training routine (see Figure 7.2), the user is instructed to gaze at the specified stimulus, which is selected randomly. This option is highlighted on the user interface (shown in Figure 7.3) for 5 seconds. None of the stimuli flicker at this phase of the training. After highlighting the target stimulus, all the graphical items are turned off for 5 seconds. After the preparation time, all the stimuli flash for 15 seconds while the user observes the previously indicated target. This process is repeated five times for each option, with resting periods of 5 seconds between each training step. If the interface consists of 3 types of stimuli, the total training time is about 7.5 minutes.

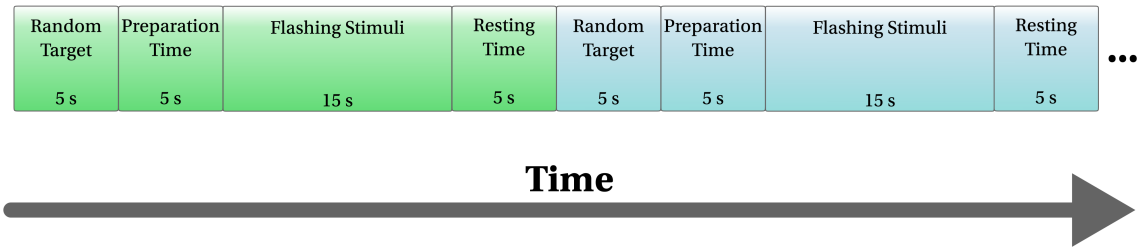


Figure 7.2. Training routine of the SSVEP-based BCI.

After splitting the training set in non-overlapped epochs of EEG data, the BCI validates the trials according to the procedure described in Chapter 4. The epochs free of artifacts are separated according to the flickering frequency f_s, and processed to calculate the feature vectors $x_k^s = [x_{k,1}^s, x_{k,2}^s, ..., x_{k,n_c}^s]'$ of n_c characteristics, where $k = 1,2,...,n_{f_s}$, and n_{f_s} is the number of trials that corresponds to the flashing stimulus of frequency f_s.

In the final steps of the model training, the system uses the epochs of each pair of stimuli to find the parameters of the classification functions $C_{i,j}(x) = l_{i,j}$; where $i < j$; $i, j = 1,2,..,n_s$; and $l_{i,j} \in \{i, j\}$. The implemented BCI allows selecting any of the classification models described previously in Chapter 5. The individual classification rules and the complete machine learning model are evaluated using a second data set or a cross-validation procedure to avoid overfitting.

The voting scheme assigns a class $C(x)$ to a new observation x as follows:

$$C(x) = \begin{cases} \textit{stimulus-s} & \text{if} \quad \sum_{i=1}^{n_s-1} \sum_{j=i+1}^{n_s} I\left(l_{i,j} = s\right) = n_s - 1, \\ \textit{non-SSVEP} & \text{otherwise.} \end{cases} \tag{7.24}$$

Here $I(A)$ denotes the indicator function for a subset A.

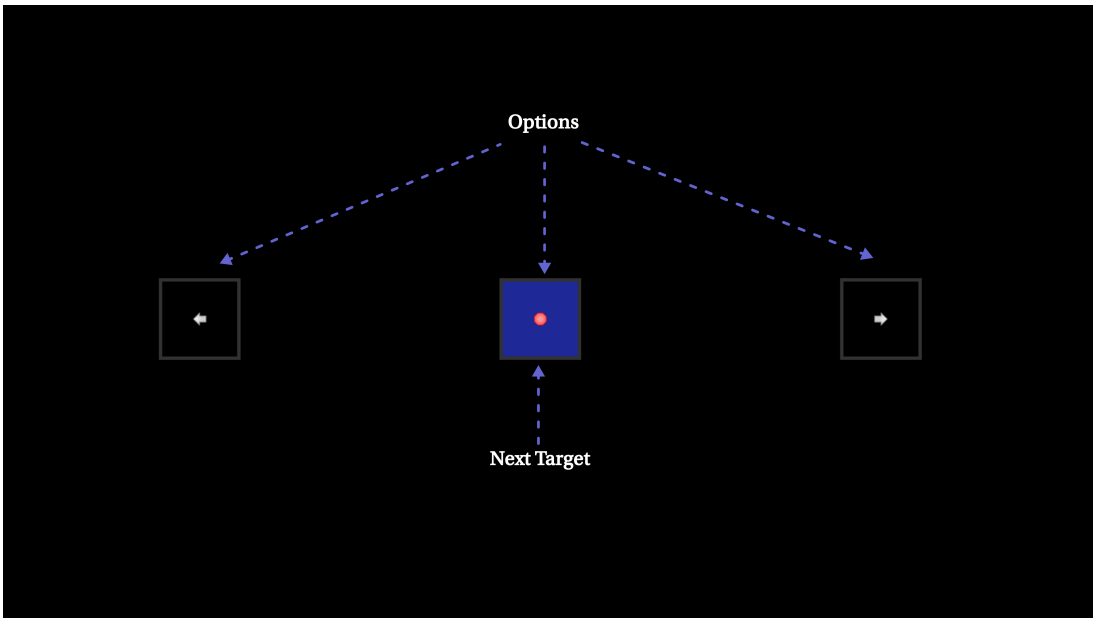


Figure 7.3. Training interface of the SSVEP-based BCI.

Equation 7.24 indicates that the class s is only chosen when this type of stimulus is selected by all the models that discriminate it. Any other case is considered as a non-SSVEP-related response. This strategy helps to define an alternative when there are discrepancies between the evaluations of the individual models. If the evidence of having SSVEP modulations is weak, it is preferred not to select any option of the interface.

The BCI also uses the null class (non-SSVEP activity) to reduce the detection of commands when the user is not performing the control task. In the ordinary operation of the interface, the user has not intentions to operate the system most of the time. Furthermore, it is important not to produce false positives in assistive technology because it may take so long in correcting a wrong decision. The described voting scheme offers the possibility of introducing the null class easily without the need of modifying the training routine.

Another way to control the null class is given by modifying the binary classifiers to include a third label when one observation lies close to the boundaries of the decision rule. In some cases, the classification model has the form

$$C(x) = I\left(f(x;\theta) > T\right), \tag{7.25}$$

where $f(x;\theta)$ is a multivariate function that transform an observation $x \in \mathbb{R}^{n_c}$ into a scalar value; θ is the set of parameters of the classification model; and T is a threshold value that indicates the decision boundary in the univariate projected space. As the function $f(x;\theta)$ moves away from the threshold value, the evidence of having a real SSVEP modulation becomes stronger. This means that it is possible to define a region around T in which is not clear what stimulus corresponds to a feature vector x. Therefore, this range can be assigned to the null class, as it is illustrated in Figure 7.4.

The range of values that correspond to the null class is calculated according to the kernel density estimates of the evaluated observations used to train each classifier. Given a probability value α, the

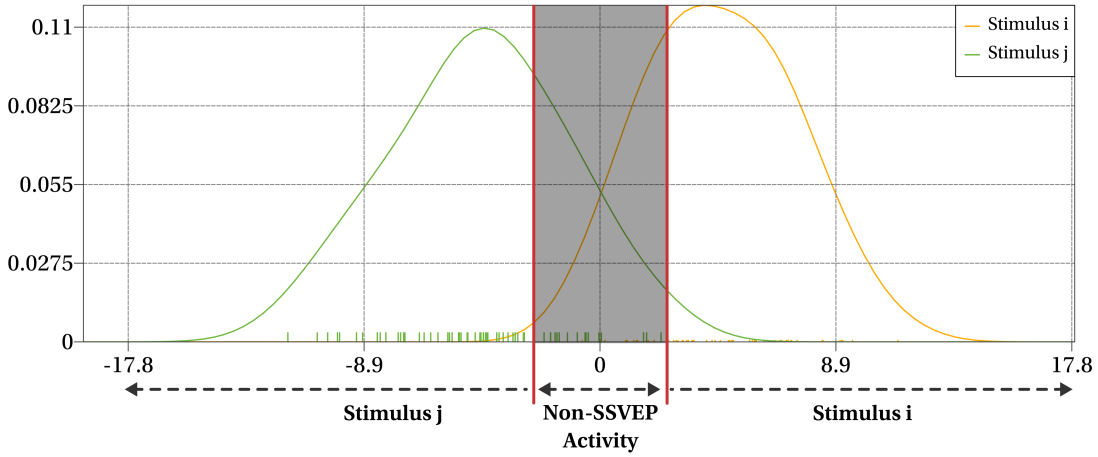


Figure 7.4. Acceptance regions for the null class (non-SSVEP responses) and two flashing stimuli. The boundaries are defined in the univariate projected space that results from evaluating the classification model function. The charts represent the kernel density estimates of the projected observations of both stimulus types.

lower and upper limits are given by

$$T_L = \inf\left\{f(x;\theta) : p\left(f(x;\theta)\,|stimulus\text{-}j\right) \geqslant \alpha\right\}, \tag{7.26a}$$

$$T_U = \sup\left\{f(x;\theta) : p\left(f(x;\theta)\,|stimulus\text{-}i\right) \leqslant 1-\alpha\right\}, \tag{7.26b}$$

where T_L and T_U are the decision boundaries for the null class, and $p(f(x;\theta)|stimulus\text{-}i)$ is the probability value of the function evaluation under the distribution of the *i-th* class.

Even though the described approach may reduce the sensitivity of the model to detect positive responses, as will be shown in the following sections, the SSVEP modulations are robust enough to be classified correctly under this methodology. Similar to the case of the ERP-based BCI, there is a compromise between reducing the false positives and the speed of the interface to detect a new command. The offline tests presented in this document help to find a balance between these two variables for an optimal operation.

7.2.2 Significant Increments of SSVEP-Related Activity

In the classification methodology described previously, the null class was introduced into the machine learning model in two different ways:

- The individual classification rules select the null class when the underlying function evaluation of one observation is close to the decision boundary. The length of the interval that corresponds to the null class depends on the probability density estimates of each pair of categories in the projected space.
- The null class is selected when none type of stimulus achieves the maximum possible number of votes.

Another strategy to control the null class consists of adding a special stimulus type for which the corresponding graphical item does not flash at all. During the training routine, the user has to focus on this graphical element to generate data without SSVEP modulations. The problem with this approach

is to find where to place this item in the interface. Any flashing stimulus close to the new option may induce activity even though the user is centering the visual field on an area without flickering elements. Also, this alternative requires more training time because there is a new option in the interface.

The training routine may also be modified in a different way to collect data for the null class. The straightforward approach would be switching off all the stimuli in some steps of the procedure. This alternative is more suitable to obtain data that represents the null class, but it also requires more training time.

This work presents another alternative to control the null class and generate a machine learning model that detects SSVEP modulations. This method requires minimum training and achieves a similar performance than the obtained with the other methodology based on binary classifiers. It is also highly adaptive and does not require a complicated training routine.

Some of the feature extraction methods described in this chapter reduce the dimensionality of the original data to a few variables. These characteristics represent the individual contribution of each flickering frequency to the observed EEG. This is the case of CCA, MEC, and LRT. In a neutral state (non-SSVEP modulations), it is expected that these features take lower values than the obtained when there are SSVEP modulations on the brain. If the distribution of one of these characteristic in the neutral condition is known, each new observation can be evaluated regarding its position on the null density distribution.

In the training stage, the system only requires data for the null class. This training set is acquired by a simple procedure in which the subject is instructed to observe for a few seconds (between 30 and 60) the center of the screen while all the flashing stimuli are turned off. After the data collection, the BCI splits the EEG into non-overlapped epochs and calculates one of the mentioned features in each validated trial. Let $F^0 = [f^0_{s,k}]_{n_s \times n_k}$ be the matrix of characteristics for the n_s stimulus types and n_k trials, where $s = 1,2,...,n_s$ and $k = 1,2,...,n_k$. If f_s represents a new observation for the stimulus s, the *significance index* I_s of f_s is defined as

$$I_s\left(f_s\right) = \begin{cases} 1 - \frac{p^0_s(f_s)}{1-p^0_s(\mu_s)} & \text{if} \quad f_s \geqslant \mu_s, \\ -1 + \frac{p^0_s(f_s)}{p^0_s(\mu_s)} & \text{if} \quad f_s < \mu_s, \end{cases} \tag{7.27}$$

where $p^0_s(f)$ is the probability value of an observation f under the null distribution of the feature s (estimated using the kernel density method), and μ_s is the sample mean of the feature s in the training set.

The index I_s indicates whether one observation is greater than (or lower than) the sample mean value in the training set for the respective stimulus, which represents the condition of non-SSVEP-related activity. Values close to one indicates that there is a significant increment in the SSVEP activity related to the stimulus s, whereas values close to minus one represent significant decrements. Figure 7.5 shows an example of the localization of a new observation (red line) in the null distribution estimated using the kernel method.

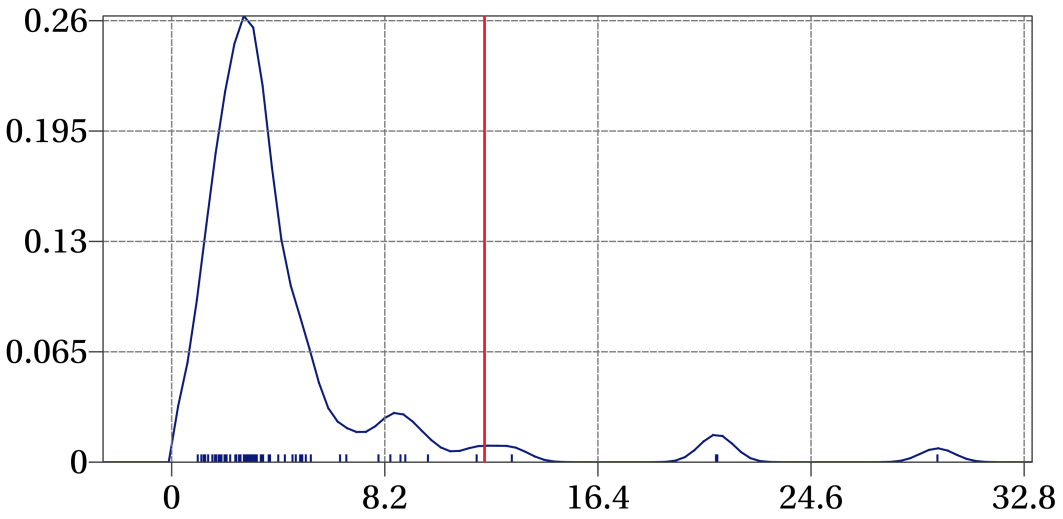


Figure 7.5. Example of kernel density distribution of one SSVEP feature (in this case MEC) for the null condition. The red line indicates the localization of one new observation.

One new epoch is classified according to the significant increments in SSVEP activity. Given threshold values $T = [T_1, T_2, \ldots T_{n_s}]'$, the classification rule is defined as

$$C(x) = \begin{cases} \textit{stimulus-s} & \text{if} \quad I_s \geqslant T_s \quad \& \quad I_i < T_i \quad \forall i \neq s, \\ \textit{non-SSVEP} & \text{otherwise.} \end{cases} \tag{7.28}$$

The threshold values control the sensitivity of the method to detect SSVEP related activity. Values close to one reduce the false positives to the minimum, but they make the interface struggle to generate control commands. On the other hand, values close to zero increases the speed of the interface, but the risks of misclassification is greater. Typically, values between 0.75 and 0.95 offer a good performance.

Some of the SSVEP features are variables with bounded domains. For instance, the canonical correlation coefficient only takes a number between 0 and 1. However, the standard kernel density method is defined for all real numbers, which means that the estimation produces a bias near of the boundaries of the analyzed characteristic. One method that corrects this problem consists of rescaling the standard density estimate according to the local kernel mass that is supported by the variable domain. This approach is known as renormalization (Jones, 1993). Let

$$a(l, u, h) = \int_l^u w(x, h)\, dx. \tag{7.29}$$

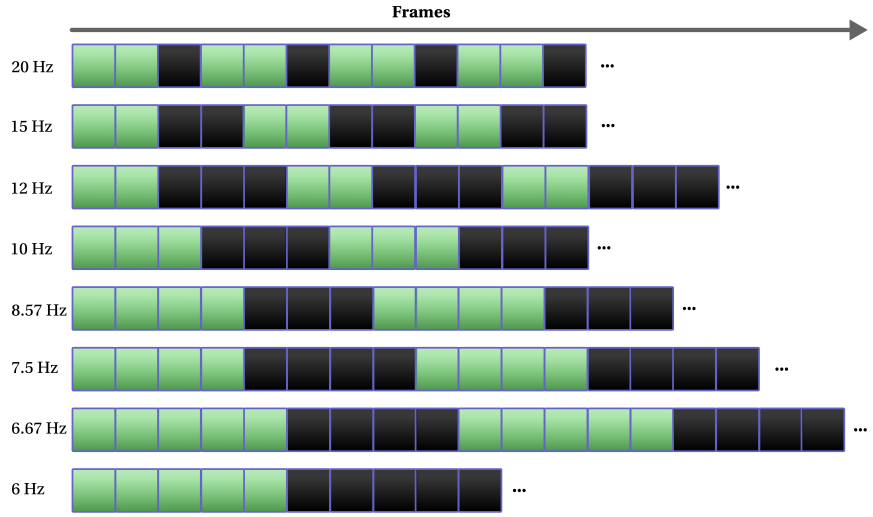
If $x \in [L, U]$, the renormalization of the density estimate $f(x, h)$ for a kernel function $w(x, h)$ is given by

$$f^*(x, h) = \frac{1}{a_0(L - x, U - x, h)} f(x, h). \tag{7.30}$$

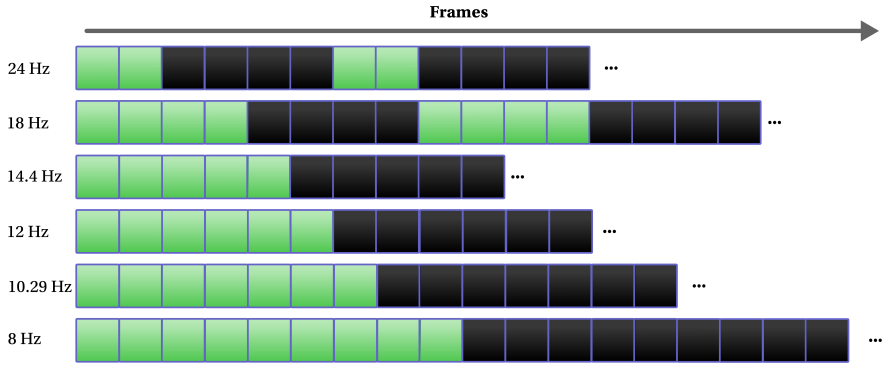
Whenever necessary, the significant increments in SSVEP activity are calculated using the renormalization of the kernel density estimates according to the domain of the processed feature.

7.3 Online Operation of the SSVEP-Based BCI

Flashing stimuli are generated on the screen of a computer as part of the GUI of the BCI. They have on-and-off patterns synchronized with the refresh rate of the monitor which allows producing accurate flashing frequencies. The BCI updates each frame of the screen following the sequences that are shown in Figure 7.6. The patterns illustrated in the figure are for 60 Hz and 144 Hz monitors.



(a) 60 Hz monitor.



(b) 144 Hz monitor.

Figure 7.6. Flashing patterns for different configurations of the SSVEP-BCI. The patters are specified for 60 Hz and 144 Hz monitors. Black boxes represent frames in which the stimulus background is black. Green boxes represent when the stimulus background is white. The flickering patterns are synchronized with the monitor refresh rate to generate stable flashing frequencies and avoid tearing and other annoying effects.

In online conditions, the BCI processes the last epoch of n_t samples periodically and assigns it one label between *artifacts*, *non-SSVEP-related activity*, and one of the flashing options. This operation can be done more than ten times per second, according to the hardware capabilities and the available computational resources. As a result of the periodicity of the classification task, consecutive windows of samples may overlap. In all the tests, the BCI processed EEG data at a rate of 5 times per second. The typical length of the epochs was set to 250 time points (one second of samples), which allows analyzing at least five full periods of frequency components above 5 Hz.

It is expected to obtain errors in classifying single trials. For this reason, the interface only generates control signals when one option has been chosen several times for a few seconds. If the number of times that a flashing option has been selected after n processed epochs is greater than a predefined threshold value n_p, the system executes the respective command and clears all the processed information to restart the classification routine. This operation prepares the BCI to accept new commands. In the opposite case, the system analyses the incoming EEG data without interruptions. If two or more options have been selected more than n_p times, the system considers that there is no SSVEP-related activity. Figure 7.7 shows an example of relative frequencies of three flashing stimuli obtained by the BCI after classifying 20 epochs.

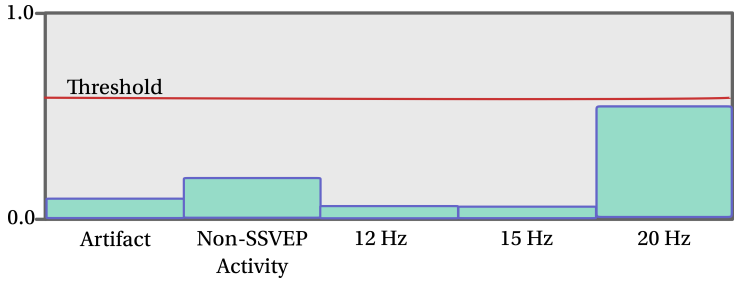


Figure 7.7. Example of relative frequencies for three flashing stimuli after classifying 20 epochs. The artifacts class is also included in the distribution. Only when one stimulus reaches the selected threshold level, the system executes the corresponding option and starts a new sequence for selecting another command.

The system also provides a continuous feedback of the results obtained after classifying n consecutive epochs. The relative frequencies of each class are represented by the size of the stimuli (see Figure 7.8). If one option is producing high SSVEP-related activity, the corresponding flashing icon looks bigger than the rest. On the contrary, if the system has not detected a dominant flickering option, all the stimuli are of the same size. This feedback allows notifying the subject when the BCI is detecting SSVEP-related activity and helps to increase the SSVEP modulations of the desired option.

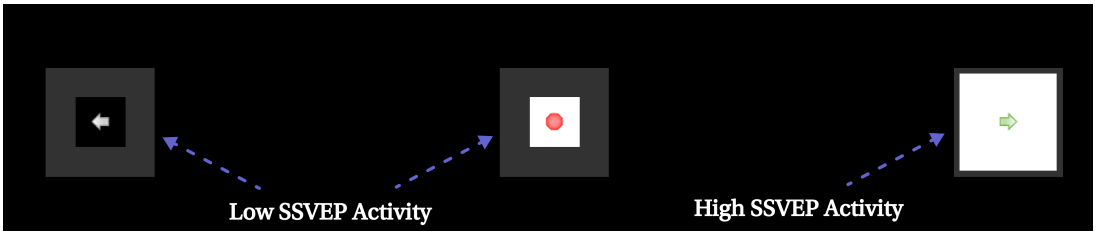


Figure 7.8. Continuous feedback of the SSVEP-based BCI. The stimulus size indicates the strength of each SSVEP modulation.

The strategy of analyzing several epochs to select one option is the same as the presented in Chapter 5 for P300-based BCIs. Likewise, the probabilistic model implemented to estimate the error rate in the P300 classifier can be adapted by using the multinomial distribution to address the multiclass problem. Let $r_{i,j}$ be the probability of classifying an epoch into the j-th class when the subject is looking at the i-th stimulus, where $i, j \in \{0, 1, 2, ..., n_s\}$. Here, the class 0 represents the case when the subject is not gazing at any flashing stimulus. If the user is observing the i-th option, the probability

that the BCI detects this stimulus as the target element is

$$\begin{aligned} r_i &= p\left(X_0 < n_p, X_1 < n_p, ..., X_i \geqslant n_p, ..., X_{n_s} < n_p\right) \\ &= \sum_{\substack{x_i = n_p,...,n, \\ x_j = 0,...,n_p-1, \\ i \neq j}} \frac{n!}{x_0! x_1! ... x_{n_s}!} r_{i,0}^{x_0} r_{i,1}^{x_1}, ..., r_{i,n_s}^{x_{n_s}}, \end{aligned} \tag{7.31}$$

where X_i is the number of times that the stimulus i has been selected in the last n processed epochs, and

$$\sum_{k=0}^{n_s} x_k = n. \tag{7.32}$$

Similarly, the probability of selecting a wrong option is given by

$$\begin{aligned} q_i &= \sum_{k=1,...,n_s, k \neq i} p\left(X_0 < n_p, X_1 < n_p, ..., X_k \geqslant n_p, ..., X_{n_s} < n_p\right) \\ &= \sum_{k=1,...,n_s, k \neq i} \sum_{\substack{x_k = n_p,...,n, \\ x_j = 0,...,n_p-1, \\ j \neq k}} \frac{n!}{x_0! x_1! ... x_{n_s}!} r_{i,0}^{x_0} r_{i,1}^{x_1}, ..., r_{i,n_s}^{x_{n_s}}. \end{aligned} \tag{7.33}$$

A special case is when the subject is not looking at any of the flashing stimulus. The probability of a false positive under this condition is obtained as follows:

$$\begin{aligned} q_0 &= \sum_{k=1}^{n_s} p\left(X_0 < n_p, X_1 < n_p, ..., X_k \geqslant n_p, ..., X_{n_s} < n_p\right) \\ &= \sum_{k=1}^{n_s} \sum_{\substack{x_k = n_p,...,n, \\ x_j = 0,...,n_p-1, \\ j \neq k}} \frac{n!}{x_0! x_1! ... x_{n_s}!} r_{0,0}^{x_0} r_{0,1}^{x_1}, ..., r_{0,n_s}^{x_{n_s}}. \end{aligned} \tag{7.34}$$

Table 7.1 presents some probabilities of success and misclassification for different values of n and n_p. All the values were calculated for three different stimuli. It is assumed that the user is looking at option 1. As can be seen from the table, it is possible to reduce the probability of misclassification and achieve success rates above 0.5 using large values of n, even when $r_{1,1} < 0.4$. On the contrary, if $r_{1,1}$ is above 0.7, the BCI may have high accuracies with low values of n. These results illustrate the relationship between the speed and accuracy of the BCI. If the application demands low misclassification rates, it is better to use at least ten epochs to increase the probability of classifying SSVEP responses correctly.

The parameters n and n_p must be selected considering at the same time all the possible flashing options and the null condition (non-SSVEP activity). If one particular class is more important than the rest, both constants may be tuned to increase the selection speed or decrease the risk of false positives for this option. For instance, in the control interface of the autonomous wheelchair, it is important to avoid the selection of any option when the user is not observing the flashing stimuli.

Table 7.2 presents some examples of false positive rates (FPRs) when the user is not producing SSVEP modulations. Again, large values of n allow decreasing the risk of misclassification regardless the capabilities of the system to identify the null condition in single trials. In the wheelchair controller, at least four seconds ($n = 20$) are analyzed to produce control commands.

BCI Configuration						p_1	q_1
n	**n_p**	**$r_{1,0}$**	**$r_{1,1}$**	**$r_{1,2}$**	**$r_{1,3}$**		
5	3	0.1	0.7	0.1	0.1	0.8369	0.0171
10	6	0.1	0.7	0.1	0.1	0.8497	0.0003
15	9	0.1	0.7	0.1	0.1	0.8689	5×10^{-6}
5	3	0.4	0.4	0.1	0.1	0.3174	0.0171
10	6	0.4	0.4	0.1	0.1	0.1662	0.0003
15	9	0.4	0.4	0.1	0.1	0.0950	5×10^{-6}
15	6	0.4	0.4	0.1	0.1	0.5968	0.0045
5	3	0.6	0.2	0.1	0.1	0.0580	0.0171
10	6	0.6	0.2	0.1	0.1	0.0063	0.0003
10	3	0.6	0.2	0.1	0.1	0.3222	0.1381
20	6	0.6	0.2	0.1	0.1	0.1958	0.0225
30	7	0.6	0.2	0.1	0.1	0.3930	0.0514
35	7	0.6	0.2	0.1	0.1	0.5672	0.1088

Table 7.1. Theoretical probabilities of success and misclassification for different configurations of the SSVEP-based BCI when the subject is looking at the stimulus 1. Three flashing options are considered in the results shown in the table.

BCI Configuration						q_0
n	**n_p**	**$r_{0,0}$**	**$r_{0,1}$**	**$r_{0,2}$**	**$r_{0,3}$**	
5	3	0.7	0.1	0.1	0.1	0.0257
10	6	0.7	0.1	0.1	0.1	0.0004
15	9	0.7	0.1	0.1	0.1	9×10^{-6}
5	3	0.4	0.2	0.2	0.2	0.1738
5	4	0.4	0.2	0.2	0.2	0.0201
5	5	0.4	0.2	0.2	0.2	0.0010
10	6	0.4	0.2	0.2	0.2	0.0191
15	9	0.4	0.2	0.2	0.2	0.0024
20	12	0.4	0.2	0.2	0.2	0.0003

Table 7.2. Theoretical false positive rates for different configurations of the SSVEP-based BCI. Three flashing options are considered in the results shown in the table.

It is important to point out that this model assumes independence among trials, which is not necessary met because of overlapping between time windows. However, Equations 7.31, 7.33 and 7.34 are useful to find an initial combination of parameters without the need of performing an exhaustive search. This model only requires the probabilities $r_{i,j}$, which may be estimated if observations of each class are available. In the offline and online tests of the SSVEP-based BCI, the training data were used to assess all the probabilities of the model. Then, some combinations of n and n_p with high

probabilities of success and low misclassification rates were tested until the user was able to select all the options of the interface without problems.

7.4 Offline and Online Tests of the SSVEP Detector

Ten healthy subjects (M = 7, F = 3) participated in one study that assessed the performance of the SSVEP-based BCI. None of the participants had previous experience with BCI technology. In this experiment, the subjects tested both classification strategies with the training routines described in this chapter. The data sets obtained in this phase of the experiment were used to evaluate the performance of the implemented features for SSVEP systems. Then, after the training routines, the speed and accuracy of the interface were measured in online conditions.

In all the tests, three visual stimuli were included in the interface. The flashing icons were shown on a 23-inch monitor with a refresh rate of 60 Hz. In the pre-processing stage, the raw EEG signals were band-pass filtered between 0.1 Hz and 60 Hz. Other processing tools such as re-referencing and artifact suppression were applied where indicated. In the online tests, the interface processed and classified five times per second new epochs of 250 sample points. Each trial was validated according to the heuristic described in Chapter 4.

At the beginning of each experimental session, the participants were instructed to look at the center of the screen for 60 seconds while the BCI recorded signals for the null condition. None of the options flickered during this task. The training data was used to fit the classifier that is based on significant increments of SSVEP features. Then, the subjects tried to generate SSVEP modulations by looking at the flashing icons. Different frequencies were tested in this phase to identify those that work for each participant. Finally, the participants trained the model based on binary classifiers and tested the interface with both classification models.

Table 7.3 shows the frequencies that were used in the experiment for each participant. Two subjects struggled to control three options at the same time, but they could use two stimuli. The rest of the participants did not have issues in selecting commands with three flashing icons.

Subject	Frequencies (Hz)
1	10, 12, 15
2	10, 12, 15
3	10, 12, 15
4	12, 15, 20
5	12, 15, 20
6	12, 15, 20
7	12, 15, 20
8	10, 12, 15
9	7.5, 8.571
10	6, 15

Table 7.3. Flickering frequencies used in the offline and online tests of the SSVEP-based BCI. Participants 9 and 10 could control only two flashing options.

7.4.1 Offline Tests of the Voting Scheme for Binary Classifiers

In the following tests, the accuracy of the classification model was determine as follows:

$$acc = \sum_{i=1}^{n_s} p_i r_i \times 100\%, \tag{7.35}$$

where n_s is the number of options or stimuli, p_i is the probability that the $i-th$ stimulus occurs, r_i is the probability of classifying correctly observations of the $i-th$ class, and

$$\sum_{i=1}^{n_s} p_i = 1. \tag{7.36}$$

In all the test, the classification rates r_i were estimated using a cross-validation scheme. Furthermore, to avoid a possible bias towards any class, $p_i = p_j = 1/n_s$ for all $i \in \{1,2...,n_s\}$.

The same participants who could not control three options at the same time obtained accuracies below 20% with the model based on binary classifiers. For these subjects, it was not possible to find a configuration that allows them to control the interface with this classification approach. For this reason, they were excluded from the analysis presented in this subsection.

Figure 7.9 shows a comparison between the accuracies obtained with different feature extraction methods. In all cases, LDA and the univariate filter for feature selection (50 features for each binary classifier) were used to find the machine learning model. The null class was not considered because the classification rates were calculated from the cross-validation of the training data, which only contains valid observations of each flashing stimulus. The methods compared in the figure are: power spectrum density (PSD), band-pass filtering, common spatial patterns (CSP), canonical correlation analysis (CCA), likelihood ratio test (LRT) and minimum energy combination (MEC). The number of harmonics for each flickering frequency considered in the feature extraction methods was two.

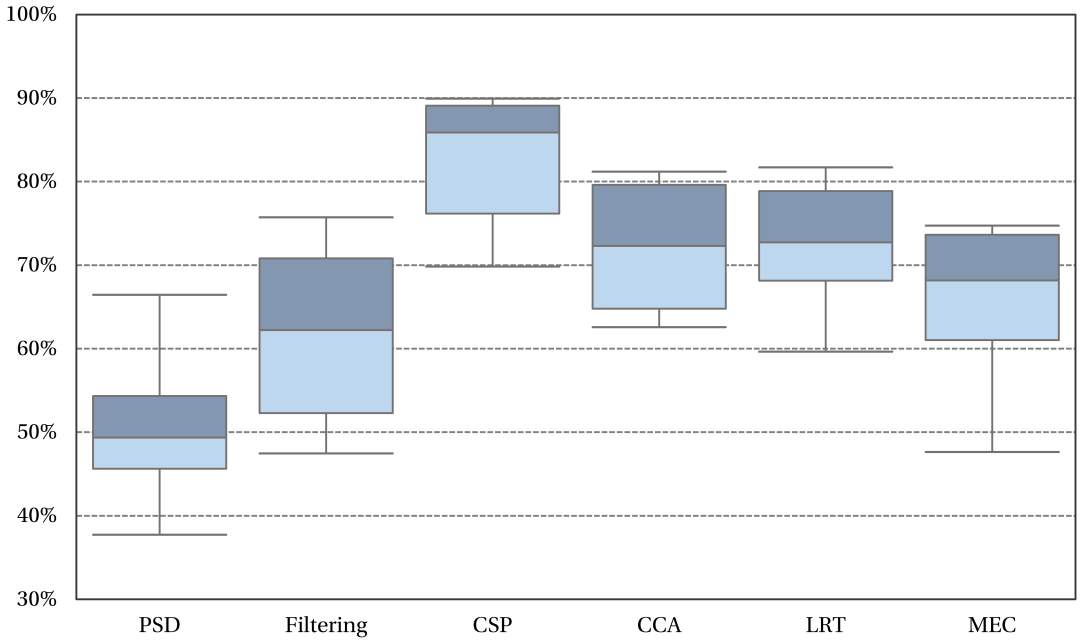


Figure 7.9. Classification rates obtained with different feature extraction methods for SSVEP-based BCIs.

The SSVEP-based BCI achieved the best accuracies with the CSP algorithm (82.26%). Then, LRT (72.62%) and CCA (72%) have similar accuracies, but LRT has lower variability than CCA. Finally, the BCI obtained lower classification rates with MEC (65.85%), bandpass filtering (61.82%) and PSD (50.61%). It is interesting to point out that bandpass filtering was better than PSD for this classification task. It is possible that the non-stationarities of the EEG signals have a negative effect on the power estimates calculated by the PSD method.

The significance levels of the classification accuracies were also calculated using a permutation method. The intention of this test is to investigate if all the accuracies are above the randomness and were not obtained by chance. For three-class classification problems, it is expected that at least 33% of the observations should be classified correctly. However, in the proposed voting scheme, this value may be lower because there is another class implicit in the classification rule (non-SSVEP-related activity). In the permutation test, the labels of the observations are exchanged among classes to create new training sets. Then, for each realization, the classifier is trained and evaluated with the permuted data. Finally, the level of significance is calculated according to the null distribution built with n permutations of the class labels. For all the subjects and feature selection methods, the p-values are below 0.001 for $n = 1000$.

One important parameter of any SSVEP-based BCI is the length of the epochs to process and classify. Long epochs may improve the accuracy of the BCI by reducing the effect of short-time transients that contaminate the EEG. However, the training sets have a limited amount of samples so that it is not possible to increase the time window length without affecting the generalization capacity of the classifier. The training routines need a minimum number of observations of each class to find the parameters of the model. Moreover, the selection speed of the BCI is also controlled by the number of samples that the BCI must collect before the classification task. In the previous comparison, the training data was split into non-overlapped windows of one second, so that around 50 samples represented each flickering frequency after the validation procedure.

Figure 7.10 shows the average accuracies calculated in the cross-validation of the machine learning model with different epoch lengths. The BCI obtained better classification rates with epochs of more than one second independently of the feature extraction method. However, with trials of 2.5 or 3 seconds, the performance drops in some cases because of the decrease in the number of samples in the training data. With bigger training sets, the tendency of increasing the system accuracy with longer epochs would be observed for four or more seconds of samples.

The expected increase in performance for having epochs of two seconds is between 2% and 4%, depending on the feature extraction method. However, the selection time would be one second longer with this configuration. In consequence, the rest of the tests were performed with trials of one second.

The CSP algorithm applied to the bandpass filtered data is by far the best option among the implemented feature extraction techniques. Again, spatial filtering has proved to be a reliable method for BCI applications. For this reason, the following tests were performed with this method.

Figure 7.11 illustrates the effect of adding re-referencing or artifact suppression methods in the pre-processing stage. The chart presents the accuracies obtained by applying to the EEG signals the common average reference (CAR), the surface Laplacian (SL), and the artifact subspace reconstruction (ASR) method. The accuracies calculated without using these techniques are also shown in the plot. As can be seen from the figure, there are no notable differences between the described algorithms. The

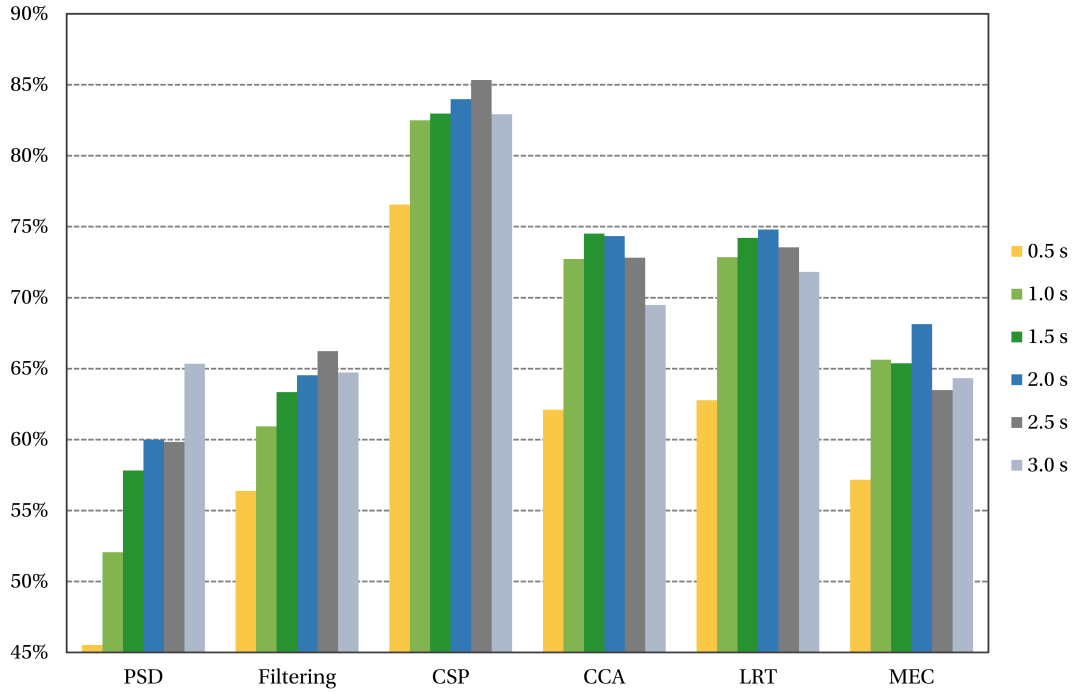


Figure 7.10. Classification accuracies obtained with different epoch lengths.

average accuracies are all around 80%. The EEG signals without re-referencing and artifact suppression obtained the best results (82.26%) followed by the ASR method (81.30%), the surface Laplacian (80.73%), and finally, the common average reference (80.17%).

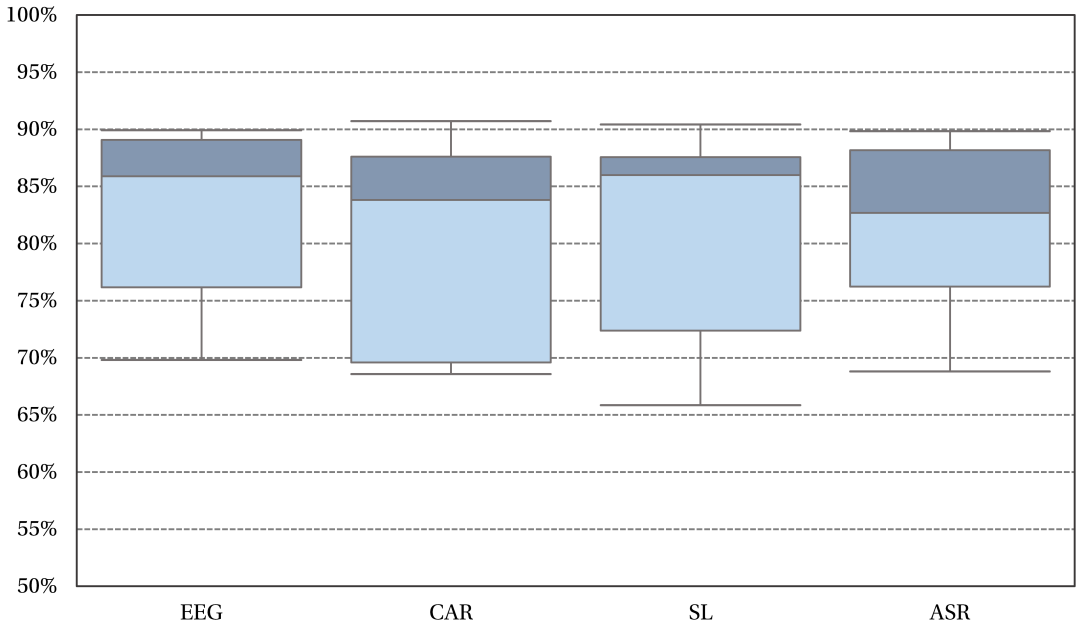


Figure 7.11. Comparison of preprocessing techniques implemented in the SSVEP-based BCI.

The binary classifiers described in Chapter 5 were also evaluated in this study. The accuracies obtained with LDA, QDA, the logistic classifier, SVM and RBF-SVM are presented in Figure 7.12. This

time, RBF-SVM outperformed the rest of the methods (83.21%). This classifier was followed by QDA (82.80%), the logistic classifier (82.31%) and LDA (82.21%). Finally, the lowest accuracies were obtained with the linear SVM (79.62%). All the classification rates are almost the same, however, it is interesting to note that the non-linear methods performed better than the linear functions. The observed differences are irrelevant, so that it is not possible to conclude that non-linear techniques are better in this type of BCI. The average accuracy of the linear models is enough to achieve a good performance in offline and online conditions.

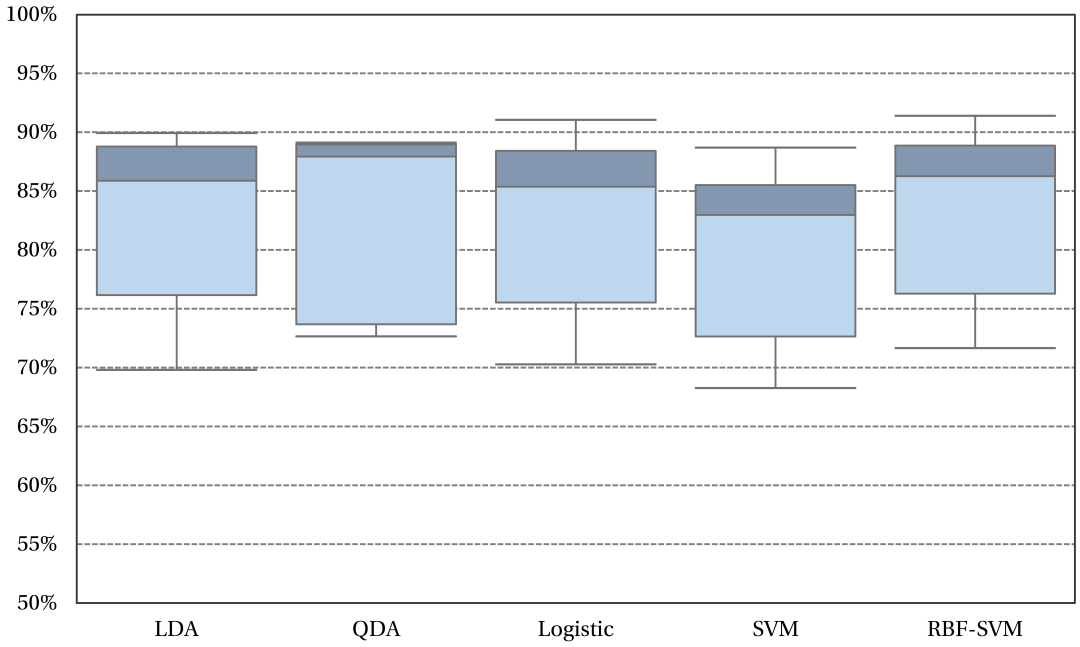


Figure 7.12. Comparison of classification methods included in the SSVEP-based BCI.

The performance of the feature selection methods implemented in the BCI is compared in Figure 7.13. At the same time, Figure 7.14 presents the average number of features selected by each technique. In this test, LDA was used as classification model, while the maximum number of features allowed in the binary classifiers was 50. The BCI obtained similar accuracies with the three univariate filters (81.7%), followed by the stepwise regression (80.65%), the filter-wrapper method (80.24 %) and the wrapped combined with the forward selection (78%).

The method that retrieved the lowest number of features was the filter-wrapper approach (around 7). The stepwise regression and the forward selection also obtained less than ten features on average, which suggests that no more than 10 or 20 features calculated with the CSP algorithm are required to build a classifier with a high accuracy.

All these results indicate that univariate filters combined with LDA, QDA or RBF-SVM are a good choice when the CSP algorithm is used to compute SSVEP features. No more than 20 features are necessary to build the binary classifiers. Furthermore, re-referencing is not needed in this BCI to improve the system performance. Finally, the ASR method may be used when the subject produces many high amplitude artifacts because this method does not worsen the accuracy of the BCI.

The following tests illustrate how the parameter α of Equation 7.26 helps to control the false positive rate when the user is not gazing at any of the flashing stimuli. As stated previously in this chapter, the null class was incorporated into the binary classification rules by limiting the ranges of

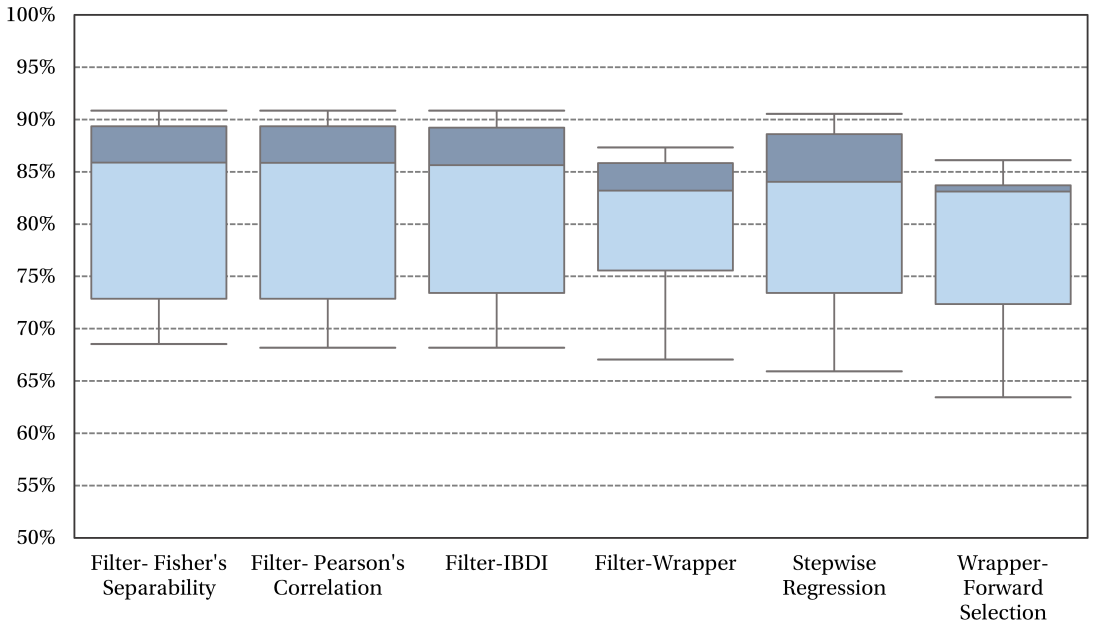


Figure 7.13. Comparison of feature selection methods included in the SSVEP-based BCI.

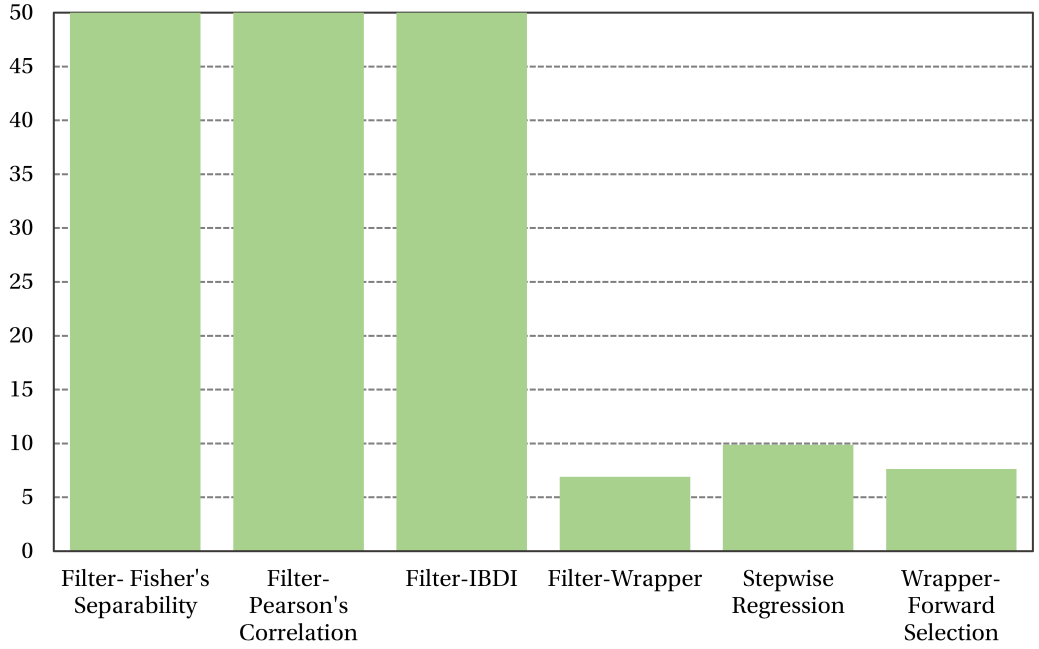


Figure 7.14. Comparison of the number of features obtained by different feature selection methods in the SSVEP-based BCI.

values of the function evaluations that are considered as SSVEP-related activity. Each classifier in the voting rule may have its α value for a fine-tuning. However, in the results presented in this chapter, the three classifiers share the same parameter.

Figure 7.15 presents the percentage of epochs classified in each category at different levels of α when one subject is not looking at any stimulus. In this example, each stimulus type is not detected more than 10% of the epochs if $\alpha > 0.25$. With these classification rates, it is possible to reduce the false positives to less than 0.05% if the system processes at least 10 epochs to generate control signals.

However, it is important to investigate if the BCI can detect each stimulus type correctly with these values of α. In Figure 7.16 it is shown the percentage of epochs classified in each category when the subject is looking at the stimulus of 20 Hz. When $\alpha = 0.25$, the BCI detects more than 60% of the time the stimulus of 20 Hz, whereas the rest of the stimulus are not selected more than 5% of the time. This means that it is possible to find values for the parameters n and n_p such that the system can detect the stimulus of 20 Hz with a high probability and at the same time the risk of misclassification in the null condition may be lower than 0.05%.

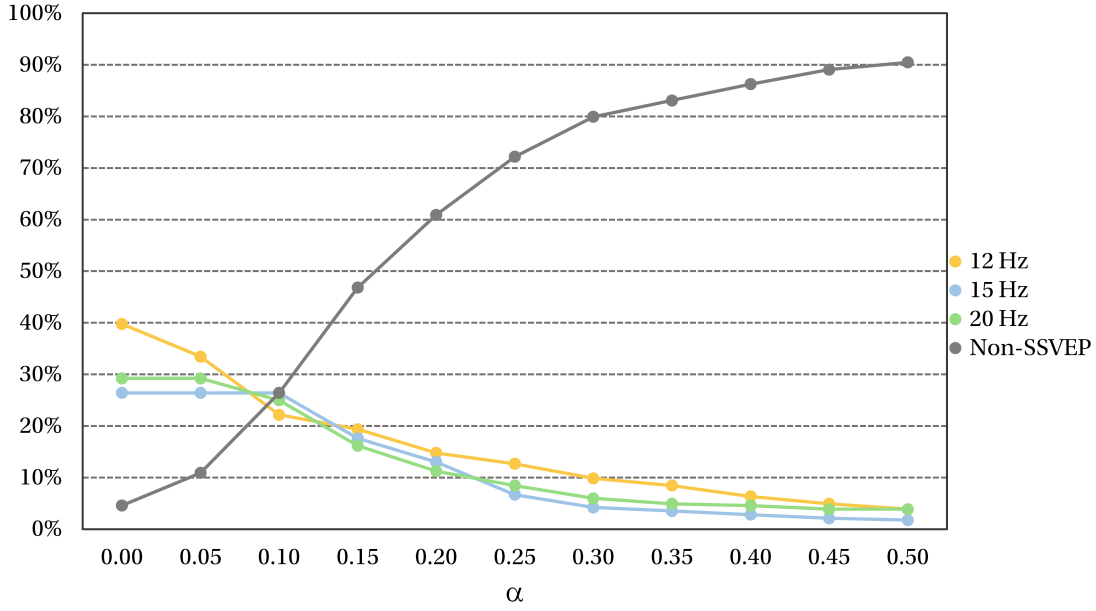


Figure 7.15. Percentage of epochs classified in each category at different levels of α. This parameter controls the area around the boundary decision that is used to define the non-SSVEP class in the binary classifiers. The figure represents the case when one participant is not focused on any of the flashing stimuli.

Other subjects may exhibit a different behavior from that outlined above. Figure 7.17 presents the classification rates at different levels of α in the null condition for another participant. In this example, the BCI has a bias towards the stimulus of 15 Hz, so that it is needed higher values of α to avoid false positives. It is possible that this participant generates strong beta oscillations in such a way that this activity is classified as SSVEP modulations. If this problem cannot be corrected by increasing α, it may be preferred to change the stimulation frequency.

The values of α used in the online tests of the SSVEP-based BCI and the corresponding probabilities $r_{i,i}$ are shown in Table 7.4. Typically, $0.2 \leq \alpha \geq 0.4$.

7.4.2 Offline Tests of the Method Based on Significant Increments of SSVEP-Related Activity

The second classification approach of the implemented SSVEP-based BCI only requires data of the null condition to train the machine learning model. The other parameters that need to be adjusted are the threshold values $T_s \in \{1,2...,N_s\}$ defined in Equation 7.28. Typical values of these constants are between 0.75 and 0.95. If the interface has two or three flashing stimuli, it is easy to find a configuration that works well for each subject. Once the system has data of the null condition, it calculates the

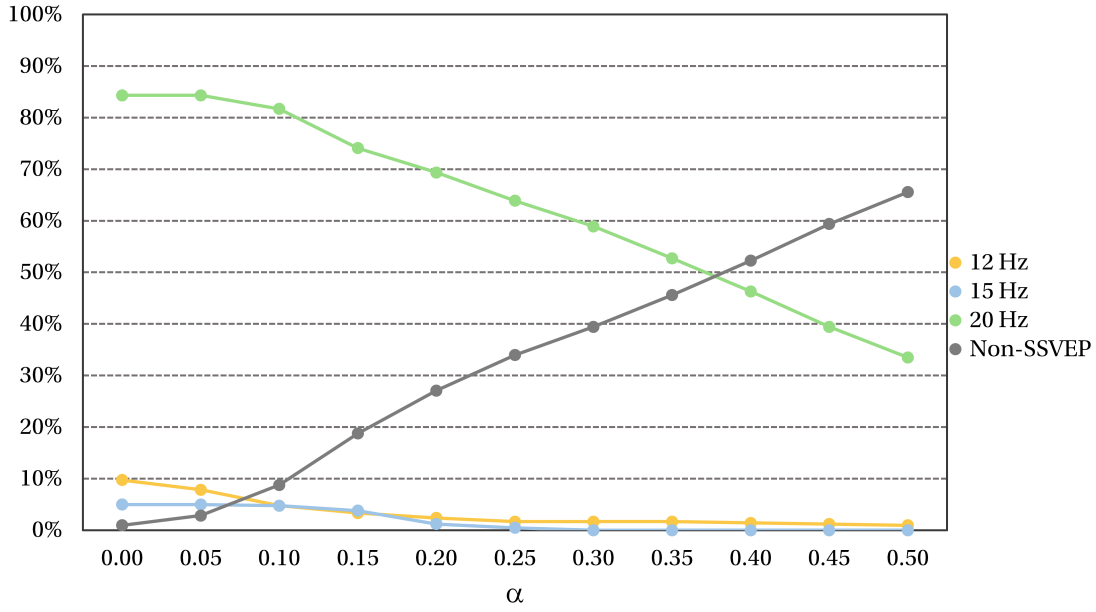


Figure 7.16. Percentage of epochs classified in each category at different levels of α. The figure represents the case when one participant is looking at the stimulus of 20 Hz.

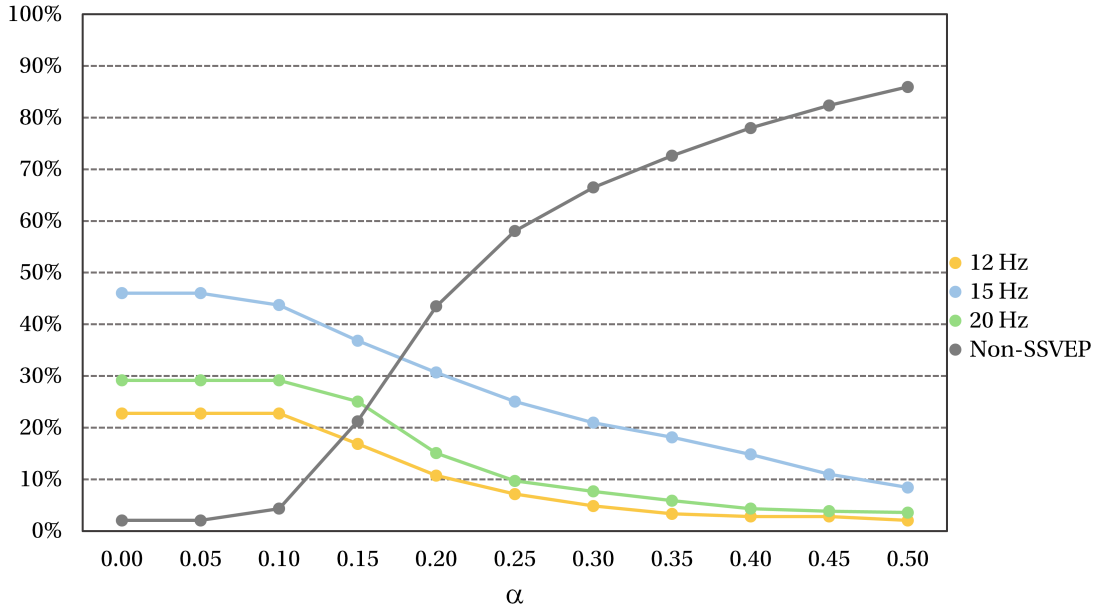


Figure 7.17. Percentage of epochs classified in each category at different levels of α. The figure was obtained with another participant for the null condition.

probabilities $r_{0,i}$ at different thresholds values and stimulation frequencies to find a combination with a low probability of false positives (for example, $r_{0,i} < 0.2 \quad \forall i > 0$). Then, the subject tries to produce SSVEP modulations with the selected configuration to determine if the system identifies correctly the target options. If this is not the case, another configuration is tested. The continuous feedback of the SSVEP-based BCI also helps to verify the speed and accuracy of the BCI visually.

The probabilities $r_{0,i}$ are estimated using cross-validation if only one data set of the null condition is available. Since the number of observations may be small in this case, leave-one-out cross-validation is preferred to estimate these probabilities.

Subject	α	$\mathbf{r_{0,0}}$	$\mathbf{r_{1,1}}$	$\mathbf{r_{2,2}}$	$\mathbf{r_{3,3}}$
1	0.2	0.6062	0.6936	0.6880	0.6684
2	0.2	0.7591	0.6611	0.6673	0.7022
3	0.35	0.7596	0.4911	0.4889	0.5194
4	0.2	0.8051	0.6733	0.6475	0.6489
5	0.2	0.7366	0.6712	0.6759	0.6806
6	0.35	0.8149	0.5224	0.5023	0.5045
7	0.4	0.6773	0.4000	0.4062	0.4576
8	0.3	0.7107	0.5030	0.5447	0.5665

Table 7.4. Values of α and the corresponding probabilities $r_{i,i}$ obtained in the offline tests of the SSVEP-based BCI.

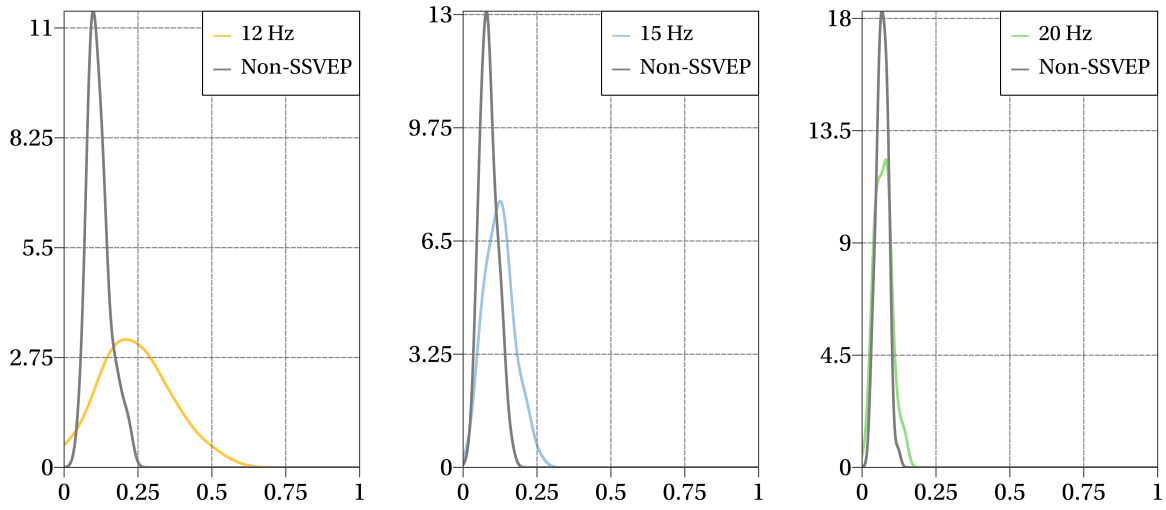
With this classification model, the ten participants could control the interface. However, two of them were able to produce modulations for only two commands. The rest of the subjects did not have issues in selecting commands with the flashing stimuli. Table 7.5 presents the threshold values used in the online tests of the BCI. The probabilities $r_{i,i}$ were estimated using the training data obtained in the other classification strategy.

Subject	$\mathbf{T_1}$	$\mathbf{T_2}$	$\mathbf{T_3}$	$\mathbf{r_{0,0}}$	$\mathbf{r_{1,1}}$	$\mathbf{r_{2,2}}$	$\mathbf{r_{3,3}}$
1	0.5	0.75	0.75	0.6289	0.5231	0.5139	0.5056
2	0.5	0.75	0.75	0.6726	0.5944	0.5305	0.5933
3	0.95	0.5	0.95	0.7494	0.3511	0.3611	0.3333
4	0.6	0.85	0.8	0.6769	0.5511	0.5122	0.4800
5	0.75	0.95	0.75	0.7215	0.5090	0.6241	0.6028
6	0.85	0.75	0.75	0.5450	0.6906	0.5138	0.4013
7	0.9	0.85	0.6	0.6395	0.4190	0.4818	0.3788
8	0.5	0.95	0.95	0.6138	0.3628	0.4855	0.4207
9	0.6	0.25		0.6566	0.3778	0.2528	
10	0.75	0.5		0.6978	0.3833	0.3245	

Table 7.5. Threshold values for each stimulus type and the corresponding probabilities $r_{i,i}$ obtained in the offline tests of the SSVEP-based BCI. Participants 9 and 10 could control only two flashing options.

The data collected to train and evaluate the model based on binary classifiers was used to test the features implemented to quantify significant increments of SSVEP activity. Figure 7.18 shows an example of density estimates of the LRT coefficient obtained when one participant is looking at the stimulus of 12 Hz. Each chart represents the density of the LRT coefficient calculated for each flickering frequency. The null distributions used to test the flashing stimulus are also shown in the plots. In this example, only the LRT coefficient calculated for the stimulus of 12 Hz produces observations outside the test distribution. The rest of the LRT coefficients are close to their respective null distributions so

that their respective significance indices are lower than the obtained with the flickering stimulus of 12 Hz.



(a) 12 Hz. **(b)** 15 Hz. **(c)** 20 Hz.

Figure 7.18. Probability density estimates of the likelihood-ratio test obtained when one participant is looking at the flashing stimulus of 12 Hz. Each chart represents the values calculated for each flashing frequency. The plots also present the distribution of the same features in the null condition.

The performance of each feature extraction method implemented in this classification approach was quantified using the overlapping coefficient defined in Equation 6.20 between the density estimates of each flickering frequency and their respective null distribution. If $H_{i,j}$ represents the probability density function of the feature calculated for the option $i \in \{1,2,...,N_s\}$ when the subject is looking at the stimulus $j \in \{0,1,2,...,N_s\}$, the overlapping coefficient OVL_i indicates the similarity between the distributions $H_{i,i}$ and $H_{i,0}$. It is expected that the overlapping between the density estimate obtained with observations of a certain stimulus type and its respective null distribution is low when the subject is looking at that particular frequency.

Figure 7.19 shows the calculated OVL coefficients for the three methods used to measure significant increments of SSVEP-related activity. The BCI obtained the lowest average with LRT (0.6169), followed by CCA (0.6283). Finally, the system achieved the worst performance level with MEC (0.7171). According to these results, LRT and CCA are good choices to measure increments in SSVEP-related activity.

Finally, Figures 7.20 and 7.21 illustrate how the threshold values T_s control the sensitivity of the BCI to detect SSVEP modulations and the false positive rate simultaneously. In this example, the threshold for the stimulus of 15 Hz must be greater than 0.8 to reduce the probability of detecting this class when the subject is not operating the interface. However, the probability of detecting modulations of this flickering frequency are lower than 0.5 in the same range of T_s. Lower threshold values increase the risk of false positives when the subject is not using the interface but increases the selection speed. It is important to note that each threshold only modifies the curve for the respective stimulus type. For this reason, this approach is relatively easy to calibrate because each stimulus type can be adjusted independently.

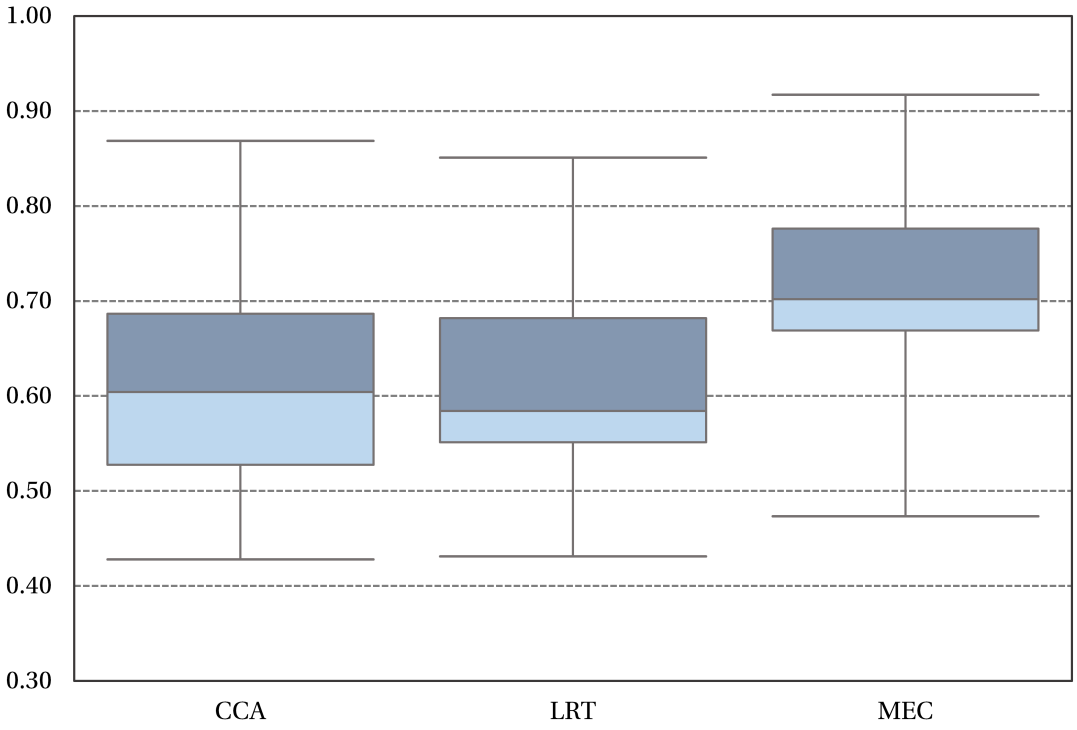


Figure 7.19. Comparison of the overlapping coefficients calculated with different feature extraction methods included in the SSVEP-based BCI. The OVL coefficients were obtained from comparing for all the participants the density estimates of the null condition and the SSVEP responses.

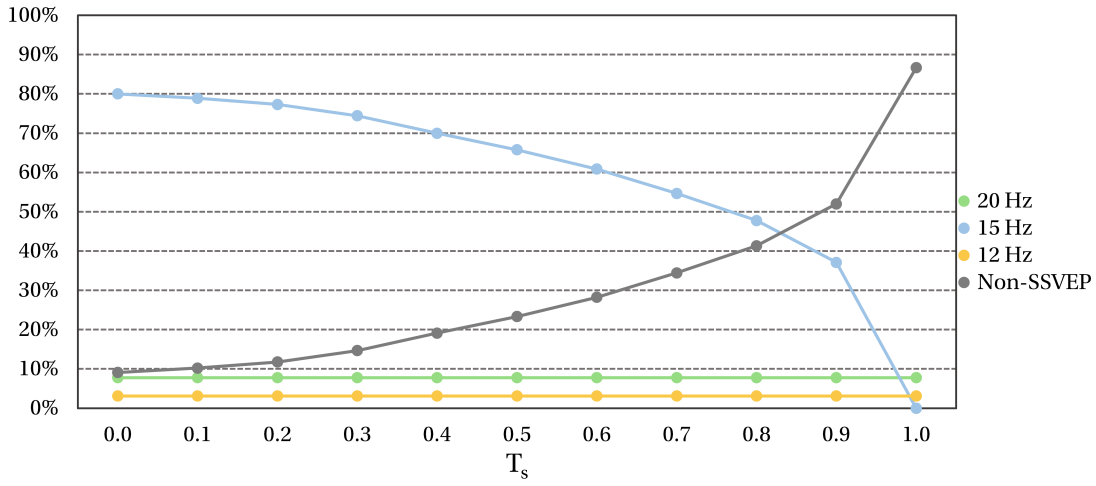


Figure 7.20. Percentage of epochs classified in each category at different threshold values for the stimulus of 15 Hz. This parameter is used to control the sensitivity of the BCI to detect SSVEP responses. The figure represents the case when one participant is looking at the stimulus of 15 Hz.

7.4.3 Online Tests of the SSVEP-Based BCI

In the online tests of the SSVEP-based BCI, each participant was instructed to select a sequence of 40 options indicated by the user interface. Each sequence is unique and is generated randomly before the experiment. Between selections, there is one second of resting time. The model based on binary functions was built with LDA classifiers and the univariate filter for feature selection (20 features). The

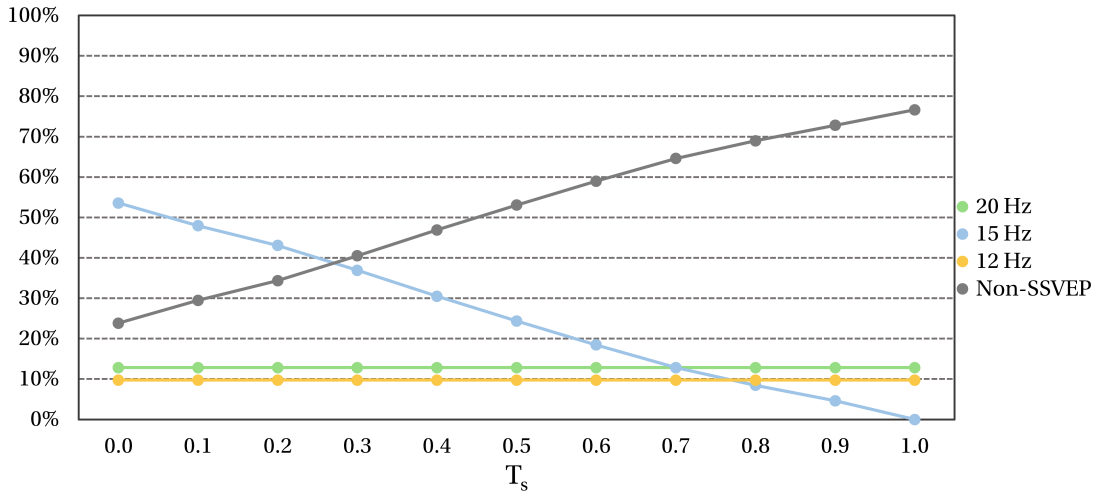


Figure 7.21. Percentage of epochs classified in each category at different threshold values for the stimulus of 15 Hz. The figure represents the case when one participant is not focused on any of the stimuli.

CSP algorithm was used to compute features for this approach, whereas the LRT method was used to construct the model that measures significant increments of SSVEP activity.

In this task, the minimum time required between command selections is about three seconds. One second is needed to acquire a whole epoch with SSVEP modulations, another second to classify five times the EEG data (if $n = 5$), and one second of resting state. Under these conditions, the maximum possible information transfer rate is 31.6992 bit/min. In contrast, if $n = 10$, the maximum ITR is 23.7744 bit/min.

Table 7.6 presents the accuracies obtained with the binary classifiers, while the classification times (without considering resting times) and the number of epochs processed before any selection are specified in Table 7.7. Only two subjects did not obtain 100% of accuracy with the voting scheme. In one case (subject 3), it was necessary to decrease the parameter n_p to reduce the selection times. In consequence, the probability of misclassification increased. Additionally, in the offline analysis, it was found that one of the frequencies has a high probability to be selected in the null condition (around 0.2). Perhaps, it is necessary to increase the parameter α or use another flashing pattern to improve the BCI accuracy. The other subject (number 7) produces weak SSVEP modulations so that it was necessary to use high values of n to avoid false positives and improve the recognition rate. However, one option was selected incorrectly by the interface, which can be adjusted by increasing n_p.

The best average ITRs were close to 20 bit/min, around 60% of the maximum theoretical information transfer rate for this experiment. The corresponding selection times were between four and six seconds, but two participants required on average more than ten seconds to select one option of the interface. Besides, some participants needed more than 15 seconds to perform some of the selections because of fatigue and eye strain. In these cases, the subjects could continue with the tasks after closing the eyes for a few seconds. This action did not produce wrong selections or false positives.

The obtained selection times and transfer rates are more than acceptable for controlling the two options available in the control interface of the wheelchair. According to these results, eight of the participants can select at least three options with accuracies close to 100%. Furthermore, it is possible to improve the false positive control by adjusting n and n_p for those cases that did not obtain 100% of

Subject	Accuracy (%)	B_t	L_t	ITR
1	100.0	1.5850	12.507 ± 5.6172	19.823 ± 8.9030
2	100.0	1.5850	11.715 ± 5.9281	18.567 ± 9.3958
3	85.00	0.8251	7.9159 ± 5.2894	6.5316 ± 4.3644
4	100.0	1.5850	9.1363 ± 5.9629	14.481 ± 9.4510
5	100.0	1.5850	9.2450 ± 6.2826	14.653 ± 9.9576
6	100.0	1.5850	9.2114 ± 5.4556	14.600 ± 8.6469
7	97.50	1.3913	4.4344 ± 2.6601	6.0044 ± 3.7010
8	100.0	1.5850	5.4538 ± 3.1746	8.6441 ± 5.0316
Mean	97.81	1.4658	8.7024 ± 5.0463	12.9130 ± 7.4314

Table 7.6. Classification accuracies and information transfer rates obtained in the online tests of the voting scheme of binary classifiers. The selection speed L_t was calculated as the average time expended to detect SSVEP modulations plus the resting time between selections.

Subject	n	n_p	Selection Time (s)			
			Minimum	Maximum	Mean	SD
1	10	9	2.0220	6.3680	3.7975	1.3038
2	10	9	2.4930	9.5300	4.1218	1.7490
3	10	7	1.9840	19.715	6.5797	5.0882
4	10	9	2.0190	15.556	5.5672	4.1135
5	5	4	2.0780	18.488	5.4900	4.4879
6	5	4	2.8190	13.471	5.5137	3.1539
7	20	14	5.5720	28.945	12.811	7.2712
8	15	9	3.3380	21.624	10.002	5.1078
Mean			2.7906	16.712	6.7354	4.0344

Table 7.7. Selection times in seconds of each participant in the online tests of the voting scheme of binary classifiers for SSVEP responses.

accuracy.

The results of the online tests of the classification method based on significant increments of SSVEP activity are shown in Tables 7.8 and 7.9. The two subjects that could not operate the interface with the binary classifiers (participants 9 and 10) obtained 90% of accuracy with this approach. The rest of the participants obtained classification rates of 100%, except one participant who had one incorrect selection.

The information transfer rates achieved with this classification approach are similar to the presented previously with the other model. Five out of the eight participants that tested both methods reported better ITRs with the binary classifiers so that it is not possible to say that one method is faster than the other one. Selection times were between five and six seconds, but three participants required more than 10 seconds to select one option. Also, there were some outliers because of fatigue and eye

Subject	Accuracy (%)	B_t	L_t	ITR
1	97.50	1.3913	9.1756 ± 3.9989	12.766 ± 5.5637
2	100.0	1.5850	9.5570 ± 5.2132	15.146 ± 8.2627
3	100.0	1.5850	8.8437 ± 5.2409	14.017 ± 8.3067
4	100.0	1.5850	8.4779 ± 5.0958	13.437 ± 8.0767
5	100.0	1.5850	3.9311 ± 2.9485	6.2306 ± 4.6733
6	100.0	1.5850	10.142 ± 6.8994	16.075 ± 10.935
7	100.0	1.5850	3.0491 ± 2.1011	4.8326 ± 3.3302
8	100.0	1.5850	8.2192 ± 3.3259	13.027 ± 5.2715
9	90.00	0.5310	6.2024 ± 3.0541	2.7625 ± 1.6217
10	90.00	0.5310	11.375 ± 7.1227	6.0401 ± 3.7822
Mean	97.75	1.3548	9.8716 ± 4.5001	10.4334 ± 5.9824

Table 7.8. Classification accuracies and information transfer rates obtained in the online tests of the model based on significant increments of SSVEP-related activity.

Subject	n	n_p	Selection Time (s)			
			Minimum	Maximum	Mean	SD
1	10	9	2.9130	8.7200	5.5391	8.7200
2	10	9	3.3630	12.752	5.2781	2.5115
3	10	6	2.0980	12.814	5.7845	3.2898
4	10	8	2.0760	13.164	6.0077	3.3554
5	10	9	2.6950	52.702	14.263	15.267
6	10	9	2.8190	19.910	4.9759	4.1955
7	20	16	3.8480	50.030	18.678	14.539
8	15	9	4.3290	9.2490	6.3000	1.6539
9	5	3	2.9130	21.887	10.533	5.4653
10	5	3	0.9480	12.330	4.2748	2.9453
Mean			2.8002	21.3558	8.1634	6.1943

Table 7.9. Selection times in seconds of each participant in the online tests of the classifier based on significant increments of SSVEP activity.

strain, which is also reported in the table of selection times.

Some examples of transfer rates reported in the literature are 42.7 bit/min (Zhang et al., 2014) and 60.78 bit/min (Lin & Shieh, 2014). So far, probably one of the fastest and most accurate SSVEP-based BCI was presented in Volosyak (2011). They developed a speller controlled by five flickering options, which achieved a peak ITR of 124 bit/min and a mean ITR of 70.41 bit/min. Even though the big discrepancy between these transfer rates and the shown in this document, the difference between the average selection times is less than 5.2 seconds. They reported an average of 3 seconds to perform

one classification correctly, whereas the proposed system requires around 6.7354 seconds if the binary classifiers are used to detect SSVEP responses, and 8.1634 seconds if the other approach is employed in the classification stage. The selection times obtained in this study illustrate the sacrifice in speed as a result of the false positive control for the null condition.

The high transfer rates presented in (Volosyak, 2011) were obtained by implementing three strategies. Firstly, the classification task is synchronized with the user's actions, so that the BCI only processes data with SSVEP modulations. In the online test, the subjects had a limited amount of time to switch to another option between selections, which is taken into consideration in the processing pipeline to avoid data with non-SSVEP-related activity. On the contrary, in the proposed BCI, all the EEG data is processed and classified to verify whether the system produces false positives. As stated above, the consequence of incorporating the null condition in the classification model is that the system requires longer selection times.

In the second strategy described in (Volosyak, 2011), all the selections were made by single-trial classification, but the epoch lengths were adapted dynamically according to the user's skills to produce strong SSVEP modulations. This method gives the chance to select one option in less than one second if the subject elicits strong SSVEP oscillations, but at the same time, it is possible to accumulate more data if the EEG has not been classified as SSVEP-related activity. The accuracies obtained in this BCI are around 96.79%, but it is possible to observe subjects with a lower performance (below 80%). In contrast, the proposed BCI uses a multi-trial classification approach to address the problem of misclassification and false positives. In the online tests, the parameters n and n_p were selected in such a way that both the expected selection times and the false positive control were acceptable. All the subjects achieved accuracies above 90% with the classifier based on significant increments of SSVEP-related activity, and as will be shown in Chapter 9, it is possible to observe 0% of errors in long experimental sessions by increasing n.

Finally, the third strategy described in (Volosyak, 2011) consists of using a real-time feedback to indicate which option is eliciting high SSVEP responses. Similar to the feedback implemented in this work, the stimulus size varies according to the results of the processing stages. However, the BCI proposed by Volosyak represents the relative power of each option, while the proposed system indicates the proportion of epochs classified in each category. Both types of feedback work similarly, and according to the participants, this type visual representations becomes appealing for users and increases the long-term use of the interface.

It is important to say that the difference between the average transfer rates reported here and in other works also has to do with the number of available options in the experiment and the type of application. The tests presented here have the intention to validate the proposed BCI for the hybrid interface, which contains only two options and require a strict control of false positives in long experimental sessions. However, as will be shown in Chapter 9, the implemented SSVEP-based BCI can be used in a different interface that can achieve transfer rates above 60 bit/min.

Another relevant characteristic to compare is the training method. In (Volosyak, 2011) was reported a calibration-less approach in which the parameters of the model were adjusted by examining several data sets. An exhaustive offline analysis determined what configuration may work for most of the users. In this BCI, the MEC algorithm is applied to the EEG signals to estimate the average power elicited by each flickering frequency. Then, the option with the highest power determines the target

element. The same approach is reported in (X. Chen, Chen, Gao, & Gao, 2014), where the option with the highest canonical correlation corresponds to the desired command. A similar strategy can be adapted to the proposed system. For instance, instead of classifying significant indices, the feature values can be quantified and classified directly by using predefined thresholds for each frequency. However, this method does not allow a strict control of false positives for the null condition. In the other BCIs, it is assumed that the subject is always producing SSVEP modulations, and it is not clear what happens when the user does not want to select any option of the interface. This strategy works well for spellers or other applications, but in the wheelchair control, the false positive control is necessary to avoid inadvertent commands that modify the state of the system drastically.

7.5 Conclusion

This chapter presented a detailed description of the SSVEP-based BCI incorporated into the proposed hybrid architecture. The purpose of this subsystem is to control two options of the wheelchair interface which help the user to start the flashing sequence of the P300-based BCI and stop the autonomous device. Several tests compared the feature extraction and classification methods implemented for this BCI with the aim of determining the optimal configuration of the interface.

All the results presented in this chapter prove that the proposed SSVEP-based BCI can achieve high accuracies when the user wants to select one of the flickering options. At the same time, it is possible to have a good control of false positives when the subject is not operating the interface. The multi-trial classification approach provides a simple way to find a balance between the selection speed and the average accuracy of the system.

Two different multi-class classification methods for SSVEP responses were implemented and tested. One approach consists of a voting scheme for binary decisions which discriminate between two flickering options. In this case, the model functions of the binary classifiers were modified around the decision boundary to define the null class (non-SSVEP-related activity). On the other hand, the other approach consists of measuring the strength of the theoretical SSVEP activity elicited by each option to determine what components explain better the observed EEG.

All the participants could operate the interface with the classifier based on significant increments of SSVEP responses, whereas two of them struggled to control the interface with the voting scheme. Nonetheless, in some cases, the voting scheme achieved higher information transfer rates. Thus, if one application requires faster ITRs, both approaches must be tested to find an optimal performance. For the wheelchair control, this is not the case, and it is more important to select the method that can be used by most of the people. The method based on significant increments of SSVEP activity achieved acceptable selection speeds, a high accuracy and an excellent control of false positives. For these reasons, it was selected in the online test of the autonomous wheelchair.

In summary, the advantages of the method based on significant increments of SSVEP-related activity are that it only requires data of the null condition and can be easily calibrated. After training the BCI for only 60 seconds, the system can estimate the probabilities of misclassification in the null condition for each possible frequency and select those that are easy to control. Then, the continuous feedback helps to find the thresholds for each flickering frequency easily with minimum intervention. If observations of all the flickering frequencies are available, the system can estimate all the probabilities

$r_{i,j}$ for different threshold values which allow a fine tuning of the parameters of the model. The proposed BCI is flexible in the sense that it can be adapted to different situations, and even the calibration routine can be modified to fulfill the requirements of the final application.

Chapter 8

Passive Measurement of Cognitive States

The hybrid BCI incorporates a tool that measures in real-time some of the cognitive states that can be observed using non-invasive brain imaging techniques. Relaxation and alertness levels, task engagement, drowsiness, and surprise, among others, are examples of mind states that EEG-based BCIs can quantify. Similarly, the detection of altered states of consciousness such as sleep disorders and drunkenness is another possibility for this kind of technology.

This BCI operates independently to the others without interfering their operation. Additionally, the passive detection of cognitive states can be switched off at any moment. The purpose of this system is to provide clues to the control interface about certain conditions that may improve the interaction between the user and the machine. In assistive technology, task engagement, sleepiness, and boredom are variables of interest, so that the real-time detection of these states may result in a better human-machine interaction.

The working principle of the measurement of mind states is similar to the detection of SSVEP related activity. Because of the oscillatory nature of the EEG, the system decomposes the electroencephalogram in band-limited signals from which features are computed to obtain metrics for representing the strength of the cognitive states. In order to reduce errors, the machine learning model analyzes several consecutive epochs which may overlap in time.

The following sections describe the BCI implemented to measure cognitive states. One application of this system is shown later with the aim of providing more information about this type of systems and how they can be used in different scenarios.

8.1 Quantification of Mind States

The estimation of the spectral content of the EEG is the most common way to characterize the brain activity when there is no information about the events that elicit changes in the electrical signals. Fast fluctuations and ERPs are not easy to interpret when there are not events localized in the time domain that help to measure the latency of the peaks and valleys in the EEG. A positive aspect is that some cognitive states can be described according to the amplitude of the different neural oscillations. For instance, attention shifts can be quantified using the amplitude of alpha waves (Treder, Bahramisharif, Schmidt, van Gerven, & Blankertz, 2011).

A second feature extraction method for cognitive states consists of finding relations between

electrodes. If two or more EEG channels have a high correlation, it is possible to assume that some brain structures are working together to solve the same task. This type of characteristics are usually known as *connectivity features* (Bashivan et al., 2015).

The passive BCI uses a bank of complex sinusoidal quadrature FIR filters to extract the oscillatory components from which power and connectivity features are computed. The bandwidth and tuning frequencies of each filter are set according to the particular cognitive state to quantify. The typical configuration of the bank consists of filters whose frequency responses cover the ranges of the brain oscillations alpha, beta, gamma, theta, and delta.

After filtering the signals, the system splits the data in epochs of length n_t, and calculates the amplitude and phase values at each time point. Let $x_e(t)$ be the potential value at time index $t = 1,2,3,...,n_t$ and electrode position $e = 1,2,3,...,n_e$; and let $z_{e,b}(t)$ be the complex filtered value for band $b = 1,2,3,...,n_b$. Let $s_{e,b}(t)$ and $\phi_{e,b}(t)$ be the magnitude and phase of $z_{e,b}(t)$. The passive BCI extracts the following features:

- *Channel power*:

$$p_e = \frac{1}{n_t}\sum_{t=1}^{n_t}(x_e(t))^2. \tag{8.1}$$

- *Average power*:

$$\bar{p}_E = \frac{1}{n_E}\sum_{e\in E} p_e, \tag{8.2}$$

  where $E \subseteq \{1,2,...,n_e\}$ is a subset of electrode indices, and n_E is the number of elements in E.
- *Channel band power*:

$$p_{e,b} = \frac{2}{n_t}\sum_{t=1}^{n_t}\left(s_{e,b}(t)\right)^2. \tag{8.3}$$

- *Average band power*:

$$\bar{p}_{E,b} = \frac{1}{n_E}\sum_{e\in E} p_{e,b}. \tag{8.4}$$

- *Band power ratio*:

$$r_{e,b_1,b_2} = \frac{p_{e,b_1}}{p_{e,b_2}}, \tag{8.5}$$

  where $b1, b2 \in \{1,2,...,n_b\}$, $b1 \neq b2$.
- *Ratio of average band power values*:

$$\overline{r}_{E,b_1,b_2} = \frac{\overline{p}_{E,b_1}}{\overline{p}_{E,b_2}}. \tag{8.6}$$

- *Band power asymmetry*:

$$q_{E_1,E_2,b} = \frac{\overline{p}_{E_1,b}}{\overline{p}_{E_2,b}}, \tag{8.7}$$

  where $E_1, E_2 \subseteq \{1,2,...,n_e\}$, $E_1 \neq E_2$.
- *Resting state ratio*:

$$RSR_{e,b} = \frac{p_{e,b} - p^0_{e,b}}{p^0_{e,b}}, \tag{8.8}$$

  where $p^0_{e,b}$ is the band power at electrode position e and band b in resting state. This value is calculated by averaging the band power values among several epochs extracted from training

data. In the calibration stage, the subject is instructed to relax and don't perform any cognitive task in order to generate a base line condition that represents the resting state.

- *Resting state ratio of average band power*:

$$\overline{RSR}_{E,b} = \frac{\overline{p}_{E,b} - \overline{p}^0_{E,b}}{\overline{p}^0_{E,b}}, \tag{8.9}$$

where

$$\overline{p}^0_{E,b} = \frac{1}{n_E} \sum_{e \in E} p^0_{e,b}. \tag{8.10}$$

- *Circular variance of phase differences*:

$$v_{e_1,e_2,b} = 1 - \frac{1}{n_t} \left| \sum_{t=1}^{n_t} \left(\cos\left(\phi_{e_1,b}(t) - \phi_{e_2,b}(t)\right) + i \sin\left(\phi_{e_1,b}(t) - \phi_{e_2,b}(t)\right) \right) \right|. \tag{8.11}$$

Sometimes is preferred to use the term $\rho_{e_1,e_2,b} = 1 - v_{e_1,e_2,b}$ because this variable represents the concentration of the phase differences instead of their circular dispersion.

- *Total connectivity*:

$$TC_b = \sum_{e_1=1}^{n_e-1} \sum_{e_2=e_1+1}^{n_e} I\left(\rho_{e_1,e_2,b} \geqslant T\right), \tag{8.12}$$

where $T \in [0,1]$ is a predefined threshold value.

- *Zero-lag coherence between channels*:

$$r_{e_1,e_2,b} = \frac{1}{\sqrt{\tilde{p}_{e_1,b}\tilde{p}_{e_2,b}}} \sum_{t=1}^{n_t} \tilde{z}_{e_1,b}(t) \left(\tilde{z}_{e_2,b}(t)\right)^*, \tag{8.13}$$

where z^* denotes the complex conjugate of z, and

$$\tilde{z}_{e,b}(t) = z_{e,b}(t) - \frac{1}{n_t} \sum_{t=1}^{n_t} z_{e,b}(t), \tag{8.14}$$

$$\tilde{p}_{e,b} = \frac{1}{n_t} \sum_{t=1}^{n_t} z_{e,b}(t) \left(z_{e,b}(t)\right)^*. \tag{8.15}$$

The set of features calculated by the BCI depends on the state to quantify and the classification strategy used to decode the information. Likewise the SSVEP-BCI, the passive BCI has two strategies to classify cognitive states:

- *Binary classifiers* for discriminating different states that have opposite meanings. Some examples are resting state vs. cognitive effort, task engagement vs. boredom, and alertness vs. drowsiness. The training routine for this approach is highly dependent on the conditions that modulate the mind states and may demand the solution of cognitive tasks.
  The preferred features of this approach are those that represent the activity of individual channels (or pairs of channels) instead of averages among electrodes. A feature selection algorithm is used to detect those features that maximize the performance of the classification rule while a cross-validation procedure reduces the risk of overfitting.
  Another class in the binary classifier is also considered to represent the label *non-cognitive*

state. This case is similar to the null class described in Chapter 7, which corresponds to non-SSVEP-related responses. An observation that is close to the boundary of the decision rule is classified as a non-cognitive state, allowing the system to manage this particular case in a different way than when a cognitive state has been detected. The threshold values for each class are adjusted according to the desired sensitivity of the BCI.

- *Significance indices* of features that have a direct connection with certain cognitive states. Similarly to the case of significant increments and decrements of SSVEP activity (see Chapter 7 for more details), each feature can be compared against its null distribution estimated from epochs of a training set. In this approach, the index $I_f(x) \in [-1,1]$ for a new observation x of a feature f is given by

$$I_f(x) = \begin{cases} 1 - \frac{p_f^0(x)}{1-p_f^0(\mu_f)} & \text{if} \quad x \geqslant \mu_f, \\ -1 + \frac{p_f^0(x)}{p_f^0(\mu_f)} & \text{if} \quad x < \mu_f, \end{cases} \tag{8.16}$$

  where $p_f^0(x)$ is the probability value estimate of x under the null distribution of the feature f, and μ_f is the sample mean of the feature f in the training set.

  In this strategy, the computed features are analyzed one by one to obtain individual indices which are combined to describe one or more cognitive states. Characteristics representing the global activity such as averages, power ratios, total connectivity, among others, are adequate to define metrics that are easy to interpret from the neurophysiological point of view.

  This approach works well when there is experimental evidence that the extracted features have a direct connection with the state to quantify. If this is not the case, or the variability among subjects is significant, a method based on classification models is more suitable because the BCI learns the set of features that separate the cognitive states.

  The training routine only consists of collecting data for the null condition. This procedure can be combined with the calibration of the other BCIs. Additionally, the system can incorporate newly recorded data to redefine the null distribution of the features at any moment if required. The simplicity of the training routine and the adaptability of the machine learning model are the most important advantages of this discrimination approach.

The passive BCI extracts the selected features and evaluates the machine learning model in consecutive epochs that may overlap in time. When the binary classification approach is used, the system assigns to each epoch one of the following labels: artifact, non-cognitive state, cognitive state 1 or cognitive state 2. The BCI quantifies the strength of the states according to the proportion that each analyzed condition has been selected by the classification model in the last n_k processed epochs. In the final stage, the control interface decides what action follows when certain states have a low or high proportion.

If the BCI characterizes the brain activity with a significance index (or a linear combination of indices), the system compares this value with a predefined threshold above or below of which each epoch is classified as *significant increment* or *significant decrement*. Similarly to the case of binary classification, the BCI assigns to each epoch one of the following classes: artifact, non-significant increment (or decrement), or significant increment (or decrement). The proportion of each class in the last n_k epochs indicates the strength of the cognitive state that the BCI measures.

8.2 Scoring System for Neurofeedback Applications

Another field of application of real-time systems that measure brain oscillations is neurofeedback. The goal of this technique is to improve the brain functionality (and its internal structure) by representing the electrical neural activity as the signals are acquired. In a typical neurofeedback protocol, the subject plays games that give a reward every time that the brain reaches a particular state. After several sessions, the subject learns how to modulate the brain activity consciously even in adverse conditions. The self-regulation of the brain oscillations may help the user to reduce stress and anxiety, improve concentration, and deal with some neurological disorders.

The BCI provides three types of variables to represent the brain activity:

- *Signal states.* The BCI indicates what channels are free of artifacts and what signals contain real EEG. The signal states help the user to identify when the BCI cannot process EEG data because of noise and connectivity problems. In this way, the subject can avoid producing artifacts or verify the state of the equipment during the neurofeedback session.
- *Instantaneous states.* Typically, significance indices or linear combinations of indices of the last processed epoch. These variables help the user to identify the moments in which the brain parameters have reached the desired state.
- *Cumulative score.* This variable represents the history of the instantaneous states. It helps to measure the overall performance of the user. The cumulative score combines the significance indices and the signal states to indicate the total reward obtained during the neurofeedback session. A high score in a moment of time means that the subject has sustained the target cognitive state for an extended time.

Let $c_e^k \in \{0,1\}$ be the state of electrode e in the last processed epoch k (0 for electrodes with noise or artifacts, 1 for clean signals), and let $x^k \in [-1,1]$ be the variable that represents the instantaneous state in the same epoch. The cumulative score $s^k \in [0,1]$ is obtained as

$$s^k = \min\left(\max\left(s^{k-1} + \Delta_s, 0\right), 1\right), \tag{8.17}$$

where

$$\Delta_s = \Delta_t \left(I\left(C^k \geqslant T_{c_1}\right)\left(\Delta_h I\left(x^k \geqslant T_h\right) + \Delta_l I\left(x^k \leqslant T_l\right)\right) + \Delta_c I\left(C^k \leqslant T_{c_2}\right)\right), \tag{8.18}$$

$$C^k = \frac{1}{n_e}\sum_{e=1}^{n_e} I\left(c^k = 1\right), \tag{8.19}$$

$I(A)$ denotes the indicator function for a subset A; $T_h \in [-1,1]$, $T_l \in [-1,1]$, $T_{c_1} \in [0,1]$, and $T_{c_2} \in [0,1]$ are threshold values; Δ_t is the elapsed time between epochs $k-1$ and k; Δ_h, is the reward (or punish) rate for the instantaneous state x^k when this value is high; Δ_l is the corresponding rate for low values of x^k; and Δ_c represents the rate for the channel states. The reward rates can also be negative numbers, which means that the corresponding terms decrease the cumulative score.

If more than one instantaneous state is analyzed in the neurofeedback paradigm (for instance one variable for alpha activity and another for beta oscillations), the increment in the cumulative score

for n_x instantaneous states is

$$\Delta_s = \Delta_t \left(I\left(C^k \geqslant T_{c_1}\right) \sum_{i=1}^{n_x} \left(\Delta_{h,i} I\left(x_i^k \geqslant T_{h,i}\right) + \Delta_{l,i} I\left(x_i^k \leqslant T_{l,i}\right)\right) + \Delta_c I\left(C^k \leqslant T_{c_2}\right)\right). \tag{8.20}$$

This thesis presents one example of neurofeedback protocol in which the described scoring system was used to provide the necessary information for the user interface. In this example, the three variables are shown in the user interface to provide the needed information to conduct the experiment.

8.3 Real-Time Enhancement of Cognitive Features

Up until this point, this thesis has described three strategies implemented in the hybrid-BCI to reduce the influence of noise and EEG artifacts. In the preprocessing stage, the signals are filtered to remove the 0 Hz component and high-frequency oscillations. These components are not associated with the brain activity that the system is analyzing (ERPs, SSVEP related responses, and cognitive states) so that IIR and FIR filtering are enough to reduce their influence in the processing stages. The second strategy consists of reconstructing the signals with a mix of PCA components obtained from the local window around each time index combined with data in resting state free of artifacts. In the ASR method, the principal components with large eigenvalues are replaced with clean signals estimated from the training set. This algorithm is effective to remove high amplitudes components, but artifacts with low amplitudes are not eliminated in the reconstruction. Finally, each sensor in all the processed epochs is validated individually to indicate the signals with real EEG. This information passes to the classification stage which determines if there is enough information to evaluate the classification rule.

Another strategy included in the BCI to improve the quality of the computed features is based on the Kalman filter (Kalman, 1960). This method was introduced originally to calculate the unknown state of a dynamic system by combining recursively the estimation of the state with measurements that are perturbed with noise. This algorithm has been widely used in control theory and signal processing to improve the accuracy and efficiency of control systems, obtaining important practical results. In contrast to the denoising and artifact rejection methods described above, in this case, the BCI works directly on the calculated features instead of the EEG signals. In the implemented version of the Kalman filter, a *vector autoregressive* (VAR) model of the feature vectors is used to represent the time series in the *state-space* were the recursive algorithm is formulated.

Let $y(t) = [y_1(t), y_2(t), ..., y_{n_f}(t)]'$ be the feature vector calculated at time index t. A VAR process is a model that describes each observation as a weighted linear sum of previous values. The model of order m is given by the expression

$$y(t) = \alpha + \sum_{i=1}^{m} \beta_i y(t-i) + e(t), \tag{8.21}$$

where $\alpha = [\alpha_1, \alpha_2, ..., \alpha_{n_f}]'$ is a vector of constants (intercept of the model), $\beta_i = [\beta_{i,j,k}]_{n_f \times n_f}$ is a matrix of model coefficients (or weights), and $e(t) = [e_1(t), e_2(t), ..., e_{n_f}(t)]'$ is additive Gaussian noise with zero mean and covariance C. In matrix form, Equation 8.21 can be rewritten as

$$y(t) = \beta \hat{y}(t) + e(t), \tag{8.22}$$

where

$$\beta = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_m & \alpha \end{bmatrix}, \tag{8.23a}$$

$$\hat{y}(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \\ \vdots \\ y(t-m) \\ 1 \end{bmatrix}. \tag{8.23b}$$

Given a training set with n_k consecutive observations of the feature vector ($n_k > m$), the next linear system represents the equivalent regression problem of Equation 8.22:

$$Y = \beta \hat{Y} + E, \tag{8.24}$$

where

$$Y = \begin{bmatrix} y(m+1) & y(m+2) & \cdots & y(n_k) \end{bmatrix}, \tag{8.25a}$$

$$\hat{Y} = \begin{bmatrix} \hat{y}(m+1) & \hat{y}(m+2) & \cdots & \hat{y}(n_k) \end{bmatrix}, \tag{8.25b}$$

$$E = \begin{bmatrix} e(m+1) & e(m+2) & \cdots & e(n_k) \end{bmatrix}. \tag{8.25c}$$

An estimate of β is obtained by the least square solution of Equation 8.24:

$$\beta^* = Y\hat{Y}'\left(\hat{Y}\hat{Y}'\right)^{-1}, \tag{8.26}$$

whose corresponding noise covariance matrix is

$$C^* = \frac{1}{N-n_p}\left(Y - \beta^*\hat{Y}\right)\left(Y - \beta^*\hat{Y}\right)', \tag{8.27}$$

where $N = n_k - m$, and $n_p = mn_f + 1$.

To select the order of the VAR model, the BCI uses the stepwise optimization procedure described in (Neumaier & Schneider, 2001). This method evaluates sequentially models of different order ($m_{min} \leqslant m \leqslant m_{max}$) and picks up the optimal configuration that minimizes an information criterion. The implemented metric that is optimized in the BCI is the *Akaike's final prediction error* (FPE)

$$FPE(m) = \frac{1}{n_f}\ln\left(\left|\left(N-n_p\right)C^*\right|\right) - \ln\frac{N\left(N-n_p\right)}{N+n_p}. \tag{8.28}$$

The Kalman filter addresses the problem of estimating the state of a linear process in discrete time that is governed by the equations

$$x(t) = Ax(t-1) + Bu(t) + w(t), \tag{8.29a}$$

$$z(t) = Hx(t) + v(t), \tag{8.29b}$$

where $x(t) = [x_1(t), x_2(t), ..., x_n(t)]'$ is the vector of n state variables at time index t; the vector $z(t) = [z_1(t), z_2(t), ..., z_l(t)]'$ represents the values of the observation obtained at time t; the vector $u(t) = [u_1(t), u_2(t), ..., u_p(t)]'$ contains the p inputs of the system; and the column vectors $w(t)$ and $v(t)$ represent Gaussian noise with zero mean and covariance matrices Q and R respectively. The matrices $A = [a_{i,j}]_{n\times n}$ and $B = [b_{i,j}]_{n\times p}$ relates the previous state and the current input vector with the new state, while the matrix $H = [h_{i,j}]_{l\times n}$ relates the current state with the observation vector.

The filter algorithm consists of a two-step recursive relation in which a new state and the corresponding error covariance matrix P are predicted for the next time point. Then, these estimates are corrected using the new vector of measurements with new information about the system. If $\hat{x}(t-1)$ denotes the corrected state vector at time index $t-1$, and $\hat{P}(t-1)$ is the corresponding error covariance, the *time update* step of the Kalman filter algorithm projects forward the state vector using the equations

$$x^-(t) = Ax(t-1) + Bu(t), \tag{8.30a}$$

$$P^-(t) = AP(t-1)A' + Q. \tag{8.30b}$$

Next, in the *measurement update*, the Kalman filter corrects the prediction by incorporating the observation $z(t)$ as follows:

$$K(t) = P^-(t)H'\left(HP^-(t)H' + R\right)^{-1}, \tag{8.31a}$$

$$x(t) = x^-(t) + K(t)\left(z(t) - Hx^-(t)\right), \tag{8.31b}$$

$$P(t) = (I - K(t)H)P^-(t). \tag{8.31c}$$

The Kalman gain $K(t)$ in the measurement update helps to keep a balance between the estimate obtained using the model equations and the information contained in the measurement vector, which in principle is contaminated by noise. In the extreme case when $R = \mathbf{0}$, the Kalman gain is equivalent to H^{-1}, and $x(t) = H^{-1}z(t)$. On the other hand, if $P^-(t) = \mathbf{0}$, then $K(t) = \mathbf{0}$ and $x(t) = x^-(t)$. If the measurement noise is negligible, the new state can be updated using the measurement vector without correction, while if the model noise is small, the time update provides a correct estimate of the state vector.

To improve the estimates of the cognitive feature vector, the VAR model defined in Equation 8.22 is represented in the state-space as

$$x(t) = \begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-m+1) \\ 1 \end{bmatrix} = Ax(t-1) + w(t), \quad u(t) = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}', \quad z(t) = y(t) + v(t), \tag{8.32a}$$

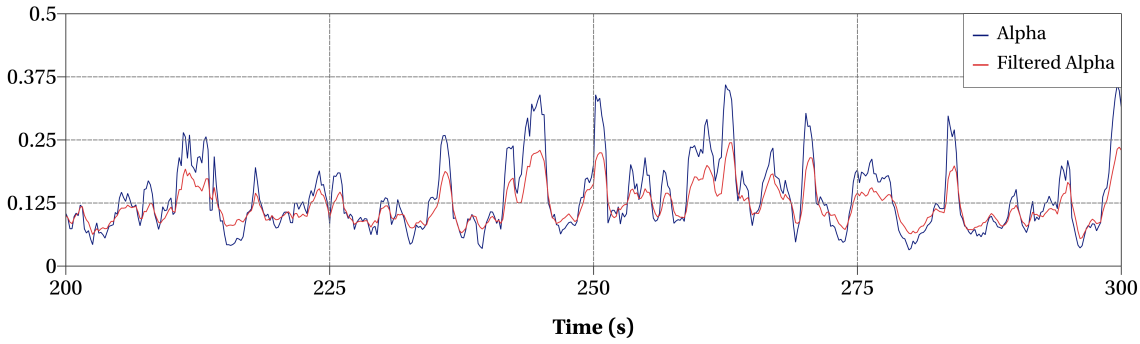
$$A = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_{m-1} & \beta_m & \alpha \\ I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}, \tag{8.32b}$$

$$B = \mathbf{0}, \quad H = \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \tag{8.32c}$$

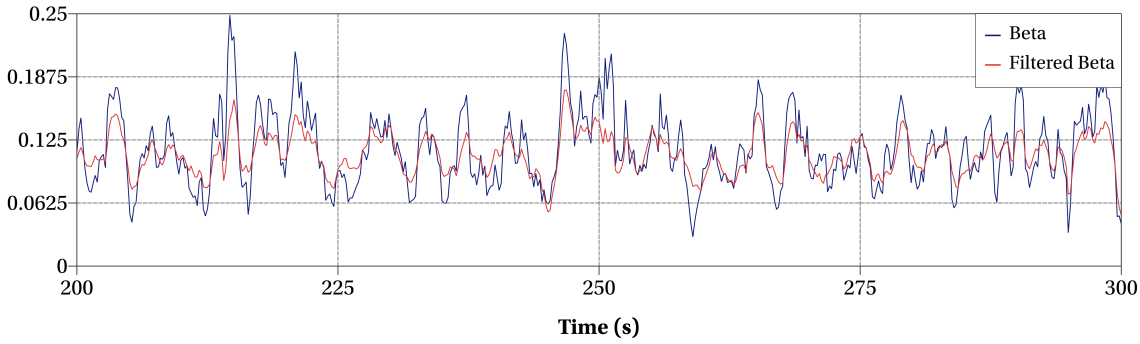
The feature vectors of the first processed epochs are used in the initial guess $x(0)$ that is required in the online operation of the Kalman filter. Moreover, $P(0)$ is calculated as the average noise covariance matrix of an offline run of the filter using the training set that also helps to estimate the parameters of the VAR model. This offline run is repeated several times to ensure convergence in the estimation. In the filter calibration, $P(0)$ is initialized using a diagonal matrix with small values in the main diagonal (0.1 for instance). Finally, the covariance matrices Q and R are estimated using the same training set and the autocovariance least-square estimation method described in (Åkesson, Jørgensen, Poulsen, & Jørgensen, 2007).

The Kalman filter reduces the effect of short-time transients produced by cognitive noise or artifacts not detected in the validation process. However, this operation may also reduce the amplitude of the calculated features. If the same filter is not applied to the training set, the Kalman filter may adversely affect the sensitivity of the system. Figure 8.1 shows an example of filtered normalized power values for three bands: theta, alpha, and beta. In these charts, the Kalman filter reduced the amplitudes for the theta band notably, so that the range of the normalized power values is different for the filtered features. In this example, the training set requires the same correction to adjust the scale of the features.

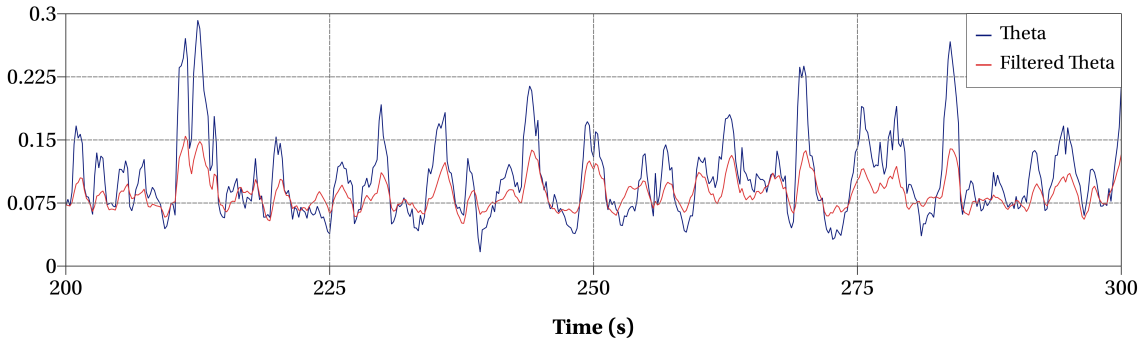
Most of the time, peaks and valleys in the filtered time-series have the same positions of the local extreme values of the non-filtered features. However, it is possible to observe differences between the filter correction and the original time-series. In the previous example, after 250 seconds, there is a small valley in the non-filtered theta activity that is not observed in the corrected data. As stated above, the Kalman filter estimates are calculated by considering the noise level of the features and the computed vector autoregressive model. The multivariate information helps to adjust the observed features, but the new values are only valid if the VAR model has real predictive power. In long online runs, the brain activity may change in such a way that the model may lose its validity to explain the EEG features. In this case, the VAR model and the Kalman filter needs to be updated with new data. For this reason, when the BCI analysis new information to adapt the machine learning model, the Kalman filter is also modified.



(a) Alpha activity.



(b) Beta activity.



(c) Theta activity.

Figure 8.1. Normalized power of tree different bands corrected by the Kalman filter. The multivariate model was defined for theta, alpha and theta.

8.4 My Virtual Dream: A Novel Neurofeedback Protocol for Collaborative Experiments

This section presents one application of the cognitive state detector described in this chapter. This example is a massive neurofeedback experiment in which the BCI provides the cumulative score and the instantaneous state of each participant to manipulate the multimedia elements of an artistic presentation called *My Virtual Dream* (MVD).

MVD is a public multimedia experiment in which simultaneous participants control with their mind the music and visual elements that are reproduced during the event. The relaxation and concentration levels of a group of subjects determine the music layers, the type of animations and the graphical elements that are shown to an audience. One of the ideas behind MVD is to create a unique representation of the collaborative mind states to reflect the tendency of the participants to be relaxed

or concentrated while they enjoy a presentation full of artistic components selected dynamically. An EEG-based brain-computer interface quantifies the alpha, beta and theta rhythms of each participant and provides the information that is needed in the audio-visual neurofeedback paradigm.

The authors of the first version of MVD found evidence that even in an open environment, it is possible to design a neurofeedback paradigm in which subjects learn in a short period how to modulate the beta rhythm (Kovacevic, Ritter, Tays, Moreno, & McIntosh, 2015). This preliminary study also introduced this new experimental setup for collecting EEG data outside the laboratory in collaborative situations. Additionally, by combining science, games, and art in a massive event, it is plausible to increase the public knowledge about the possibilities of brain-controlled technology and neurofeedback. The public awareness of BCI research is crucial for introducing this type of applications in daily life.

In the second version of MVD, it was investigated the perception of the participants regarding their ability to control the interface with some level of dexterity. The majority of the participants (73%) reported good control even though no one had previous experience with BCI applications (Ghani, Mendoza-Montoya, Mclntosh, & Ritter, 2016). The brain-computer interface that translates the subjects' mind states in this new implementation of MVD is a modified version of the system described in this document, which demonstrates the flexibility of the BCI tools developed in this research project to be adapted to different kind of applications.

The main MVD event took place inside a geodesic dome with a 360-degree view (see Figure 8.2). Animations and graphical effects were projected around the dome, while a computer reproduced the musical pieces and sound effects that accompany the presentation. In the final part of the experiment, a professional musician and a singer played some compositions according to their personal perception of the collaborative mind state of the participants, which is represented in the projected visual effects. In total, the experiment consisted of 16 sessions of 20 minutes. The presentation was also available for a public that did not interact with the BCI.

In each session of the MVD experiment, four volunteers interact simultaneously with the BCI. Their mind states are represented by avatars with unique colors and shapes (Figure 8.3). The avatars symbolize the three variables of the scoring system:

- The instantaneous state, represented by the size and shininess of the avatars. A high relaxation level (i.e. low concentration state) corresponds to a big shape with glowing particles turning around the figure. On the contrary, high concentration levels result in little avatars without glowing particles.
- The cumulative score, which is represented by the avatar's position. When an avatar is close to the top side of the projection, the corresponding subject has achieved a high relaxation level for an extended time. On the contrary, the bottom side of the image represents consistent high concentration levels.
- The average electrode state, which is represented by the color of the avatars. When an avatar is painted with a colored tone, the BCI is not detecting artifacts, and the instantaneous states are valid. On the contrary, when an avatar is gray, the corresponding subject is producing artifacts in all the electrode positions, and the BCI cannot compute valid cognitive features. This information encourages the users to generate signals with acceptable quality free of artifacts during the experiment.

Figure 8.2. My Virtual Dream. The participants control with their relaxation levels the multimedia elements reproduced during the experiment. The main event took place inside a geodesic dome with a 360-degree view.

The instantaneous values and the average electrode states help the user to identify how to modulate the desired brain oscillations. These two variables only have a direct effect on the presentation of the avatars (size, color, and shininess). The animations, graphical items, and music compositions are selected according to the cumulative scores. Only the scores can modify the sound effects and the graphical components of the presentation.

Figure 8.3. Avatars that are used to represent the mind states of the participants in the MVD experiment. The size, position, and brightness correspond to the instantaneous state, the cumulative score and the average electrode state.

In the dome, the projection covered almost the complete hemisphere in such a way that the bottom side of the animation was situated a meter above ground. At the same time, the top side of the scene was close to the highest part of the dome (Figure 8.4). The scene with the avatars was repeated twice in the projection with the aim of facilitating the view for observers situated at different points

inside the semispherical structure.

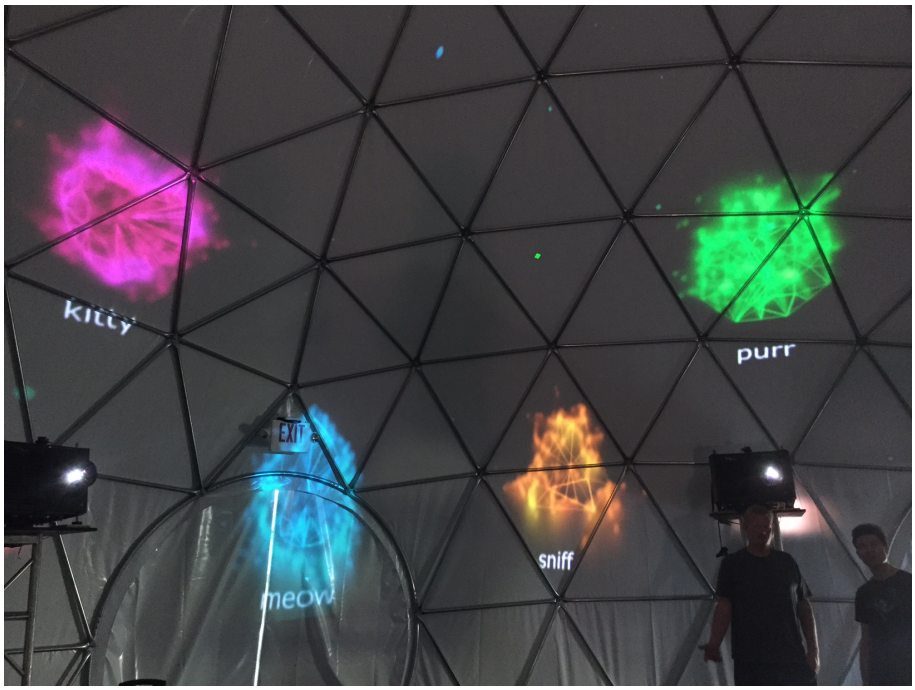


Figure 8.4. Avatars projected in the dome. Their positions with respect to the top of the dome represent the cumulative scores of the users.

The EEG activity was recorded using commercial grade EEG headsets (Emotiv Epoc, 14 sensors, 128 Hz of sampling rate). Only four sensors were used to quantify the brain oscillations: P7, P8, O1 and O2. A fifth sensor (AF3) was also recorded to project in some parts of the experiment the EEG signals of the participants. Figure 8.5 shows the positions of the sensors used in the MVD experiment. In total, 64 subjects participated in the experiment. All of them were informed that their EEG activity would be recorded for scientific purposes, and signed a consent sheet.

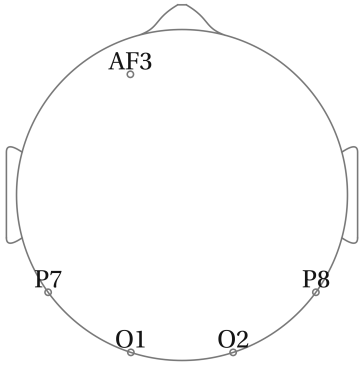


Figure 8.5. Electrode positions used to collect data in the MVD experiment.

Each experimental session in MVD consists of the following stages:

1. *Preparation time* (2-5 minutes). Volunteers help to place the headsets on the participants and explain the general idea about MVD.
2. *Live EEG traces* (1-2 minutes). The EEG signals of the participants (only one channel) are projected on the scene (Figure 8.6). In this stage, the presenter of the event explains what the

charts represent, what is the electroencephalography, and what actions perturb the EEG. The participants can blink, talk and chew to see the effect of these movements in the projected signals.

3. *Neurofeedback training 1* (2-3 minutes). The presenter explains the meaning of the avatars, and what happens when the system detects artifacts. In this case, the avatars turn gray and stand still until the BCI detects valid signals. The participants are instructed to relax with the aim of reaching the highest possible score. The presenter also describes some possible strategies to get relaxed.
4. *Neurofeedback training 2* (2-3 minutes). The participants are instructed to concentrate with the aim of reaching the lowest possible score. Again, the presenter helps the participants with some strategies that may increase the beta activity.
5. *Dream preparation* (1 minute). The presenter indicates that the virtual dream is about to start. The music compositions and the dream themes are selected in this stage. A dream theme consists of backgrounds and graphical components that are added and removed from the projection according to the subjects' performance. One example of a dream background is shown in Figure 8.7. In total, three themes are selected randomly.
6. *Dream 1* (3 minutes). The animations of the first dream are shown in this stage. The subjects are free to select between relaxing and focusing. When one subject has a high cumulative score, part of the background and the graphical components painted with the color of the own avatar are shown. At the same time, each participant controls in a similar way a set of instruments which fade in and out according to the avatar's position. The task of the subjects is to discover what elements are controlled by each one.
7. *Dream 2* (3 minutes). The animations of the second dream are shown in this stage. The subjects interact with the interface in the same way as they do it in the first dream.
8. *Dream 3* (3 minutes). The animations of the third dream are shown in this stage. The subjects have control of the graphical components, but a musician and a singer have a live performance in this dream. They select the musical theme according to the tendency of the group to be relaxed or focused in the past dreams.
9. *Wake up* (1 minutes). The virtual dream ends at this point showing the EEG signals again.

MVD has also been adapted for small presentations in museums and scientific demonstrations. In this case, LCD monitors show the animations and interactive components (see Figure 8.8). The live performance is replaced by musical compositions which are also controlled by the participants. The EEG headsets were also changed by commercial headbands (MUSE, four sensors, 220 Hz of sampling rate). The electrode positions of these headbands are shown in Figure 8.9.

In MVD, the BCI is configured to process two cognitive features: normalized alpha power and normalized beta power. For each feature, the system computes one significance index. Alpha activity is used to increase the cumulative score ($\Delta_{h,\alpha} = 0.2, \Delta_{l,\alpha} = 0.0$), whereas beta activity decreases the cumulative score ($\Delta_{h,\beta} = 0.0, \Delta_{l,\beta} = -0.2$). Artifacts do not modify the cumulative score ($\Delta_c = 0.0$), but new instantaneous states are only processed if at least 25% of the channels are free of noise ($T_{c_1} = 0.25$). The instantaneous feedback represented in the interface corresponds to the normalized activity that provided the last change in the cumulative score. If alpha and beta waves have high amplitudes at the same time, the one with the highest index in absolute value is used as feedback.

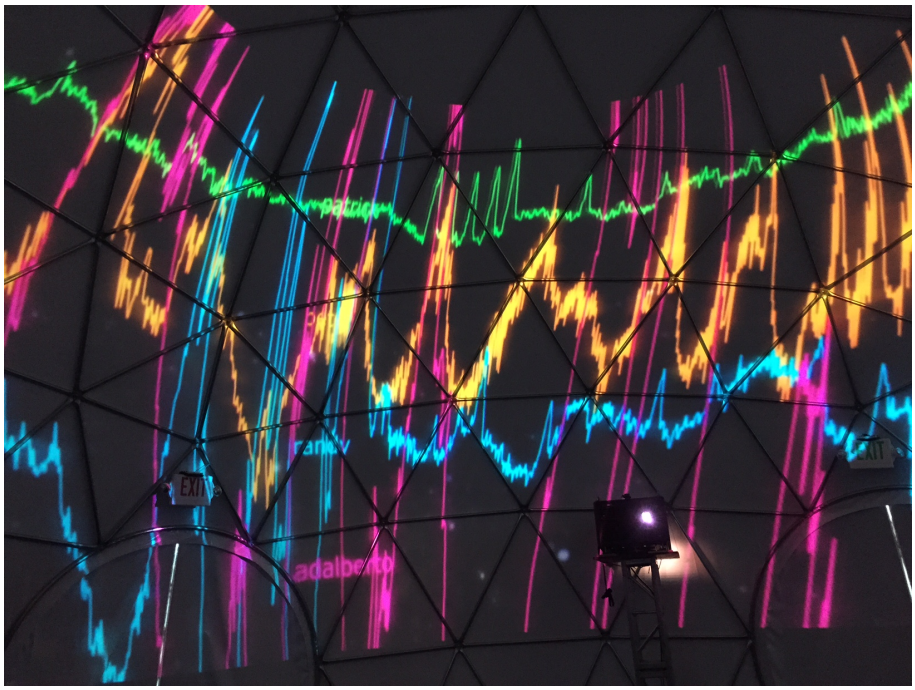


Figure 8.6. Individual traces of the participants. At the beginning and at the end of the presentation, the users see their EEG activity projected in the dome and receive a brief explanation about what is being monitored.

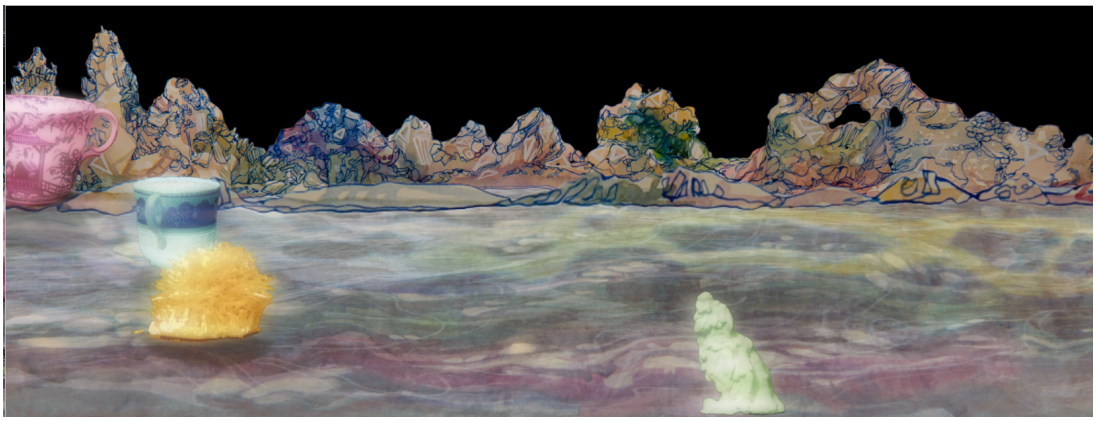


Figure 8.7. Example of a dream theme used in the MVD experiment.

There are not training routines or specific moments in MVD that are used to collect data for the neutral condition. The reason of this design is to reduce the feeling that the participants are involved in one formal experiment. Instead, the system computes the test distributions with data collected before the computation of new scores. The BCI uses a continuous adaptation approach in which each new score is obtained by analyzing the previous time samples.

The Kalman filter was not used in this application because the BCI requires time to estimate the VAR model and initialize the covariance matrices of the filter. Unfortunately, the continuous adaptation for four participants demands most of the computational resources and the incorporation of the Kalman filter is not possible in MVD.

The epoch length and the number of observations of the training set are another critical parameters of the BCI that need to be defined before the online run. Long epochs reduce the variability

Figure 8.8. Another public event of My Virtual Dream. The multimedia elements were shown on LCD screens. The EEG headsets were also replaced by headbands with only four sensors.

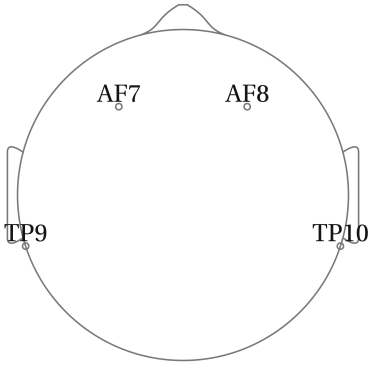


Figure 8.9. Electrode positions of the MUSE system used in the MVD project.

between contiguous observations but produce slow variations in the cumulative and instantaneous scores. In contrast, short epochs allow observing faster variations, but the calculated scores are less stable. In general, epoch lengths between one and two seconds are adequate for MVD. At the same time, the software that renders the multimedia components creates smooth animations for the avatars, so that the variability between consecutive scores is not perceived by the user even if short epochs are used to compute the cognitive scores.

With regards the training set size, a small number of epochs may produce unstable results, but the time required to adapt the system to new conditions is shorter than the arising from having more epochs in the training set. Figure 8.10 illustrates the median values of the normalized alpha power in the training sets of one participant calculated for different numbers of epochs. The sequence of values cover the three dreams of the experiment. In this example, the epoch length is one second. The median value is used in this comparison because thresholds for significance indices are computed

from quantiles of the training set. It is expected that a similar behavior should be observed if another quantile (for instance 0.7) were used.

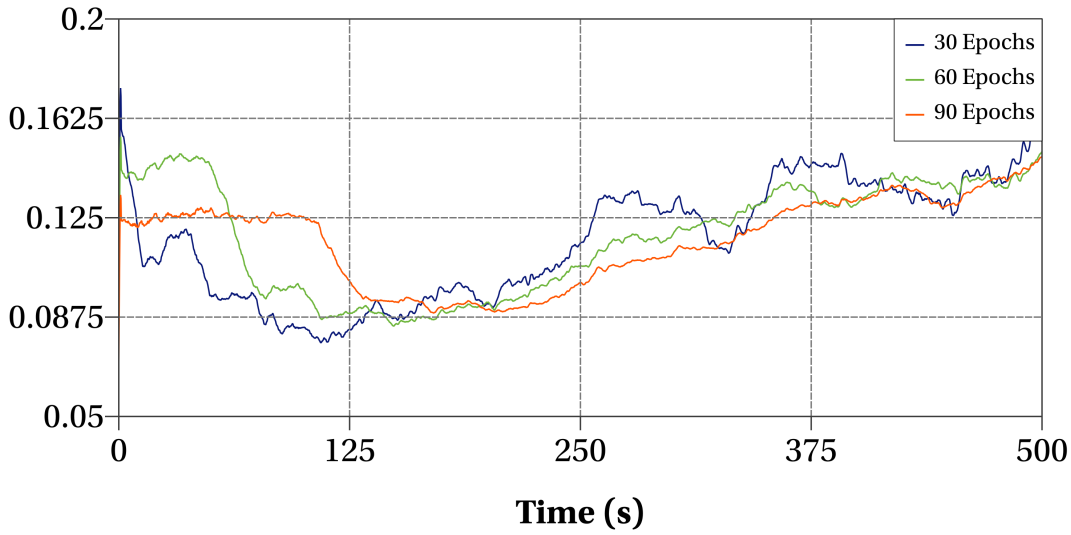


Figure 8.10. Comparison of median values of the normalized alpha power calculated for different training set sizes.

The origin in the chart corresponds to the beginning of the first dream. Before this moment, the participants receive instructions from the presenter and try to play with the avatar. When the first dream starts, the experimental conditions change drastically, so that it is possible to observe significant shifts in the median power values as it is shown in the plot. With 90 epochs in the training set, the system requires almost two minutes to reach a similar value than the observed with 60 or 30 epochs. Local variations of the median value are observed more often when the training set only has 30 epochs. According to this chart, training sets of 60 epochs are a reasonable selection for this application.

It is important to highlight that one disadvantage of the continuous adaptation in neurofeedback experiments is that the thresholds used to evaluate the user's performance vary on time. For example, the continuous adaptation of the threshold for alpha oscillation may increase the difficulty of the task when the participant begins to understand how to modulate this activity. However, according to the user's feedback, the continuous adaptation was not a problem for most of the participants. As mentioned above, 73% of the subjects reported having some degree of control over the avatars. One of the strategies that allowed the subjects to control the interface consisted in using low thresholds for the significance indices ($T_{h,\alpha} = 0.2$, $T_{l,\beta} = -0.2$). In practice, if one subject produces alpha or beta activity above the median value, the system provides a reward and modifies the status of the avatar. These actions are easily perceived by the participants and help to illustrate what mental conditions may control the interface.

Figure 8.11 shows one example of scores calculated for one subject during the first and second dream. The charts indicate the individual indices for alpha and beta oscillations, the cumulative score and the instantaneous feedback. It is interesting to point out that the indices for alpha and beta oscillations are not easy to interpret in the plot, but the cumulative score and the instantaneous feedback provide a better insight of the user's brain oscillations.

In the MVD experiment, the participants receive a different kind of reward according to their

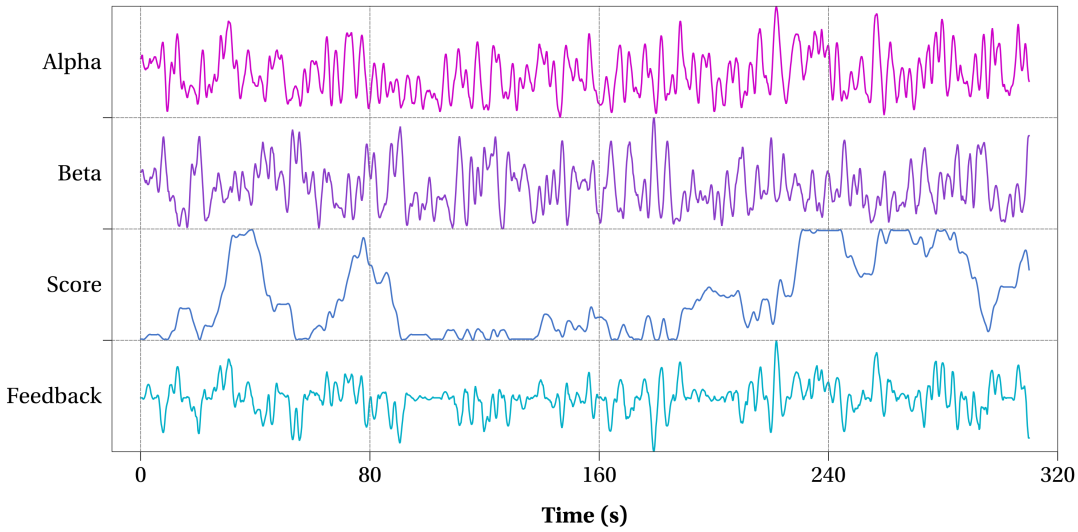


Figure 8.11. Cognitive scores calculated in MVD for one subject during the first and second dreams.

relaxation and concentration levels. When users have high relaxation levels, music and visuals are richer and more interesting. On the contrary, high concentration levels reduce the number of multimedia elements to represent a peaceful environment. Theoretically, the participants are encouraged to relax with the aim of discovering the visual and music effects. However, the users are free to decide what to do in the experiment, so that it is not expected to observe a common pattern for all of them in an offline analysis. For this reason, each subject exhibits a different story in one MVD session, as is illustrated in Figures 8.12 and 8.13. These charts represent the local average (15 seconds of epochs) of alpha and beta power calculated during the three dreams.

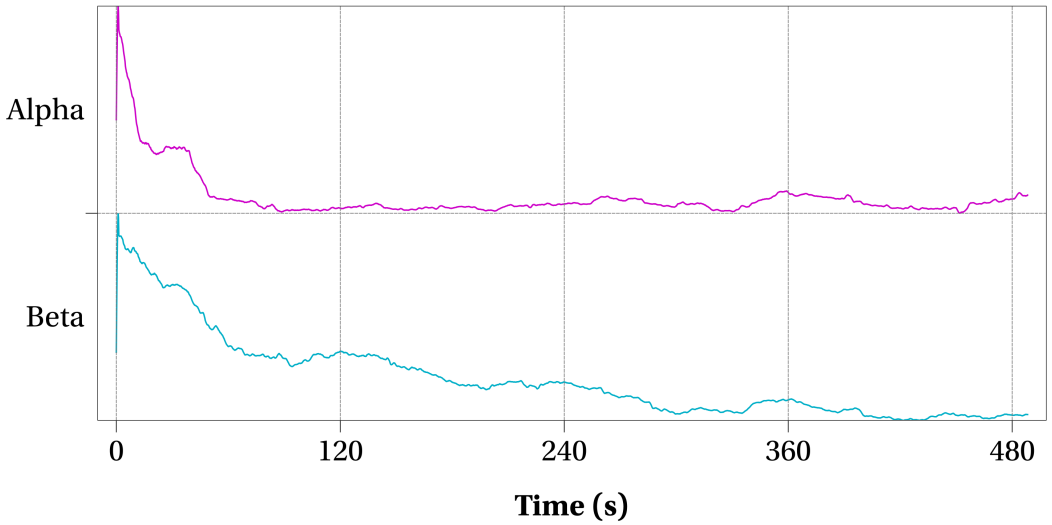


Figure 8.12. Local averages of alpha and beta power calculated for one participant in the MVD experiment.

As can be seen from the charts, the first participant did not exhibit important changes in alpha activity after the settling time, but the beta power constantly decayed during the experimental session. In contrast, the second participant had an increase in alpha after 360 seconds, but the beta activity

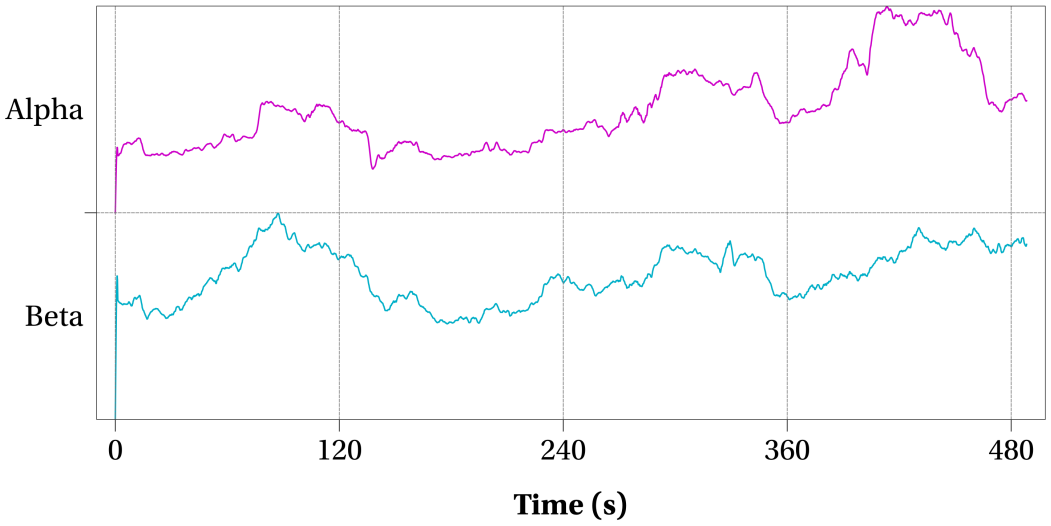


Figure 8.13. Local averages of alpha and beta power calculated for another participant in the MVD experiment.

oscillated after the 60 seconds. Both participants tried different ways to control the avatar, and in the end, they had different experiences. The same kind of analysis may be applied to all the participants, but the results are unique for each subject as a consequence of the experiment design.

8.5 Conclusion

This chapter described the passive BCI implemented to quantify cognitive states. This subsystem in the last component incorporated into the hybrid architecture of the autonomous wheelchair and has the intention to interpret certain user's states during the operation of the control interface. This BCI was presented as a general tool for control applications and neurofeedback protocols. It is highly customizable and can be adapted to different scenarios.

In the proposed cognitive state detector, the user's performance is represented by a scoring system that indicates three types of variables: signal states, instantaneous states and cumulative scores. This representation of the brain activity was tested in one novel neurofeedback protocol, in which relaxation and concentration levels are quantified according to the amplitude of alpha and beta oscillations. Most of the participants achieved some degree of control, which demonstrates that the implemented BCI can be used in neurofeedback applications as an interpreter of cognitive information.

Chapter 9

Performance Evaluation of the Wheelchair BCI Controller

All the components of the hybrid architecture have been described in previous chapters including the proposed control interface for autonomous devices. With the aim of finding the optimal configuration of each processing stage, the four implemented BCIs were tested individually in offline and online conditions. Finally, this chapter presents the performance evaluation of the entire hybrid BCI designed to operate the autonomous wheelchair.

The software included in the wheelchair has a simulator which is used to test the navigation system indoors without the need of the autonomous device. It provides all the information that the BCI needs to update the GUI. The online tests of the proposed BCI were conducted with the simulator and the autonomous device. Each section in this chapter indicates what tests were performed in the simulated environment.

The following sections summarize the results obtained in the online tests of the hybrid BCI. The final sections present a comparison between the proposed system and other brain-controlled wheelchairs reported in previous studies.

9.1 Experimental Setup

As detailed in Chapter 3, the control interface of the autonomous device can work in four different ways:

- *Hybrid control interface for indoor navigation.* The user can select with the BCI target places and basic navigation commands.
- *P300-based interface for indoor navigation.* The user can select the same options included in the hybrid interface.
- *SSVEP-based BCI for active steering* (indoor and outdoor).
- *P300-based interface for outdoor navigation.* The user only can select known target places.

Each operation mode was evaluated as follows. Ten participants tested the hybrid control interface with the simulator (M = 7, F = 3), and five subjects with the autonomous wheelchair (M = 4, F = 1). The same participants evaluated the P300-based interface for indoor navigation. Other two participants (M = 2, F = 0) tested the P300-based interface for outdoor navigation, and finally, three users tried the

active steering (M = 3, F = 0).

The results presented in this chapter are divided into four groups according to the method used to test the interface. One group represents the subjects that tested the interface with the simulator. The second group corresponds to the participants that tested the indoor navigation with the hybrid BCI and the P300-based controller. The third group indicates the participants that operated the wheelchair outdoors. Finally, the four group corresponds to the participants that moved the wheelchair with the SSVEP-based BCI for active steering.

The tests with the wheelchair simulator were performed with a 23-inch monitor, while the tests with the autonomous device were conducted with an 11.6-inch touch screen. The refresh rate in both cases was 60 Hz.

At the beginning of an experimental session, the participant trained the P300-based BCI with the routine described in Chapter 5. Then, the SSVEP-based BCI was fitted with resting state data (60 seconds of EEG samples). Afterwards, the subject performed some selections to calibrate the frequencies and thresholds of the SSVEP detector. The frequencies of the flashing stimuli were 8.57 Hz and 12 Hz. Only participants 6 and 8 of the group that tested the BCI with the simulator required other stimulus rates (7.5 Hz and 8.571 Hz for participant 6, 6 Hz and 15 Hz for subject 8).

The resting state data was also used to train the cognitive state detector and the SSVEP-based BCI for active steering. The frequencies used for free navigation were 10 Hz, 12 Hz and 15 Hz for all the users. Finally, only two participants had available training data for the ErrP detector. In these cases, another test was performed to evaluate the user's experience when this passive system works with the rest of the BCIs.

The online tests consisted of a sequence of selections performed by the participants using one of the previously described operation modes. They were free to choose any option of the interface. At the same time, they could make new selections while the wheelchair was in transit. The system measured the time required to choose one of the flashing options with the SSVEP detector, and the number of flashes required by the users to make one selection with the P300-based BCI. These two variables were used to evaluate the overall performance of the BCI. In the case of false positives, the users were instructed to report what option was chosen incorrectly and select the stop command to cancel the last action.

The ten participants that used the simulator performed in total 15 command selections with the hybrid interface, 5 selections with the P300 interface for indoor navigation and 5 selections with the interface for outdoor control. The five participants that operated the wheelchair indoors performed 10 selections with the hybrid system and 10 with the P300 interface. Finally, the two subjects that tested the wheelchair outdoors selected 10 target places in the experimental session. Each subject tested the interface for around 40 minutes.

In the pre-processing stage, a Butterworth band-pass filter (IIR) of 0.1 to 40 Hz was applied to the EEG signals. Another FIR filter from 4 to 14 Hz was also applied to extract the EEG components processed in the P300-based BCI. Re-referencing techniques and artifact suppression were not used in these tests. Each processed epoch was validated according to the heuristic described in Chapter 4. In the P300-based BCI and the ErrP detector, the data was separated in epochs of 200 ms of pre-stimulus activity and 800 ms of post-stimulus activity. In the SSVEP-based BCI and the cognitive state analyzer, the EEG was separated in overlapped epochs of one second at a rate of five times per second.

LDA and stepwise regression were used to find the classification model of the ERP-based BCIs. The system calculated spatial filters in the feature extraction stage of the P300 classifier with the canonical correlation analysis applied to target responses (CCA2). CCA applied to the target and non-target observations (CCA3) were used in the ErrP-based BCI. The parameters n_f^{min} and q^{max} that control the error rate of the P300 classifier were set to 10 and 0.05 respectively.

The SSVEP detector used the classification method based on significant increments of SSVEP-related activity. Features were calculated with the LRT algorithm. In the hybrid control, the system processed twenty consecutive epochs before selecting one option of the interface. At least twelve trials with significant SSVEP activity for one flickering frequency were required to execute one command. In the active steering, the number of processed epochs were ten. The minimum number of trials with SSVEP-related activity to send one command was six.

The cognitive state analyzer was configured to detect task engagement. The BCI computed the ratio between beta power and the sum of alpha power and theta power to characterize this cognitive state. Bandpass FIR filters were applied to extract the oscillatory components of these rhythms. In the scoring system, high cumulative values indicated significant increments of task engagement while low cumulative values were used to represent significant decrements.

The stimulus presentation of the P300 interface consisted of standard flashing icons for all the subjects, except for participants 4 and 8 of the group that tested the BCI with the simulator. In these cases, happy faces were used to highly the options instead. In the same interface, the options flickered for 150 ms. However, subjects 7 and 9 of the group that used the simulator, and participant number 3 of the group that evaluated the wheelchair indoors, preferred faster-flashing rates (100 ms).

9.2 False Positives, Classification Errors and Average Selection Times

Errors in the classification stage can be separated into two different categories: false positives (or false alarms), which occur when the subject is not using the interface, but the BCI selects one option; and classification errors, which happen when the system chooses an option incorrectly.

In the hybrid BCI, false positives in the P300-based interface are not frequent because of the locking mechanism. The user interacts with this BCI only when needed. As a result, in the experiment, no false positives were produced by the P300 classifier. In contrast, in the P300-based controller, false positives are more common because the interface is processing new data all the time. In the experimental protocol, the selections with the P300-based controller were consecutive, so that no false positives were detected.

However, another test was performed to determine how easy is to trigger one false positive with the P300-based controller. The participants were instructed to talk, read, or do anything else until the system selects one option incorrectly. For all the participants, the BCI produced at least one false positive in less than five minutes.

Tables 9.1, 9.2 and 9.3 show the classification errors detected during the 40-minute experiments conducted to evaluate the wheelchair BCI controller. Each table presents the number of errors for one testing group.

Among all the tests, only four times the P300-based BCI classified incorrectly. Considering each subject performed more than twenty selections with the P300-based BCI, the error rate is below the

BCI	Subject									
	1	2	3	4	5	6	7	8	9	10
P300	0	1	0	1	0	0	0	1	0	0
SSVEP	0	0	0	0	0	0	0	0	0	0

Table 9.1. Classification errors detected in the online tests conducted with the wheelchair simulator for indoor navigation.

BCI	Subject				
	1	2	3	4	5
P300	0	0	0	1	0
SSVEP	0	0	0	0	0

Table 9.2. Classification errors detected in the online tests of the wheelchair controller for indoor navigation.

BCI	Subject	
	1	2
P300	0	0

Table 9.3. Classification errors detected in the online tests of the wheelchair controller for outdoor navigation.

desired level (5%). Nevertheless, the false positives may be reduced by decreasing the parameter that controls the misclassification rate in the proposed adaptive strategy. The natural consequence of reducing the false positives would be an increase in the selection times, but it would help to avoid the execution of incorrect commands.

The SSVEP-based BCI always classified correctly when the users were looking at one of the flickering stimuli. However, it is important to point out that it is more likely to observe false positives with this type of BCI in the hybrid architecture. Tables 9.4 and 9.5 show the false positives detected with the SSVEP-based BCI.

Subject									
1	2	3	4	5	6	7	8	9	10
0	0	0	1	1	2	0	2	1	0

Table 9.4. False positives produced by the SSVEP-based BCI in the online tests conducted with the wheelchair simulator for indoor navigation.

About 50% of the participants that tested the SSVEP interface of the hybrid controller did not have false positives during the experiment. In the worst case, two false positives were detected by the BCI. Taking into account that the SSVEP-based BCI was tested for around 30 minutes, no more than 0.067 false positives per minute are expected with the selected configuration. Longer classification

Subject				
1	2	3	4	5
0	2	0	2	0

Table 9.5. False positives produced by the SSVEP-based BCI in the online tests of the wheelchair controller for indoor navigation.

windows may reduce this value, but the trade-off between speed and accuracy must be considered in the calibration process of the BCI.

In the hybrid architecture, the P300-based BCI is used to select target places and commands organized in nested menus. Each section of this menu may have a different number of available options so that the selection times vary according to the target option. To measure the speed of the interface, the system determined the number of times that a target icon flashes before being selected. Then, the mean number of flashes was calculated, which is used to estimate the expected selection time for a particular number of options. Tables 9.6, 9.7 and 9.8 present the selection times for various numbers of available choices in the P300 interface. The same tables also contain the times obtained for the SSVEP-based BCI.

Subject	SSVEP	P300					
		8	9	10	11	12	13
1	5.5	9.60	10.80	12.00	13.20	14.40	15.60
2	5.9	7.76	8.73	9.70	10.67	11.64	12.61
3	5.8	8.03	9.03	10.03	11.03	12.04	13.04
4	7.2	11.93	13.42	14.91	16.40	17.89	19.38
5	7.5	8.18	9.20	10.22	11.24	12.26	13.28
6	10.1	10.48	11.79	13.11	14.42	15.73	17.04
7	4.9	7.30	8.22	9.13	10.04	10.96	11.87
8	5.1	10.36	11.65	12.95	14.24	15.54	16.83
9	13.1	13.72	15.44	17.15	18.87	20.58	22.30
10	5.8	10.64	11.97	13.30	14.63	15.95	17.28
Mean	7.1	9.80	11.02	12.25	13.47	14.70	15.92

Table 9.6. Average selection times in seconds obtained in the online tests of the hybrid BCI controller for indoor navigation with the wheelchair simulator. The table indicates the selection times for different number of options shown in the P300 menu.

In the P300-based BCI for indoor navigation, selection times vary from 10 seconds to 16 seconds depending on the number of options shown in the interface. Most of the participants required between five and ten flashes to perform one selection. These results are the expected according to the tests described in Chapter 5, which suggest that accuracies close to 100% are possible with five or more flashes with the proposed system. As can be seen from the results, the adaptive mechanism implemented to control false positives allows finding an optimal balance between speed and accuracy.

Subject	SSVEP	P300					
		8	9	10	11	12	13
1	5.6	9.11	10.25	11.39	12.53	13.66	14.80
2	9.6	10.50	11.81	13.13	14.44	15.75	17.06
3	4.8	5.77	6.49	7.21	7.93	8.65	9.37
4	5.2	9.20	10.35	11.50	12.65	13.80	14.95
5	5.9	13.44	15.12	16.80	18.48	20.16	21.84
Mean	6.2	9.60	10.80	12.00	13.20	14.40	15.61

Table 9.7. Average selection times in seconds obtained in the online tests of the hybrid BCI controller for indoor navigation. The table indicates the selection times for different number of options shown in the P300 menu.

Subject	P300					
	3	4	5	6	7	8
1	3.95	5.27	6.58	7.90	9.22	10.53
2	3.85	5.13	6.41	7.69	8.97	10.25
Mean	3.90	5.20	6.50	7.80	9.09	10.39

Table 9.8. Average selection times in seconds obtained in the online tests of the hybrid BCI controller for outdoor navigation. The table indicates the selection times for different number of options shown in the P300 menu.

Moreover, it is interesting to point out that similar results were obtained with the simulator and the autonomous device.

Selection times for the SSVEP-based BCI are between five and seven seconds. Some participants required longer times because their EEG responses to the flashing stimuli are weaker than the rest. However, they did not report any problem in controlling the interface. Moreover, the results obtained in the online tests with the simulator and the wheelchair are similar to the selection times presented in Chapter 7. The increase in the number of processed epochs before any selection did not produce longer average times, but helped to reduce the false positive rate. However, in this case, the minimum time to choose one option was four seconds. In the tests reported in Chapter 7, some participants could generate control commands after two seconds.

9.3 Indoor Navigation

Figure 9.1 shows the map of the office building where the autonomous wheelchair was tested indoors. In this place, the wheelchair travels around 50 meters to reach the farthest room from the starting point. For the experiments, there was no special preparation. Students could transit the building as normal. Trash cans and other obstacles were not removed, and only the doors of some rooms were opened. This experimental setup intended to preserve the real operating conditions of the autonomous wheelchair.

As explained in Chapter 3, the selection of a target place or a navigation command is performed

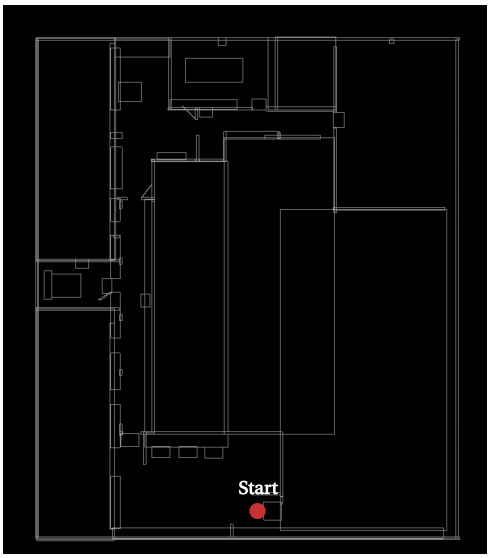


Figure 9.1. Map of the office building where the autonomous wheelchair was tested.

in two or more steps. For instance, if one target location is shown on the main map, the subject must unlock the P300-based interface by selecting with the SSVEP-based BCI the command *"start P300"*. Then, the user has to choose a target place with the P300-based BCI. However, if the desired location belongs to a group of the main map, the subject has to perform two selections with the P300-based BCI. Figure 9.2 shows an example of how the BCI is used to select a target place with the hybrid BCI. Three consecutive single selections are needed in this case to specify the destination.

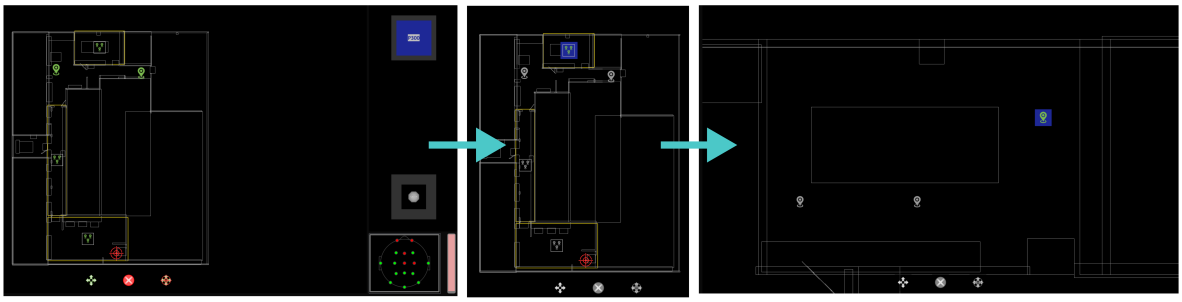


Figure 9.2. Example of a target place selection with the hybrid BCI. The blue icons represent the options chosen at each step.

The *"stop"* command only requires one selection with the SSVEP-based BCI, while target place needs two or three consecutive selection steps, one with the SSVEP-based BCI and the rest with the P300-based BCI. Finally, after unlocking the P300-based BCI, low-level and high-level navigation commands (*"go straight"*, *"turn left"*, *"rotate 90º"*, etc.) only require one selection. The P300-based BCI can also be locked again by selecting the *"cancel"* option or the *"stop"* command.

The average selection times of the hybrid controller obtained in the experiments are shown in Tables 9.9 and 9.10. Likewise, the results of the P300-based controller are shown in Tables 9.11 and 9.12. Between selection steps, the resting time was three seconds. The tables specify the results for some particular actions that the subjects can perform with the BCI. For target places, it is indicated the minimum and maximum times required to choose one location.

Action	Subject										
	1	2	3	4	5	6	7	8	9	10	Mean
Start P300	5.50	5.90	5.80	7.20	7.50	10.10	4.90	5.10	13.10	5.80	7.09
Stop	5.50	5.90	5.80	7.20	7.50	10.10	4.90	5.10	13.10	5.80	7.09
Target Selection (Minimum)	18.10	16.66	16.83	22.13	18.68	23.58	15.20	18.46	29.82	19.44	19.89
Target Selection (Maximum)	40.30	35.18	35.88	48.98	38.03	47.55	32.81	42.17	60.26	43.71	39.49
High-Level Navigation	9.60	7.76	8.03	11.93	8.18	10.48	7.30	10.36	13.72	10.64	9.80
Low-Level Navigation	14.40	11.64	12.04	17.89	12.26	15.73	10.96	15.54	20.58	15.95	14.70

Table 9.9. Average selection times in seconds for different types of commands available in the hybrid BCI controller. The results shown in the table were obtained with the wheelchar simulator.

Action	Subject					
	1	2	3	4	5	Mean
Start P300	5.60	9.60	4.80	5.20	5.90	6.22
Stop	5.60	9.60	4.80	5.20	5.90	6.22
Target Selection (Minimum)	17.71	23.10	13.57	17.40	22.34	18.82
Target Selection (Maximum)	38.93	47.10	28.10	38.80	52.22	41.03
High-Level Navigation	9.11	10.50	5.77	9.20	13.44	9.60
Low-Level Navigation	13.66	15.75	8.65	13.80	20.16	14.40

Table 9.10. Average selection times in seconds for different types of commands available in the hybrid BCI controller. The results reported in the table were obtained with the real autonomous wheelchair.

Considering all the subjects, on average, the participants could stop the wheelchair with the hybrid controller in 6.8 seconds. Most of the users did not need more than six seconds to perform this action. In contrast, the same participants required around 10.94 seconds to stop the wheelchair with the P300-based controller. However, the hybrid interface demands more time to select target places. Finally, the selection of consecutive low-level and high-level commands need almost the same time with both types of controllers.

The cognitive effort for selecting the *"stop"* command is another difference between types of controllers. In the hybrid interface, users only need to look at the flashing stimulus, whereas in the P300-based controller, they have to count flashes. Almost all the participants reported that they prefer the hybrid strategy for this operation. The increase in the selection time for target places was not a problem for the subjects. However, one participant preferred the P300-based controller because this

Action	Subject										
	1	2	3	4	5	6	7	8	9	10	Mean
Stop	10.80	8.73	9.03	13.42	9.20	11.79	8.22	11.65	15.44	11.97	11.02
Target Selection (Minimum)	10.80	8.73	9.03	13.42	9.20	11.79	8.22	11.65	15.44	11.97	11.02
Target Selection (Maximum)	34.20	28.22	29.08	41.76	29.57	37.07	26.74	36.66	47.59	37.57	34.85
High-Level Navigation	9.60	7.76	8.03	11.93	8.18	10.48	7.30	10.36	13.72	10.64	9.80
Low-Level Navigation	14.40	11.64	12.04	17.89	12.26	15.73	10.96	15.54	20.58	15.95	14.70

Table 9.11. Average selection times in seconds for different types of commands available in the P300-based BCI controller for indoor navigation. The results shown in the table corresponds to the tests performed with the wheelchair simulator.

Action	Subject					
	1	2	3	4	5	Mean
Stop	10.25	11.81	6.49	10.35	15.12	10.80
Target Selection (Minimum)	10.25	11.81	6.49	10.35	15.12	10.80
Target Selection (Maximum)	32.60	37.13	21.74	32.90	46.68	34.21
High-Level Navigation	9.11	10.50	5.77	9.20	13.44	9.60
Low-Level Navigation	13.66	15.75	8.65	13.80	20.16	14.40

Table 9.12. Average selection times in seconds for different types of commands available in the P300-based BCI controller for indoor navigation. The results reported in the table were obtained with the real autonomous wheelchair.

user has the filling that the flickering stimuli are annoying.

It is also interesting to observe how each user combines the navigation commands with other options of the interface to reach specific places. For instance, some participants indicated that the semi-free navigation commands would be only useful to explore big rooms because they find hard to go through doors with them. However, one participant discovered that it is possible to move the wheelchair to the entrance of one room with this type of commands, and then use the *"stop"* option in front of the door to prepare the wheelchair for another command to go to the other room. This kind of continuous learning is only possible in longer experimental sessions. However, in less than 40 minutes, the participants started to imagine how to solve several navigation problems with the options shown in the interface. Other choices that were attractive to the participants were the discrete rotations because they allow changing the pose of the wheelchair at any moment.

In another experiment, two participants tested the ErrP-based BCI. In this case, the training sets collected in the experiments described in Chapter 5 were used to fit the classifier. Then, the passive

system was used to monitor the selections performed by the P300-based BCI of the hybrid interface. As expected, around 20 percent of the selections were canceled by this subsystem incorrectly. Without this passive BCI, the two subjects did not have problems in operating the interface, which demonstrates that the ErrP detector is not needed when the participants achieve high accuracies with the P300-based BCI.

The results presented in this section reveal that the hybrid BCI is highly efficient in decoding the user's intentions and communicating this information to the autonomous wheelchair. The hybrid strategy brings the possibility to reduce false positives and provides an easy way to select the stop command. For all the participants, the interactive menu was intuitive and easy to operate. However, if one particular subject struggles to control the SSVEP-based BCI, the system provides an alternative in which P300 evoked potentials control everything.

9.4 Outdoor Navigation

For outdoor navigation, the system has been configured to operate in two different places: the central square of a botanical garden (Figure 9.3) and outside a campus building (Figure 9.4). The maps used to indicate target places in the interface are shown in the figures. The results presented in this document were obtained in the botanical garden.

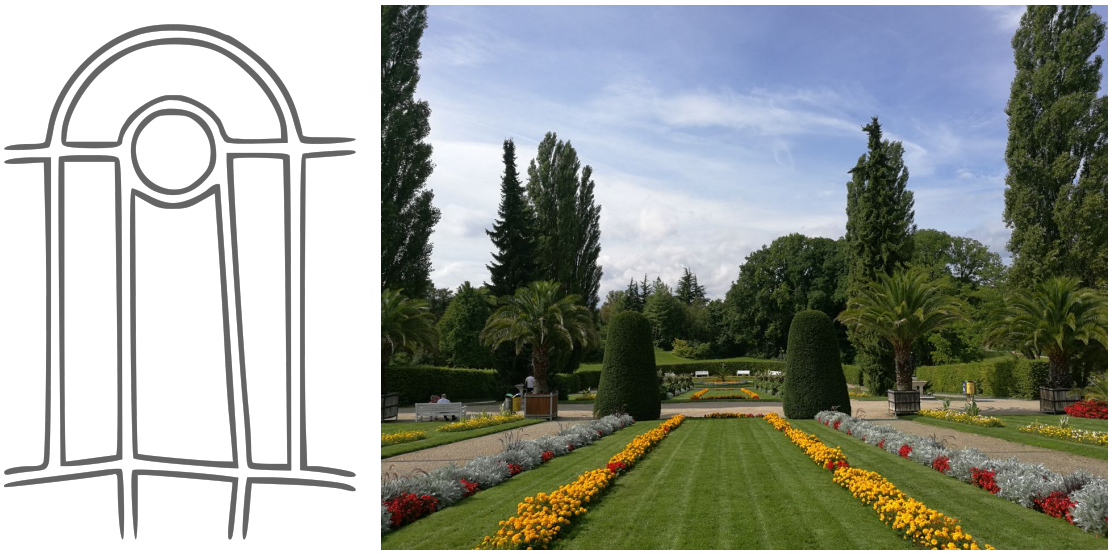


Figure 9.3. Central square of the botanical garden.

The average selection times obtained in the online experiments with the wheelchair are shown in Table 9.13. Both subjects did not report problems in selecting target places with the P300-based interface. As long as the icons can be distinguished in the interface, changes in lighting conditions do not alter the BCI performance. Both participants needed on average between six and seven seconds to select a target place, and around four seconds to stop the wheelchair.

Figure 9.4. Outside the campus building.

Action	Subject		
	1	2	Mean
Stop/Resume	6.58	6.41	6.50
Target Selection	3.95	3.85	3.90

Table 9.13. Estimated selection times in seconds for different types of commands available in the P300-based BCI controller for outdoor navigation.

9.5 Active Steering

In the fourth operation mode of the BCI, the user moves or rotates the wheelchair by looking at one of the three flashing stimuli of the interface. When the BCI detects SSVEP-related activity for one of the options, the frame of the respective icon turns red while the wheelchair controller modifies the position of the device (see Figure 9.5). The icon remains red until the system does not detect more SSVEP-related activity. If the frames of all the icons are gray, the wheelchair receives the *"stop"* command continuously. In this way, the user can select four commands with the interface, three of them by modulating SSVEP oscillations, and the fourth option by not producing SSVEP-related activity.

The information transfer rates in this application are greater than the obtained in the experiments described in Chapter 7 because the BCI is always sending information to the autonomous wheelchair. In the worst case, when the system only can classify 50% of the time correctly, the information transfer rate is about 62.26 bit/min. This classification rate is only obtained in the transitions when the user wants to select another option. Typically, the system needs between one or two seconds to detect a new target option.

The online experiment consisted of moving with the SSVEP-based BCI the wheelchair from a starting point to a final destination (see Figure 9.6). There was no limit time to complete the task. The autonomous wheelchair needs around 155 seconds to reach the target place and travels 31.5 meters

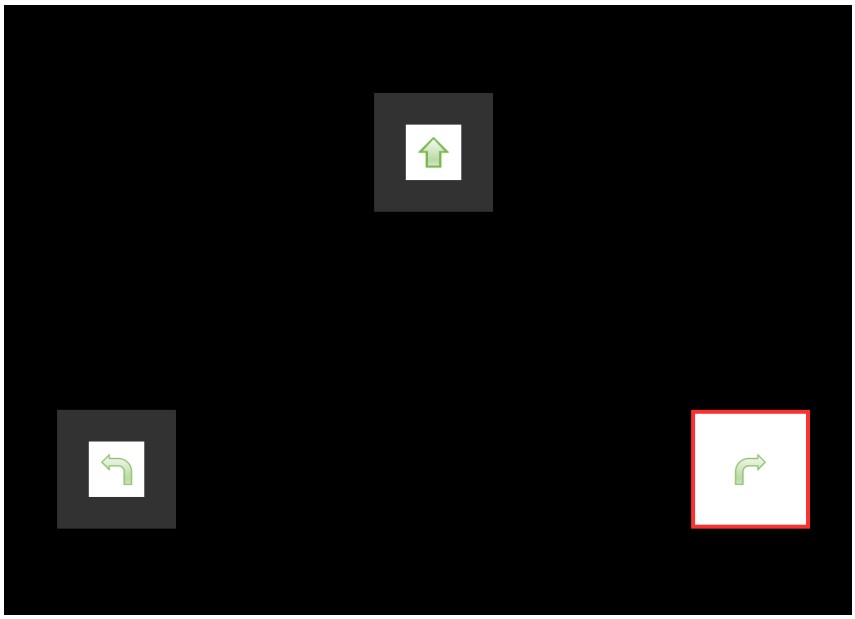


Figure 9.5. SSVEP-based BCI for active steering.

from the starting point.

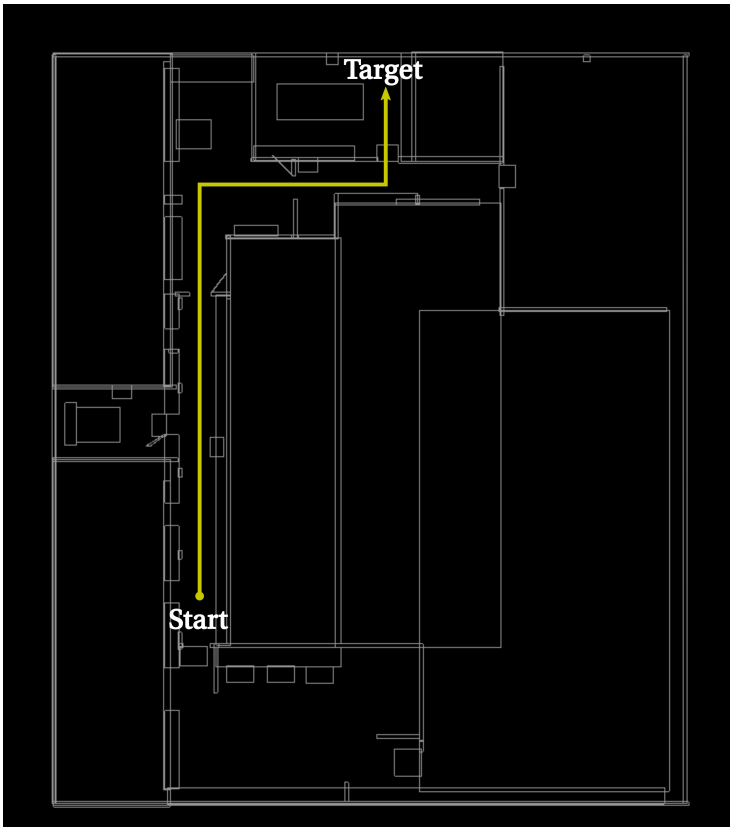


Figure 9.6. Test of the SSVEP-based BCI for active steering.

One subject completed the task in 230 seconds, around 50% more time than the required by the wheelchair to reach the same place with the autonomous navigation. The second participant required 210 seconds, 35% more time than the reference value. Finally, the last participant completed the task

in 190 seconds, only 20% more time than the autonomous navigation. All the participants could finish the experiment in the first trial. None of the subjects reported problems in completing the task.

These results prove that the active steering is possible with the proposed SSVEP-based BCI. Subjects can move and stop the wheelchair freely while the autonomous device corrects the path and avoid obstacles. A dexterous navigation with this BCI would be possible in longer experimental sessions, even though only a few minutes were enough for the subjects to reach the specified place.

9.6 Passive Detection of Cognitive States

The last component of the hybrid arquitecture tested in the experimental protocol was the passive detection of cognitive states. The system was configured to quantify task engagement, which may be characterized as the ratio between beta power and the sum of alpha power and theta power. High cumuative scores indicate high levels of this mind state.

This section presents some examples of cognitive scores calculated at different moments of the experiment. It is not expected to observe the same patterns with all the subjects. However, it is intended to illustrate how the mind states may be related to some tasks and actions performed by the subjects.

Figure 9.7 shows the cognitive score obtained by one participant during the training phase of the P300-based BCI. Gray boxes indicate when the subject had to count flashes. In this example, the cognitive score tends to increase during the counting task. This trend was observed in six out of the eight flashing sequences. The score only had precipitous drops in the fourth block and the eighth block.

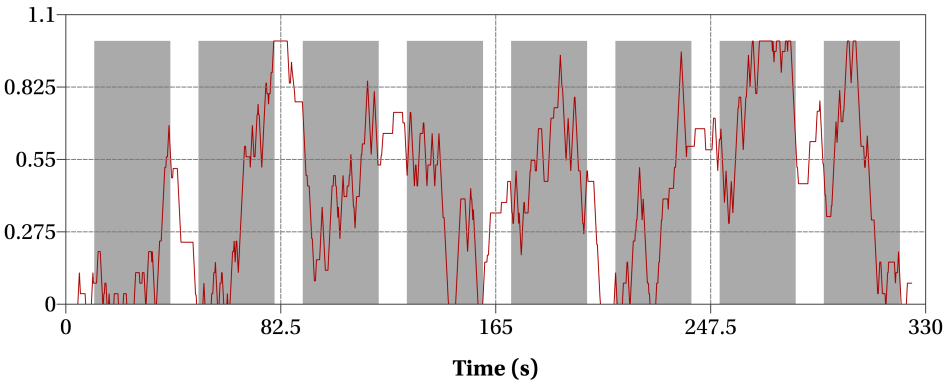


Figure 9.7. Task engagement of one participant quantified during the training routine of the P300-based BCI.

A similar behavior was also observed in other participants in the training routine of the P300-based BCI, as can be seen in Figure 9.8. These results may suggest that it is expected to observe an increase in task engagement when the user is performing a mental counting. This information could be used by the auto-calibration method of the P300-based BCI to identify when the user is counting flashes, which may improve the automatic detection of a valid training set.

Figures 9.9 and 9.10 show the cognitive score for task engagement obtained by two participants during the operation of the autonomous wheelchair. Red lines indicate when the users performed a selection with the hybrid BCI. In the first example, there is a time interval in which the participant did

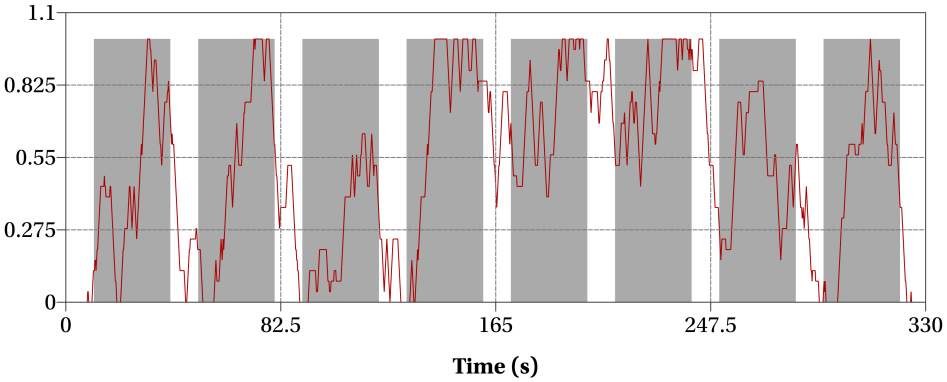


Figure 9.8. Task engagement of another participant quantified during the training routine of the P300-based BCI.

not select anything with the interface. The cognitive score in this range was high, which may indicate that the user was performing another mental while the wheelchair was static.

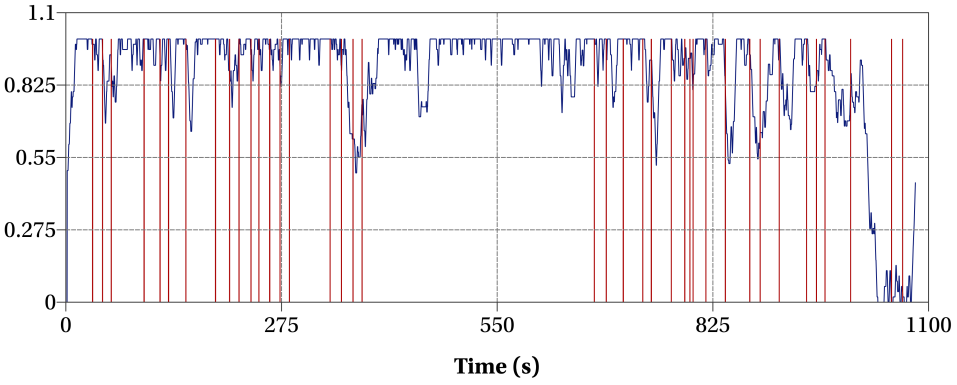


Figure 9.9. Task engagement measured during an online test of the autonomous wheelchair. Red lines indicate when the subject performed one selection with the hybrid BCI.

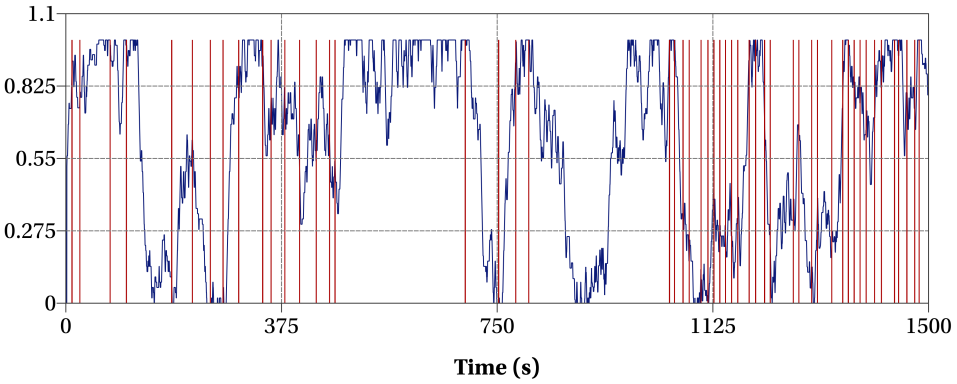


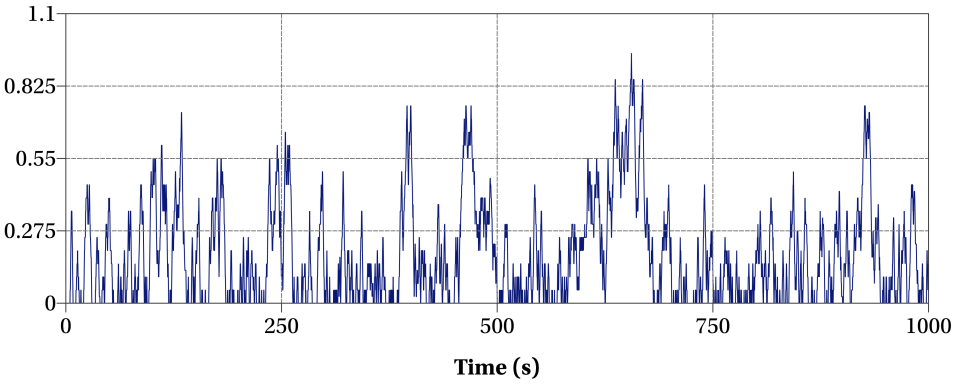
Figure 9.10. Task engagement measured during the online test of the autonomous wheelchair with another participant.

In the second example, there were two time intervals in which the user did not interact with the interface. In the first interval, the subject had high scores, whereas in the second range the cognitive score was lower. This results may suggest that the user was performing different kinds of mental tasks in both cases. It is also interesting to observe that this subject obtained lower cognitive scores than the

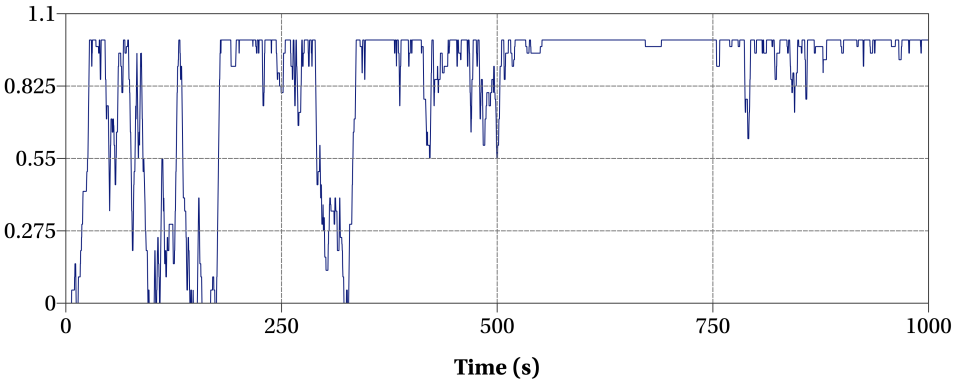
other participant during the selection of target places or navigation commands. This fact illustrates the variability that may be observed in cognitive experiments. Two subjects performing the same task may elicit different brain activity.

Finally, another interesting experiment conducted to evaluate the proposed cognitive score system consisted of measuring fatigue with the interface at two different times. Only one participant performed this test. In (Craig, Tran, Wijesuriya, & Nguyen, 2012) is suggested that this mental state may be characterized by the amplitude of theta oscillations. In this way, the interface was configured to detect significant changes in theta power to score the fatigue level.

The results of this experiment are shown in Figure 9.9. As expected, the cognitive score was greater at 4 a.m. than at 2 p.m. This simple test indicates that the system can measure other variables that may be helpful to increase the safety of the user.



(a) 2 p.m.



(b) 4 a.m.

Figure 9.11. Fatigue levels mesaured for one participant at different times.

9.7 Comparison with Other Brain-Controlled Wheelchairs

To compare this system with other brain-controlled wheelchairs, it is important to consider the control paradigm implemented to decode the user's intentions and how easy is to master the interface. For instance, motor imagery has been employed in previous designs to select a reduced number of options for active navigation. The system described in (Vanacker et al., 2007) and (Galán et al., 2008) is one example of a BCI that follows this strategy. Their brain-computer interface required several training

sessions to find the best mental tasks for each user to control three commands: *"forward"*, *"left"* and *"right"*.

On the contrary, the proposed BCI follows a different approach. For active steering, it was presented a novel method that only requires 60 seconds of training data to find a machine learning model to classify SSVEP-related activity. At the same time, the hybrid controller provides a myriad of options that allow the users to move the wheelchair in known and unfamiliar places. All the participants were able to control the interface without a significant effort, which is an important advantage of the proposed hybrid BCI over systems based on the modulation of sensorimotor rhythms.

The same argument can be used to compare the hybrid BCI with the system described in (Carlson & del R. Millan, 2013). They proposed a wheelchair that was programmed to go forward all the time, while the user controls the direction by producing modulations for two commands: *"left"* and *"right"*. The *"forward"* command was not included in the mental task to reduce the cognitive effort for controlling three or more directions. In the new hybrid BCI, three commands are used for active steering, whereas the *"stop"* option is produced by not looking at any of the flashing stimuli. The possibility to stop voluntarily the wheelchair is an important feature for this technology in realistic scenarios.

Rebsamen et al. (2006) developed an interface for selecting target places in familiar environments with a P300-based BCI. In this case, the wheelchair was restricted to travel only along pre-defined paths to reduce the complexity of the navigation system. Later, the same research group presented in (Rebsamen et al., 2010) a newer version of the same system that improved the selection of the *"stop"* command. The first advantage of the proposed BCI over the brain-controlled wheelchair proposed by Rebsamen et al. (2010) is that the navigation is not restricted to pre-defined paths. The user can select target places or explore freely with the options shown in the interface. This difference is a significant step towards the autonomy of the user. A BCI with only target places is not useful in complex environments and realistic situations. At least one mechanism for free exploration is preferred in any brain-controlled wheelchair. To this end, in the proposed system, two different methods were implemented: one for semi-free navigation controlled by the hybrid interface, and a SSVEP-based interface for active steering.

Another difference between these two BCIs is how the P300-based interface is unlocked. In (Rebsamen et al., 2010), the P300-based interface works like a password protected application. There is one option in the interface for locking the BCI at any time, but the user has to select a sequence of three consecutive flashing stimuli for unlocking. In this way, the user requires at least four consecutive selections with P300 evoked responses to unlock the interface and choose one target place. They reported that a single selection required around 15.4 seconds so that it is possible to say that a complete selection needs more than one minute. In the hybrid interface, the user can select a target place in 19.53 seconds, including the unlocking and resting times. The maximum average time required for selecting target places within groups is 40 seconds. These selection times prove that the SSVEP-based BCI is highly efficient as a mechanism for unlocking the P300 interface.

Finally, the stopping function in the BCI controller described in (Rebsamen et al., 2010) is provided in two different ways: with P300 evoked potentials and with the modulation of mu and beta rhythms. In the first approach, false positives are common, so that the wheelchair stops involuntary in online conditions. Moreover, the participants required several training sessions to master the second method,

and reported that it was difficult to control the modulation of beta and mu rhythms. They obtained selection times between 5 and 8 seconds, depending on the stopping method. In this work, on average, the selection time for the *"stop"* command is 6.8 seconds. All the users could trigger this command without problems so that it can be concluded that the method proposed in this work is easier to master and provides an excellent control of false positives without sacrificing speed.

In a different direction, Iturrate, Antelis, and Minguez (2009) presented a P300-based BCI for selecting where to move next between close and distant locations. The user completes a path by choosing a sequence of target points. The main disadvantage of this strategy is the number of selections with the P300-based interface to complete a navigation task. For instance, it is reported that between 8 and 14 selections and more than 500 seconds were required to navigate between 20 and 40 meters. In contrast, the proposed BCI only requires one selection to reach a target place, and a few of other commands to adjust the pose and the position of the wheelchair. Around 30 meters may be covered in no more than 200 seconds. At the same time, even the active steering is faster than the brain-controlled wheelchair described in (Iturrate, Antelis, & Minguez, 2009).

Another wheelchair controlled by a P300-based BCI was described in (D. Puanhvuan & Wongsawat, 2012), which was improved later in (Dilok Puanhvuan, Khemmachotikun, Wechakarn, Wijarn, & Wongsawat, 2017). This system allows selecting target places (up to nine) and four commands for semi-free navigation (*"forward"*, *"backward"*, *"turn left"* and *"turn right"*). The stimuli are shown in a 3×3 led matrix. The P300-based BCI is unlocked by blinking three times in less than two seconds. This action is monitored in the EEG in frontal channels. Additionally, two blinks help to stop the wheelchair, and four blinks are used to change between the semi-free navigation and the selection of target places.

The stimulation matrix was used to improve the P300 responses. However, the accuracies reported in (Dilok Puanhvuan et al., 2017) are lower than the presented here. They obtained with healthy subjects accuracies between 83.42% and 95.31%, whereas most of the subjects that tested the proposed hybrid BCI achieved accuracies of 100%. These results were obtained as a consequence of the strategy implemented to control the false positive rate and other improvements included in the processing stages. Finally, other stimulation techniques such as icons with happy faces were also included to improve the ERP signals. According to the study presented in this document, the small touch screen was not a problem to stimulate the users.

The led matrix also reduces the number of possible options. In the proposed system, the interface may contain hundreds of target places and navigation commands, whereas in the BCI developed by Dilok Puanhvuan et al. (2017), only nine target places and four navigation commands were included. The GUI design of the hybrid interface is organized in nested menus easy to navigate. Lastly, this BCI also includes low-level commands for another kind of interactions with the wheelchair, which is another advantage of this system.

Again, the stopping method and the unlocking mechanism is another important difference between these BCIs. Probably, the blinking strategy is one of the fastest methods to generate a single command. In only two seconds, the BCI can detect two, three or four blinks with a high accuracy. However, inadvertent commands can be easily produced after long sessions as a consequence of eyestrain and fatigue. People use to blink once involuntarily every four seconds, but this rate may increase because of several factors. Dilok Puanhvuan et al. (2017) did not present a study that estimates the false positive rate and the selection times of this technique. Probably, patterns of three or four

blinks are not so frequent, but two blinks in less than two seconds are more likely. In (Cao et al., 2014), it is shown that a similar approach may produce false positives when the selection times are close to two seconds. However, to reduce the false positive rate, they increased the selection times to more than four seconds but the accuracies were below 90%. In contrast, the proposed SSVEP-based BCI is slower (more than 6 seconds to select one command), but it is confirmed that only a few false positives are expected in long experimental sessions. Furthermore, it is shown that this BCI has accuracies of 100%.

Another hybrid strategy was presented in (Long et al., 2012). This system combines motor imagery and P300 responses to control a wheelchair. The direction is selected by modulating sensorimotor rhythms, while a P300-based BCI coupled with another motor imagery pattern (food movements) is used to increase and decreased the speed of the wheelchair. This BCI does not have a special mechanism for locking and unlocking the P300-based interface, so that classification errors and false positives may change the speed of the wheelchair. Later, they improved this design in (H. Wang et al., 2014) by adding an electrooculogram to detect blinks. In this new version, two, three and four blinks are used to select *"forward"*, *"stop"* and *"backward"*.

In the BCI described in (H. Wang et al., 2014), the average selection time for the *"stop"* command was 4.5 seconds, and the accuracies were between 85% and 90%. The false positive rate was zero. In the proposed system, the accuracy is 100%, but selection times are 6.8 seconds, and in the worst case, it is expected around 0.034 false positives per minute, a half of the rate of the SSVEP-based BCI. On average, the BCI may produce 0.012 false positives per minute for the *"stop"* command. The proposed stopping method is still competitive because of its high accuracy. The user does not need to repeat the same action to halt the wheelchair.

The most significant advantage of the proposed hybrid BCI over the brain-controlled wheelchair developed by H. Wang et al. (2014) is the possibility to select target places and other navigation commands. The selection of target points reduces a lot the cognitive effort to navigate between distant places. Additionally, the semi-free navigation allows exploring big rooms with the execution of a few orders. Other commands that are important are the discrete rotations, which modify the wheelchair's pose without the need of moving the device. This action is impossible in the system described in (H. Wang et al., 2014), because the steering commands are only allowed when the wheelchair is in transit.

Additionally, H. Wang et al. (2014) reported that in one online test, their wheelchair completed a path of around 40 meters in more than 300 seconds. The proposed wheelchair can complete by choosing only one command a path of 30 meters in 170 seconds (150 seconds of autonomous navigation and 20 seconds to select the target place). This difference demonstrates that the autonomous system is more efficient than the active steering to reach a target location.

Another interesting BCI design was proposed by Cao et al. (2014). They developed a hybrid interface that combines motor imagery and SSVEP modulations to select six commands: *"left"*, *"right"*, *"stop"*, *"go"*, *"accelerate"* and *"decelerate"*. The steering functions are controlled with motor imagery, the speed is controlled by a SSVEP-based BCI, and finally, the *"stop"* and *"go"* commands are produced through the two mental tasks. The user has to modulate mu rhythms and SSVEP oscillations at the same time to select these functions. Besides the limitations of this brain-controlled wheelchair, it is not clear the advantages of combining motor imagery and SSVEPs to produce the stop command. On

the contrary, in this document, it is shown that SSVEPs are very effective for stopping the wheelchair. Unfortunately, they did not provide the false positive rates and the selection times for this combination.

Finally, the proposed BCI is the first prototype of a control interface for an autonomous wheelchair that can navigate outdoors. Probably, this is the first time that a P300-based BCI has been tested outdoors to select target places. This is a major step towards the development of 100% functional brain-controlled devices.

9.8 Conclusion

This chapter presented the performance evaluation of the hybrid architecture for controlling autonomous systems. In the proposed hybrid BCI, the user can select with a P300-based BCI target places and navigation commands for semi-free navigation and low-level interactions. The user has available all the functions provided by the autonomous wheelchair. At the same time, the hybrid architecture uses the modulation of SSVEP oscillations to unlock the P300-based interface and choose the *"stop"* command. Moreover, active steering was incorporated into the system for free exploration. Probably, so far, this is the most complete solution for controlling an electric wheelchair with a BCI.

The results described in this chapter reveal that the SSVEP-based BCI is very effective in unlocking the P300-based interface and stopping the wheelchair. The unlocking mechanism reduces the false positives significantly in the P300-based, while the stop command can be triggered easily without cognitive effort. The high accuracy of the SSVEP-based BCI and the short selection times make this novel hybrid design highly competitive in comparison with other brain-controlled wheelchairs.

Selection times of the hybrid architecture are between 10 and 40 seconds, depending on the location of the desired command in the interactive menu. Target places require more time than other options, but they only need to be selected once to reach a distant place. High-level and low-level navigation commands can be selected faster because the interface is not locked automatically after choosing one of them. The online tests showed that the options for semi-free navigation and some low-level commands are useful for the users to explore freely and modify the pose of the wheelchair. These features are not available in other brain-controlled wheelchairs.

Active steering was also tested with the autonomous wheelchair. It was proved that this modality is also possible with the implemented SSVEP-based BCI. The participants completed the navigation task without problems, but they needed between 20% and 50% more time than the required with the autonomous navigation. Furthermore, the ErrP-based BCI was also evaluated. As expected, the classification errors of this subsystem canceled several selections performed by the users, so that this BCI is not required in the hybrid architecture as a consequence of the high accuracy of the P300-based BCI.

Finally, the cognitive state detector was also tested in the experimental protocol. This chapter described some possibilities of this passive BCI to improve the hybrid system. For instance, task engagement could be used to identify when the user is controlling the P300-based BCI in the auto-calibration routine. In this way, it could be possible to avoid trials without useful information. Likewise, it was also illustrated that the same system could detect other variables such as fatigue. The quantification of this cognitive state may be valuable for autonomous systems, which can perform special actions when the user has high levels of this cognitive state.

Chapter 10

Conclusion

The development of new assistive technology for people with severe neuromuscular problems has been the main goal of BCI research in the last decades. Scientists and engineers of several disciplines have collaborated in this field to provide an alternative communication channel to control computers and other devices by using the mind. The potential impact of this new technology in the daily life of partially or fully paralyzed patients has motivated the community to investigate new methods for translating the brain activity. This work represents another step towards the design of neural prostheses for communicating the user's intentions to smart devices.

EEG-based BCIs are an excellent choice for brain controlled applications regarding availability, price, and risk. However, the reduced amount of information provided by these systems and the possibility of false positives and classification errors do not allow a natural interaction between the user and the controlled device. Active steering is possible with this type of BCIs, however, the cognitive effort to solve the mental task and the time required to modulate the brain activity limit its long-term use.

The smart navigation offers an integral solution to many of the problems observed in brain-controlled wheelchairs. In the first case, this type of systems incorporates different ways to solve a navigation task. It is up to the users to choose a strategy to reach a final destination. Moreover, these systems are capable of correcting the navigation path and avoiding obstacles. The smart device increases the safety and comfort of the end-user. Lastly, an autonomous wheelchair can be integrated into a smart environment. Different devices can be synchronized with the actions performed by the wheelchair to provide other services.

The BCI proposed in this thesis was designed to operate an autonomous wheelchair in three different ways:

- *Full autonomous navigation.* The user only needs to select target places. The wheelchair controller solves the complete navigation task.
- *Semi-free navigation.* The wheelchair navigates according to the last direction specified by the user. Changing the direction at any time is possible.
- *Low-level navigation.* The user selects basic commands that move one or two meters the wheelchair or rotate the device.

These navigation modes were integrated into the same interface so that the user could be able to freely select the sequence of commands that move the wheelchair to the desired location. This

freedom is one of the most important features of the proposed system. As was discussed in Chapter 9, previous designs only consider one or two of these modalities. For instance, in some systems only target locations are available. In these cases, it is not possible to explore other places with the wheelchair. Likewise, BCIs for active steering only consider low-level interactions, which requires a mental effort to move the wheelchair a few meters. Even more, other BCIs do not consider basic functions such as rotations without translations. Therefore, the BCI described in this work provides the most complete set of navigation options so far.

In the hybrid interface, the main method for selecting target places and navigation commands are P300 evoked responses. This control paradigm was chosen because it makes possible to design an interface with tens or hundreds of options. The proposed BCI consists of nested menus that contain all the available functions. The fifteen subjects that tested the interface did not report any problem in controlling the wheelchair with this BCI. In just a few minutes, the participants that tested the real wheelchair understood how to move the device to specific places and how to adjust the wheelchair position. Additionally, in less than 40 minutes, the same participants realized how to explore freely outside the target positions and how to enter into other rooms. These results reveal that the selection times of the P300-based interface are not a problem for solving complex navigation tasks. It is more important to have a significant number of options in the interface, even if this means sacrificing speed. As long as the number of interactions between the user and the interface is not too big, users can move the wheelchair and thus, explore freely successfully.

The second component of the hybrid architecture is a SSVEP-based BCI, which is used to select the *"stop"* command and unlock the P300-based interface. This design adresses the most relevant problems of the P300-based interface: how to avoid false positives when the user is not controlling the interface, and how to stop the wheelchair quickly. The P300-based menu is locked after the selection of a target place. The only way to start again the flashing sequence is by selecting the *"start P300"* command with the SSVEP-based BCI. High-level and low-level commands do not unlock automatically the menu, but this option is always available in the interface.

This mechanism for locking and unlocking the P300-based BCI reduces the risk of false positives for the navigation commands. The user only interacts with the menu when required. However, this strategy only can work if the probability of false positives of the SSVEP-based BCI is low. To this end, in this work was proposed a novel method for detecting SSVEP-related activity. This new algorithm consists of analyzing clean data without SSVEP oscillations (between 30 and 60 seconds). Thresholds for significant increments of SSVEP-related activity are selected in such a way that a multi-trial classification approach controls the risk of false positives and classification errors. The online tests performed with this new method showed that no more than 0.024 false positives per minute are expected with this BCI. This means that the *"start P300"* command is selected accidentally once every 83.33 minutes, more than one hour, whereas the P300-based BCI may produce one false positive in less than five minutes. Therefore, the method for locking and unlocking the P300-based menu is highly efficient in reducing the false positives of the hybrid BCI.

The stopping method is another important improvement presented in this work. On average, 6.8 seconds were required to trigger the *"stop"* command. No more than 0.012 false positives per minute were detected, and the classification accuracy was 100%. In contrast, P300-based approaches for stopping the wheelchair are more prone to false positives. Motor imagery has also been tested for

this task, but needs longer training sessions and is harder to master. With this technique, the user needs to concentrate on a mental task to modulate the brain oscillations, which may be risky in case of an emergency. On the contrary, the modulation of SSVEP-related activity is easier, and the proposed method only requires 60 seconds of training. Finally, some recent developments have incorporated the detection of blink patterns as a mechanism for stopping the wheelchair. It has been reported selection times of 4.5 seconds and accuracies between 85% and 90% with this technique. None of these studies have tested this method in long experimental sessions to determine how easy is to trigger the *"stop"* command involuntarily. In contrasts, the proposed method is slower than the approach based on blinks but has higher accuracies. Users always obtain what they want, so that it can be said that the implemented SSVEP-based BCI is very competitive for stopping the wheelchair.

The high accuracy of the P300-based interface was obtained by implementing two strategies in the processing stages of this BCI. Firstly, it was proposed a new adaptive method for determining the optimal number of flashes to select one option of the interface. This algorithm estimates the probability of classifying one non-target option incorrectly after multiple epochs. It only requires the accuracy of the single-trial classifier to control the error rate below the desired level. It was demonstrated in this work that this new algorithm reduces the risk of classification errors without sacrificing the interface speed.

The second strategy consisted of adapting the P300 classifier using newly processed data. The results of the classification stages are used to modify the machine learning model in the online run. In this way, slow changes in the neural activity do not alter the overall classifier performance. This auto-adaptation strategy and a novel method proposed for auto-calibrating the P300-based BCI ensure an optimal performance and a complete autonomy of this system. Users do not need to run a calibration routine to use the interface, while in case of a drop in performance, the system is capable of using new data to estimate a new calibration model.

A passive BCI for the automatic detection of machine errors was also evaluated in this work. This subsystem detects error-related potentials elicited when the user perceives classification errors. Theoretically, this direct feedback could boost the system performance by correcting mistakes without user's intervention. Several researchers recommend to include this kind of BCIs because some P300-based spellers have achieved greater information transfer rates with this automatic correction. However, the results presented in this work reveal that the hybrid BCI does not require this passive subsystem because of the high accuracy of the P300-based BCI. False positives of the error detector cancel correct selections, whereas the probability of observing a mistake corrected by the ErrP-based BCI is low. Users prefer not to use this passive BCI in online conditions.

The last component of the hybrid architecture is another passive BCI that quantifies cognitive states. This information may be used to improve the understanding of how the users react when they interact with the wheelchair and other elements of the environment. Smart systems can customize the user's experience and modify parts of the interface according to the subject's states. In this thesis, two scenarios were described to exemplify the potential use of this technology. Firstly, task engagement was measured in the training sessions and online experiments. In this case, it was shown that for some participants this variable increases during the counting task of the P300-based BCI. This means that this cognitive state can be used to recognize when the user is performing the flashing counting. As discussed along the chapters of this work, this information is useful to avoid false positives and identify

a training set in the proposed auto-calibration method for P300-based BCIs.

In the second scenario, fatigue was measured in one subject at two different times (2 p.m. and 4 a.m.). In this extreme case, it was shown that the cognitive score was much greater at 4 a.m., just in the typical sleep time of the participant. In both cases, the user was conscious using the interface. This experiment shows that the detection of this variable is possible with the proposed system. The quantification of fatigue is relevant for assistive technology because it makes possible to take actions to ensure safety and comfort.

This thesis also presented three alternatives to the hybrid BCI system for operating the wheelchair. One of this interfaces is controlled only with P300 evoked responses. In this case, the BCI is always unlocked, and the stop command was incorporated into the options of the interactive menu. All the participants could operate the wheelchair without problems, but false positives were observed because of the lack of a mechanism for locking the P300-based menu. Only one participant preferred this operation mode over the hybrid interface for indoor navigation. The rest indicated that the hybrid strategy is more satisfying than the P300-based controller.

The second interface is another P300-based BCI designed to operate the wheelchair outdoors. Currently, the autonomous wheelchair only can navigate to target places in open spaces, so that it requires another GUI to indicate the possible options. The participants that tested this interface did not struggle to select target places and stop the wheelchair, even though the lighting conditions are variable along the navigation paths. As long as the users can distinguish the flashing options, the ERP components can be classified correctly. Probably, this is the first time that a P300-based BCI has been used to operate a wheelchair outdoors.

Finally, a SSVEP-based BCI for active steering was also implemented. With this interface, users can rotate and move forward the wheelchair by looking at one of the flashing stimuli. At the same time, the navigation system avoids obstacles and identifies the location of possible entrances for other places. In the online tests, the participants were able to move the wheelchair and enter into other rooms without problems. These results demonstrate that this modality is also possible with the proposed BCI.

In conclusion, the objective of developing a new hybrid BCI for controlling autonomous devices was achieved. The proposed system not only is highly competitive compared to other brain-controlled wheelchairs but also incorporates the most complete navigation tools to date. This hybrid BCI is highly reliable, and users can operate the interface after a few minutes of training.

10.1 Main Contributions of this Work

The following subsections summarize the most important contributions of this work.

10.1.1 Development and Testing of a New Brain-Computer Interface for Autonomous Systems

The main objective of this project was the development of a multi-paradigm brain-computer interface for controlling autonomous systems. To this end, it was proposed a new BCI that combines P300-evoked potentials and SSVEPs to select target places and navigation commands. In this thesis, all the components of this BCI were tested in offline and online conditions to identify the optimal configuration of each subsystem. Finally, the online tests of the hybrid architecture were presented in

Chapter 9. The results of these tests demonstrated that users could operate the wheelchair with the proposed system indoors and outdoors.

10.1.2 Development of a New Hybrid BCI Paradigm for Autonomous Wheelchairs

The two main components of the hybrid architecture are two BCIs that work in parallel to control different options of the interface. On the one hand, P300 evoked responses are used to select target places, high-level commands for semi-free navigation, and low-level commands for discrete rotations and translations. On the other hand, a SSVEP-based BCI helps to choose the *"stop"* command and unlock the P300-based menu.

As discussed in Chapter 9, the proposed design addresses two problems observed in wheelchairs controlled by P300-based BCIs: how to stop the autonomous wheelchair quickly, and how to avoid false alarms when the user is not controlling the P300-based interface. However, this method is reliable only if the SSVEP-based BCI has low probabilities of false positives. For this reason, it was necessary to develop a new method for classifying SSVEP-related activity that allows a strict control of false positives and classification errors.

10.1.3 Integral Evaluation of Preprocessing and Feature Extraction Methods for ERP- and SSVEP-Based BCIs

This thesis includes an integral evaluation of all the processing stages of the hybrid BCI. Some tests had the purpose of corroborating the results reported in other brain-computer interfaces. This is the case of the experiments presented in Chapters 5 and 7 to evaluate the classification and feature selection methods for ERP- and SSVEP-Based BCIs. In both cases, linear models combined with stepwise regression for feature selection are a good choice to discriminate ERP components and SSVEP-related activity. This model selection has also been recommended in several BCI studies.

Apart from that, some offline experiments were conducted to test some combination of preprocessing and feature extraction methods that have not been reported in other works. For instance, the artifact subspace reconstruction method has never been tested in ERP- and SSVEP-based BCIs. In this thesis was evaluated the classifier performance of the implemented BCIs after filtering out high amplitude artifacts with the ASR method.

Moreover, the common average reference and the surface Laplacian are not usually combined with spatial filters because other researchers consider that both methods are another kind of feature extraction techniques, which is an error. The purpose of both approaches is to estimate a new set of signals with some special properties so that both methods can be combined with any other feature extraction algorithm. In Chapters 5 and 7 was evaluated the effect of including both techniques in the preprocessing stage before applying the feature extraction algorithms.

Finally, other important tests presented in this work revealed the minimum number of features needed to achieve an optimal accuracy, and the minimum training set size required to find discriminative information in ERP data. This information is essential to design similar systems, including the training routines.

10.1.4 A New Adaptive Method for Controlling the Error Rate in P300-Based BCIs

This thesis presented a new adaptive method for finding the optimal number of flashes to detect ERP responses in P300-based BCIs. Single-trial classification is not feasible in P300-based interfaces because accuracies are between 80% and 95%. Multi-trial classification is used in this kind of systems to reduce the risk of misclassification.

The proposed method only requires an estimation of the accuracy of the single-trial classifier to calculate the probability of classifying one option correctly after multiple flashes. As explained in Chapter 5, in online conditions, the number of flashes to analyze varies according to the subject's performance. In the end, the system performs one selection only when the probability of error is below a predefined threshold.

10.1.5 A New Algorithm for Auto-Calibrating P300-Based BCIs

Another significant contribution of this work is the development of a new method for auto-calibrating P300-based BCIs. This method finds observations of the target and the non-target classes without user's intervention. This method, combined with the self-adaptation mechanism implemented in the classification stage, allows the complete autonomy of the BCI by reducing the external intervention for training or adjusting the interface.

The BCI uses this algorithm in two different scenarios: when the system has not been trained with calibration data, and when the interface struggles to classify ERP data in the online run. In the last case, the current classification model is not valid to discriminate P300 responses, so that the auto-adaptation mechanism is not enough to improve the classifier.

The results presented in Chapter 6 demonstrate that this new method finds a new training set without the need of prior information about the users. Additionally, after training the P300-based BCI, the self-adaptation procedure ensures the optimal performance in the online run. The combination of both strategies is necessary to improve the autonomy of the BCI for long-term use.

10.1.6 Development of a Novel Classification Method for SSVEP-Related Activity

A central component of the hybrid architecture is the mechanism for locking and unlocking the P300-based BCI. This function reduces the risk of executing inadvertent commands whose cancellation may take several seconds. In the proposed interface, a SSVEP-based BCI is used to start the flashing sequence of the P300-based interface, whereas the P300-based menu is locked by selecting target places or by choosing this option from the menu. In this way, the SSVEP-based BCI is responsible for controlling the false positive rate of the interface.

In this thesis was presented a novel method to detect SSVEP-related activity. This approach uses data of the null condition (non-SSVEP-related activity) to determine the level of significance of the oscillatory activity elicited by flickering stimuli. Thresholds are selected directly in the null distributions so that it is relatively easy to infer the probability of selecting incorrectly one option.

In Chapter 7 was demonstrated that this method is just as effective as the classical approach based on binary classifiers regarding selection times and classification errors, even though it only requires data of the null condition. Finally, in Chapter 9, it is shown that this new proposal is highly efficient in controlling false positives.

10.1.7 Implementation and Testing of a New Scoring System for Quantifying Cognitive States

In Chapter 8 was described the BCI implemented to measure cognitive states in the hybrid architecture. This subsystem incorporates a scoring system which indicates the instantaneous and cumulative user's performance for one particular state. Both variables are useful in neurofeedback applications, but also can be used to identify events in the timeline that produce changes in the brain activity.

This BCI has been tested in neurofeedback experiments and in the hybrid system developed to control the wheelchair. In both cases, it has been demonstrated that the proposed scoring system represents the instantaneous and long-term variations of a cognitive state effectively, and at the same time, provides all the information required by the system to interpret the user's conditions.

10.2 Open Questions

Although the hybrid BCI is highly efficient in communicating the user's intentions to the autonomous device, there are still many opportunities for improving the proposed design. For instance, selections times for some of the low-level and high-level commands could be improved by reorganizing the interactive menu. In the current interface, all the low-level commands are included in the same section, so that there are many options in the interface when the user operates the wheelchair in this mode. However, the participants only were interested in selecting a few options from all the low-level commands. This fact may indicate that the most important commands could be presented in another section with fewer choices, which reduces the selection times significantly. In this direction, it would be interesting to investigate the satisfaction levels for different GUI designs. Another possibility would be to provide different customization levels for each user to speed up the interface and increase the comfort levels.

Another component that may be improved is the adaptation routine of the P300-based BCI. So far, only the classification model is adjusted in the online run. However, the stimulus presentation and the speed of the flashing sequence can also be adjusted according to the user's attention levels. In the same direction, The flickering frequencies of the SSVEP-based BCI could also be adapted to improve the recognition task. In this case, if one flickering frequency is close to the alpha frequency of one subject, and suddenly the user starts to modulate this kind of activity, it would be better to change the flashing frequency to another one to reduce the probability of false positives.

This thesis also described some opportunities for the detection of cognitive states in control applications. So far, it was proved that the proposed system could detect variables such as fatigue and task engagement. However, is still open how to use this information to improve the interaction between users and machines. Perhaps, the incorporation of fatigue levels into the smart system is straightforward, but many other interesting cognitive states can also be included in the hybrid system.

These new ideas show that there is still a long way ahead. The development of reliable BCI systems for long-term use is still an open question, but the latest discoveries in this research field encourage the community to continue investigating how to decode the human mind and how to use this information to control devices. The outstanding results presented in this document are another step in this direction.

Bibliography

Åkesson, B. M., Jørgensen, J. B., Poulsen, N. K., & Jørgensen, S. B. (2007). A tool for kalman filter tuning. In *17th european symposium on computer aided process engineering - ESCAPE17, 2007*.

Allison, B. Z., Leeb, R., Brunner, C., Müller-Putz, G. R., Bauernfeind, G., Kelly, J. W., & Neuper, C. (2012). Toward smarter BCIs: Extending BCIs through hybridization and intelligent control. *Journal of Neural Engineering, 9*(1), 013001.

Bashivan, P., Rish, I., & Heisig, S. (2015). Mental state recognition via wearable EEG. In *Proceedings of 5th NIPS workshop on machine learning and interpretation in neuroimaging*.

Baudat, G. & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Comput. 12*(10), 2385–2404.

Benjamini, Y., Krieger, A. M., & Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika, 93*(3), 491.

Birbaumer, N. (1999). Slow cortical potentials: Plasticity, operant control, and behavioral effects. *The Neuroscientist, 5*(2), 74–78.

Birbaumer, N. (2006). Breaking the silence: Brain–computer interfaces (BCI) for communication and motor control. *Psychophysiology, 43*(6), 517–532.

Bishop, C. M. (2006). Linear models for classification. In *Pattern recognition and machine learning (information science and statistics)* (pp. 179–224). Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Boukerroui, D., Noble, J. A., & Brady, M. (2004). On the choice of band-pass quadrature filters. *Journal of Mathematical Imaging and Vision, 21*(1), 53–80.

Bowman, A. & Azzalini, A. (1997). *Applied smoothing techniques for data analysis: The kernel approach with s-plus illustrations*. Oxford statistical science series. Oxford: Clarendon Press.

Brumberg, J. S., Nieto-Castanon, A., Kennedy, P. R., & Guenther, F. H. (2010). Brain-computer interfaces for speech communication. *Speech Commun. 52*(4), 367–379.

Buch, E., Weber, C., Cohen, L. G., Braun, C., Dimyan, M. A., Ard, T., ... Birbaumer, N. (2008). Think to Move: A Neuromagnetic Brain-Computer Interface (BCI) System for Chronic Stroke. *Stroke; a Journal of Cerebral Circulation, 39*(3), 910–917.

Cao, L., Li, J., Ji, H., & Jiang, C. (2014). A hybrid brain computer interface system based on the neurophysiological protocol and brain-actuated switch for wheelchair control. *Journal of Neuroscience Methods, 229*, 33–43.

Carlson, T. & del R. Millan, J. (2013). Brain-controlled wheelchairs: A robotic architecture. *IEEE Robotics Automation Magazine, 20*(1), 65–73.

Carvalhaes, C. G. & Acacio de Barros, J. (2015). The surface laplacian technique in EEG: Theory and methods. *International Journal of Psychophysiology*, *97*(3), 174–188. On the benefits of using surface Laplacian (current source density) methodology in electrophysiology.

Carvalhaes, C. G. & Suppes, P. (2011). A spline framework for estimating the EEG surface laplacian using the euclidean metric. *Neural Computation*, *23*(11), 2974–3000.

Carvalho, S. N., Costa, T. B. S., Uribe, L. F. S., Soriano, D. C., Yared, G. F. G., Coradine, L. C., & Attux, R. (2015). Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs. *Biomed. Signal Proc. and Control*, *21*, 34–42.

Chen, L., Jin, J., Zhang, Y., Wang, X., & Cichocki, A. (2015). A survey of the dummy face and human face stimuli used in {bci} paradigm. *Journal of Neuroscience Methods*, *239*, 18–27.

Chen, X., Chen, Z., Gao, S., & Gao, X. (2014). A high-itr ssvep-based bci speller. *Brain-Computer Interfaces*, *1*(3-4), 181–191.

Cincotti, F., Mattia, D., Aloise, F., Bufalari, S., Schalk, G., Oriolo, G., ... Babiloni, F. (2008). Non-invasive brain–computer interface system: Towards its application as assistive technology. *Brain Research Bulletin*, *75*(6), 796–803. Special Issue: Robotics and Neuroscience.

Cohen, M. K. (2014a). EEG artifacts: Their detection, influence and removal. In *Analyzing neural time series data: Theory and practice* (pp. 87–96). London: The MIT Press.

Cohen, M. K. (2014b). Preprocessing steps necessary and useful for advanced data analysis. In *Analyzing neural time series data: Theory and practice* (pp. 73–86). London: The MIT Press.

Craig, A., Tran, Y., Wijesuriya, N., & Nguyen, H. (2012). Regional brain wave activity changes associated with fatigue. *Psychophysiology*, *49*(4), 574–582.

Dien, J. (1998). Issues in the application of the average reference: Review, critiques, and recommendations. *Behavior Research Methods, Instruments, & Computers*, *30*(1), 34–43.

Dornhege, G., Blankertz, B., Krauledat, M., Losch, F., Curio, G., & Muller, K. R. (2006). Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Transactions on Biomedical Engineering*, *53*(11), 2274–2281.

Doud, A. J., Lucas, J. P., Pisansky, M. T., & He, B. (2011). Continuous three-dimensional control of a virtual helicopter using a motor imagery based brain-computer interface. *PLoS ONE*, *6*(10), 1–10.

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in sobolev spaces. In W. Schempp & K. Zeller (Eds.), *Constructive theory of functions of several variables: Proceedings of a conference held at Oberwolfach april 25 – may 1, 1976* (pp. 85–100). Berlin, Heidelberg: Springer Berlin Heidelberg.

Farwell, L. A. & Donchin, E. (1988). Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*, 510–523.

Fazel-Rezai, R. & Abhari, K. (2009). A region-based p300 speller for brain-computer interface. *Canadian Journal of Electrical and Computer Engineering*, *34*(3), 81–85.

Fazli, S., Danóczy, M., Popescu, F., Blankertz, B., & Müller, K.-R. (2010). Using rest class and control paradigms for brain computer interfacing. In S. D. Tan & A. Nijholt (Eds.), *Brain-computer interfaces: Applying our minds to human-computer interaction* (pp. 55–70). London: Springer London.

Fitzgibbon, S. P., Powers, D. M. W., Pope, K. J., & Clark, C. R. (2007). Removal of EEG noise and artifact using blind source separation. *Journal of Clinical Neurophysiology*, *24*(3).

Friedrich, E. V. C., Suttie, N., Sivanathan, A., Lim, T., Louchart, S., & Pineda, J. A. (2014). Brain-computer interface game applications for combined neurofeedback and biofeedback treatment for children on the autism spectrum. *Frontiers in Neuroengineering*, *7*(21).

Friman, O., Volosyak, I., & Graser, A. (2007). Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, *54*(4), 742–750.

Furdea, A., Halder, S., Krusienski, D., Bross, D., Nijboer, F., Birbaumer, N., & Kübler, A. (2009). An auditory oddball (p300) spelling system for brain-computer interfaces. *Psychophysiology*, *46*(3), 617–625.

G. Brys, A. S., M. Hubert. (2004). A robust measure of skewness. *Journal of Computational and Graphical Statistics*, *13*(4), 996–1017.

Gabor, D. (1946). Theory of Communication. *J. IEE*, *93*(26), 429–457.

Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J., & Millán, J. d. R. (2008). A brain-actuated wheelchair: Asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, *119*(9), 2159–2169.

George, L. & Lécuyer, A. (2010). An Overview of Research on "Passive" Brain-Computer Interfaces for Implicit Human-Computer Interaction. In *International conference on applied bionics and biomechanics ICABB 2010 - workshop w1 "brain-computer interfacing and virtual reality"*. Venise, Italy.

Ghani, A., Mendoza-Montoya, O., Mclntosh, A. R., & Ritter, P. (2016). My virtual dream: Brain computer interface in an immersive art environment. In *Proceedings of the 6th international brain-computer interface meeting, organized by the BCI society*. Graz, Austria.

Gibbons, J. D. & Chakraborti, S. (2011). Nonparametric statistical inference. In M. Lovric (Ed.), *International encyclopedia of statistical science* (pp. 977–979). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gollee, H., Volosyak, I., McLachlan, A. J., Hunt, K. J., & Gräser, A. (2010). An SSVEP-based brain-computer interface for the control of functional electrical stimulation. *IEEE Transactions on Biomedical Engineering*, *57*(8), 1847–1855.

Good, P. I. (2006). *Resampling methods: A practical guide to data analysis*. Boston: Birkhäuser.

Guerrero, J. A., Marroquin, J. L., Rivera, M., & Quiroga, J. A. (2005). Adaptive monogenic filtering and normalization of espi fringe patterns. *Opt. Lett. 30*(22), 3018–3020.

Hardoon, D. R., Szedmak, S., & Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, *16*(12), 2639–2664.

Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. (2009). Model assessment and selection. In *The elements of statistical learning: Data mining, inference, and prediction* (pp. 219–260). Springer series in statistics. Autres impressions : 2011 (corr.), 2013 (7e corr.) New York: Springer.

Hinterberger, T., Schmidt, S., Neumann, N., Mellinger, J., Blankertz, B., Curio, G., & Birbaumer, N. (2004). Brain-computer communication and slow cortical potentials. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1011–1018.

Horki, P., Neuper, C., Pfurtscheller, G., & Müller-Putz, G. (2010). Asynchronous Steady-State Visual Evoked Potential Based BCI Control of a 2-DoF Artificial Upper Limb. *Biomedizinische Technik. Biomedical engineering*, *55*(6), 367–374.

Hubert, M. & Vandervieren, E. (2008). An adjusted boxplot for skewed distributions. *Computational Statistics and Data Analysis*, *52*(12), 5186–5201.

Hwang, H.-J., Kwon, K., & Im, C.-H. (2009). Neurofeedback-based motor imagery training for brain–computer interface (BCI). *Journal of Neuroscience Methods*, *179*(1), 150–156.

Ille, N., Berg, P., & Scherg, M. (2002). Artifact correction of the ongoing EEG using spatial filters based on artifact and brain signal topographies. *Journal of Clinical Neurophysiology*, *19*(2).

Iturrate, I., Antelis, J. M., Kubler, A., & Minguez, J. (2009). A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation. *IEEE Transactions on Robotics*, *25*(3), 614–627.

Iturrate, I., Antelis, J., & Minguez, J. (2009). Synchronous EEG brain-actuated wheelchair with automated navigation. In *Robotics and automation, 2009. ICRA '09. IEEE international conference on* (pp. 2318–2325).

James, C. J. [C. J.] & Gibson, O. J. (2003). Temporally constrained ica: An application to artifact rejection in electromagnetic brain signal analysis. *IEEE Transactions on Biomedical Engineering*, *50*(9), 1108–1116.

James, C. J. [Christopher J] & Hesse, C. W. (2005). Independent component analysis for biomedical signals. *Physiological Measurement*, *26*(1), R15.

Jones, M. C. (1993). Simple boundary correction for kernel density estimation. *Statistics and Computing*, *3*(3), 135–146.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In *Information and communication technology, electronics and microelectronics (MIPRO), 2015 38th international convention on* (pp. 1200–1205).

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)), 35–45.

Käthner, I., Kübler, A., & Halder, S. (2015). Rapid p300 brain-computer interface communication with a head-mounted display. *Frontiers in Neuroscience*, *9*, 207.

Khalid, M. B., Rao, N. I., Rizwan-i-Haque, I., Munir, S., & Tahir, F. (2009). Towards a brain computer interface using wavelet transform with averaged and time segmented adapted wavelets. In *Computer, control and communication, 2009. IC4 2009. 2nd international conference on* (pp. 1–4).

Kovacevic, N., Ritter, P., Tays, W., Moreno, S., & McIntosh, A. R. (2015). 'my virtual dream': Collective neurofeedback in an immersive art environment. *PLoS ONE*, *10*(7), e0130129.

Kreilinger, A., Kaiser, V., Breitwieser, C., Williamson, J., Neuper, C., & Müller-Putz, G. (2012). Switching between manual control and brain-computer interface using long term and short term quality measures. *Frontiers in Neuroscience*, *5*, 147.

Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2006). A comparison of classification techniques for the p300 speller. *Journal of Neural Engineering*, *3*(4), 299.

Kumari, B. & Swarnkar, T. (2011). Filter versus wrapper feature subset selection in large dimensionality micro array: A review. *International Journal of Computer Science and Information Technologies*, *2*(3), 1048–1053.

Le, J., Menon, V., & Gevins, A. (1994). Local estimate of surface laplacian derivation on a realistically shaped scalp surface and its performance on noisy data. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, *92*(5), 433–441.

Ledoit, O. & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, *88*(2), 365–411.

Lemm, S., Blankertz, B., Curio, G., & Muller, K. R. (2005). Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Transactions on Biomedical Engineering*, *52*(9), 1541–1548.

Lenhardt, A., Kaper, M., & Ritter, H. J. (2008). An adaptive p300-based online brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *16*(2), 121–130.

Leuthardt, E. C. [Eric C], Gaona, C., Sharma, M., Szrama, N., Roland, J., Freudenberg, Z., … Schalk, G. (2011). Using the electrocorticographic speech network to control a brain–computer interface in humans. *Journal of Neural Engineering*, *8*(3), 036004.

Lin, J.-S. & Shieh, C.-H. (2014). An ssvep-based bci system and its applications. *International Journal of Advanced Computer Science and Applications(IJACSA)*, *5*(10).

Linden, D. E. (2005). The P300: Where in the Brain Is It Produced and What Does It Tell Us? *The Neuroscientist : A Review Journal Bringing Neurobiology, Neurology and Psychiatry*, *11*(6), 563–576.

Llarena, A. & Rojas, R. (2016). I am alleine, the autonomous wheelchair at your service. In E. Menegatti, N. Michael, K. Berns, & H. Yamaguchi (Eds.), *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13* (pp. 1613–1626). Cham: Springer International Publishing.

Long, J., Li, Y., Wang, H., Yu, T., Pan, J., & Li, F. (2012). A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *20*(5), 720–729.

Lotte, F. [F.] & Guan, C. [C.]. (2011). Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms. *IEEE Transactions on Biomedical Engineering*, *58*(2), 355–362.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *4*(2), R1.

Lotte, F. [Fabien]. (2014). A tutorial on EEG signal-processing techniques for mental-state recognition in brain–computer interfaces. In R. E. Miranda & J. Castet (Eds.), *Guide to brain-computer music interfacing* (pp. 133–161). London: Springer London.

Lotte, F. [Fabien] & Guan, C. [Cuntai]. (2009). An Efficient P300-Based Brain-Computer Interface with Minimal Calibration Time. In *Assistive machine learning for people with disabilities symposium (NIPS'09 symposium)*. Vancouver, Canada.

Makeig, S., Bell, A. J., Jung, T.-P., & Sejnowski, T. J. (1996). Independent component analysis of electroencephalographic data. In D. S. Touretzky & M. E. Hasselmo (Eds.), *Advances in neural information processing systems 8* (pp. 145–151). MIT Press.

Manyakov, N. V., Chumerin, N., Combaz, A., & Van Hulle, M. M. (2011). Comparison of classification methods for p300 brain-computer interface on disabled subjects. *Intell. Neuroscience*, *2011*, 2:1–2:12.

Mardia, K. V. & Jupp, P. E. (2000). Tests of uniformity and tests of goodness-of-fit. In *Directional statistics* (pp. 93–118). Chichester: Wiley.

Marroquin, J. L., Harmony, T., Rodriguez, V., & Valdes, P. (2004). Exploratory EEG data analysis for psychophysiological experiments. *NeuroImage*, *21*(3), 991–999.

Mason, S. G. & Birch, G. E. (2003). A general framework for brain-computer interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *11*(1), 70–85.

Millan, J. R., Renkens, F., Mourino, J., & Gerstner, W. (2004). Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1026–1033.

Moore Jackson, M. & Mappus, R. (2010a). Applications for brain-computer interfaces. In S. D. Tan & A. Nijholt (Eds.), *Brain-computer interfaces: Applying our minds to human-computer interaction* (pp. 89–103). London: Springer London.

Moore Jackson, M. & Mappus, R. (2010b). Neural control interfaces. In S. D. Tan & A. Nijholt (Eds.), *Brain-computer interfaces: Applying our minds to human-computer interaction* (pp. 21–33). London: Springer London.

Mullen, T., Kothe, C., Chi, Y. M., Ojeda, A., Kerth, T., Makeig, S., … Jung, T.-P. (2013). Real-time modeling and 3d visualization of source dynamics and connectivity using wearable EEG. *Conf Proc IEEE Eng Med Biol Soc*, *2013*, 2184–2187. 24110155[pmid].

Müller-Putz, G. R., Scherer, R., Pfurtscheller, G., & Rupp, R. (2005). EEG-based neuroprosthesis control: A step towards clinical practice. *Neuroscience Letters*, *382*(1–2), 169–174.

Murguialday, A. R., Aggarwal, V., Chatterjee, A., Cho, Y., Rasmussen, R., O'Rourke, B., … Thakor, N. V. (2007). Brain-computer interface for a prosthetic hand using local machine control and haptic feedback. In *2007 IEEE 10th international conference on rehabilitation robotics* (pp. 609–613).

Neumaier, A. & Schneider, T. (2001). Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans. Math. Softw. 27*(1), 27–57.

Neuper, C., Scherer, R., Reiner, M., & Pfurtscheller, G. (2005). Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial {EEG}. *Cognitive Brain Research*, *25*(3), 668–677.

Ng, A. Y. & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems 14* (pp. 841–848). MIT Press.

Nicolas-Alonso, L. F. & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, *12*(2), 1211.

Noether, G. E. (1992). Introduction to wilcoxon (1945) individual comparisons by ranking methods. In S. Kotz & N. L. Johnson (Eds.), *Breakthroughs in statistics: Methodology and distribution* (pp. 191–195). New York, NY: Springer New York.

O'She, R. P., Roeber, U., & Bach, M. (2009). Evoked potential: Vision. In E. B. Goldstein (Ed.), *Encyclopedia of perception* (pp. 399–400). Arizona: Sage Publications, Inc.

Odom, J. V., Bach, M., Brigell, M., Holder, G. E., McCulloch, D. L., Tormene, A. P., & Vaegan. (2010). Iscev standard for clinical visual evoked potentials (2009 update). *Documenta Ophthalmologica*, *120*(1), 111–119.

Ortner, R., Allison, B. Z., Korisek, G., Gaggl, H., & Pfurtscheller, G. (2011). An SSVEP BCI to control a hand orthosis for persons with tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(1), 1–5.

Panicker, R. C., Puthusserypady, S., & Sun, Y. (2010). Adaptation in p300 brain-computer interfaces: A two-classifier cotraining approach. *IEEE Transactions on Biomedical Engineering*, *57*(12), 2927–2935.

Park, C., Koo, J.-Y., Kim, P. T., & Lee, J. W. (2008). Stepwise feature selection using generalized logistic loss. *Computational Statistics and Data Analysis*, *52*(7), 3709–3718.

Pfurtscheller, G. [G.] & Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, *110*(11), 1842–1857.

Pfurtscheller, G., Guger, C., Müller, G., Krausz, G., & Neuper, C. (2000). Brain oscillations control hand orthosis in a tetraplegic. *Neuroscience Letters*, *292*(3), 211–214.

Pfurtscheller, G. [Gert], Müller, G. R., Pfurtscheller, J., Gerner, H. J., & Rupp, R. (2003). 'thought' – control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. *Neuroscience Letters*, *351*(1), 33–36.

Philips, J., del R. Millan, J., Vanacker, G., Lew, E., Galan, F., Ferrez, P. W., … Nuttin, M. (2007). Adaptive shared control of a brain-actuated simulated wheelchair. In *2007 IEEE 10th international conference on rehabilitation robotics* (pp. 408–414).

Picton, T. W. (1992). The p300 wave of the human event-related potential. *Journal of Clinical Neurophysiology*, *9*(4), 456–479.

Podder, P., Mehedi Hasan, M., Rafiqul Islam, M., & Sayeed, M. (2014). Design and Implementation of Butterworth, Chebyshev-I and Elliptic Filter for Speech Signal Analysis. *International Journal of Computer Applications*, *98*(7), 12–18.

Polich, J. (2007). Updating P300: An Integrative Theory of P3a and P3b. *Clinical Neurophysiology*, *118*(10), 2128–2148.

Polich, J., Ellerson, P. C., & Cohen, J. (1996). P300, stimulus intensity, modality, and probability. *International Journal of Psychophysiology*, *23*(1–2), 55–62.

Politis, D. N., Romano, J. P., & Wolf, M. (1999). Subsampling in the i.i.d. case. In *Subsampling* (pp. 39–64). New York, NY: Springer New York.

Priftis, K. (2014). Effectiveness of the p3-speller in brain-computer interfaces for amyotrophic lateral sclerosis patients: A systematic review and meta-analysis. *Frontiers in Neuroengineering*, *7*(12).

Puanhvuan, D. [D.] & Wongsawat, Y. (2012). Semi-automatic p300-based brain-controlled wheelchair. In *Complex medical engineering (CME), 2012 ICME international conference on* (pp. 455–460).

Puanhvuan, D. [Dilok], Khemmachotikun, S., Wechakarn, P., Wijarn, B., & Wongsawat, Y. (2017). Navigation-synchronized multimodal control wheelchair from brain to alternative assistive technologies for persons with severe disabilities. *Cognitive Neurodynamics*, *11*(2), 117–134.

Rebsamen, B., Burdet, E., Guan, C., Teo, C. L., Zeng, Q., Ang, M., & Laugier, C. (2007). Controlling a wheelchair using a BCI with low information transfer rate. In *2007 IEEE 10th international conference on rehabilitation robotics* (pp. 1003–1008).

Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C. L., Zeng, Q., … Laugier, C. (2006). 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.* (pp. 1101–1106).

Rebsamen, B., Guan, C., Zhang, H., Wang, C., Teo, C., Ang, M. H., & Burdet, E. (2010). A brain controlled wheelchair to navigate in familiar environments. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 18*(6), 590–598.

Rivet, B., Souloumiac, A., Attina, V., & Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: Aplication to brain-computer interface. *IEEE Transactions on Biomedical Engineering, 56*(8), 2035–2043.

Roberts, S. & Stals, L. (2004). Discrete thin plate spline smoothing in 3d. In J. Crawford & A. J. Roberts (Eds.), *Proc. of 11th computational techniques and applications conference CTAC-2003* (Vol. 45, pp. C646–C659).

Rupp, R. (2014). Challenges in clinical applications of brain computer interfaces in individuals with spinal cord injury. *Frontiers in Neuroengineering, 7*, 38.

Saa, J. F. D., de Pesters, A., McFarland, D., & Çetin, M. (2015). Word-level language modeling for p300 spellers based on discriminative graphical models. *Journal of Neural Engineering, 12*(2), 026007.

Schalk, G. & Leuthardt, E. C. [E. C.]. (2011). Brain-computer interfaces using electrocorticographic signals. *IEEE Reviews in Biomedical Engineering, 4*, 140–154.

Scheffers, M. K. & Coles, M. G. H. (2000). Performance monitoring in a confusing world: Error-related brain activity, judgments of response accuracy, and types of errors. *Journal of Experimental Psychology: Human Perception and Performance, 26*(1), 141–151.

Schettini, F., Aloise, F., Aricò, P., Salinari, S., Mattia, D., & Cincotti, F. (2014). Self-calibration algorithm in an asynchronous p300-based brain–computer interface. *Journal of Neural Engineering, 11*(3), 035004.

Schmidt, N. M., Blankertz, B., & Treder, M. S. (2012). Online detection of error-related potentials boosts the performance of mental typewriters. *BMC Neuroscience, 13*(1), 19.

Seno, B. D., Matteucci, M., & Mainardi, L. (2008). A genetic algorithm for automatic feature extraction in p300 detection. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 3145–3152).

Silvoni, S., Volpato, C., Cavinato, M., Marchetti, M., Priftis, K., Merico, A., … Piccione, F. (2009). P300-based brain-computer interface communication: Evaluation and follow-up in amyotrophic lateral sclerosis. *Frontiers in Neuroscience, 3*(1).

Sirvent, J. L., Azorín, J. M., Iáñez, E., Úbeda, A., & Fernández, E. (2010). P300-based brain-computer interface for internet browsing. In Y. Demazeau, F. Dignum, J. M. Corchado, J. Bajo, R. Corchuelo, E. Corchado, …A. Campbell (Eds.), *Trends in practical applications of agents and multiagent systems: 8th international conference on practical applications of agents and multiagent systems* (pp. 615–622). Berlin, Heidelberg: Springer Berlin Heidelberg.

Spüler, M. [M.], Walter, A., Rosenstiel, W., & Bogdan, M. (2014). Spatial filtering based on canonical correlation analysis for classification of evoked or event-related potentials in EEG data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22*(6), 1097–1103.

Spüler, M. [Martin], Rosenstiel, W., & Bogdan, M. (2012). Online adaptation of a c-vep brain-computer interface(bci) based on error-related potentials and unsupervised learning. *PLOS ONE*, *7*(12), 1–11.

Stone, J. V. (2004). Strategies for blind source separation. In *Independent component analysis: A turorial introduction* (pp. 13–18). London: The MIT Press.

Takano, K., Hata, N., & Kansaku, K. (2011). Towards intelligent environments: An augmented reality–brain–machine interface operated with a see-through head-mount display. *Front Neurosci*, *5*, 60. 21541307[pmid].

Tanaka, K., Matsunaga, K., & Wang, H. O. (2005). Electroencephalogram-based control of an electric wheelchair. *IEEE Transactions on Robotics*, *21*(4), 762–766.

Thede, L. (2004a). Analog filter approximation functions. In *Practical analog and digital filter design* (pp. 15–54). Norwood: Artech House.

Thede, L. (2004b). Finite impulse response digital filter design. In *Practical analog and digital filter design* (pp. 161–186). Norwood: Artech House.

Tomioka, R., Dornhege, G., Norte, G., Blankertz, B., Aihara, K., & Müller, K.-R. (2006). *Spectrally weighted common spatial pattern algorithm for single trial EEG classification*. Mathematical engineering technical reports. Department of Mathematical Informatics, Graduate School of Information Science and Technology, the University of Tokyo.

Treder, M. S., Bahramisharif, A., Schmidt, N. M., van Gerven, M. A., & Blankertz, B. (2011). Brain-computer interfacing using modulations of alpha activity induced by covert shifts of attention. *Journal of NeuroEngineering and Rehabilitation*, *8*(1), 24.

Urigüen, J. A. & Garcia-Zapirain, B. (2015). EEG artifact removal—state-of-the-art and guidelines. *Journal of Neural Engineering*, *12*(3), 031001.

Vanacker, G., del R. Millán, J., Lew, E., Ferrez, P. W., Moles, F. G., Philips, J., ... Nuttin, M. (2007). Context-based filtering for assisted brain-actuated wheelchair driving. *Intell. Neuroscience*, *2007*, 3–3.

Volosyak, I. (2011). SSVEP-based bremen–BCI interface—boosting information transfer rates. *Journal of Neural Engineering*, *8*(3), 036020.

Vos, M. D., Kroesen, M., Emkes, R., & Debener, S. (2014). P300 speller BCI with a mobile EEG system: Comparison to a traditional amplifier. *Journal of Neural Engineering*, *11*(3), 036008.

Wahba, G. (1981). Spline interpolation and smoothing on the sphere. *SIAM Journal on Scientific and Statistical Computing*, *2*(1), 5–16.

Wang, C. M. & Xiao, W. C. (2013). Second-order iir notch filter design and implementation of digital signal processing system. In *Instruments, measurement, electronics and information engineering* (Vol. 347, pp. 729–732). Applied Mechanics and Materials. Trans Tech Publications.

Wang, H., Li, Y., Long, J., Yu, T., & Gu, Z. (2014). An asynchronous wheelchair control by hybrid EEG-EOG brain-computer interface. *Cognitive Neurodynamics*, *8*(5), 399–409.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, *15*(2), 70–73.

Wolpaw, J. R., Ramoser, H., McFarland, D. J., & Pfurtscheller, G. (1998). EEG-based communication: Improved accuracy by response verification. *IEEE Transactions on Rehabilitation Engineering*, *6*(3), 326–333.

Yeom, S.-K., Fazli, S., Müller, K.-R., & Lee, S.-W. (2014). An efficient ERP-based brain-computer interface using random set presentation and face familiarity. *PLoS ONE*, *9*(11), 1–13.

Zander, T. O., Kothe, C., Welke, S., & Roetting, M. (2008). Enhancing Human-Machine Systems with Secondary Input from Passive Brain-Computer Interfaces. In *4th international brain-computer interface workshop and training course*. Graz, Austria.

Zander, T. O., Kothe, C., Jatzev, S., & Gaertner, M. (2010). Enhancing human-computer interaction with input from active and passive brain-computer interfaces. In S. D. Tan & A. Nijholt (Eds.), *Brain-computer interfaces: Applying our minds to human-computer interaction* (pp. 181–199). London: Springer London.

Zeyl, T., Yin, E., Keightley, M., & Chau, T. (2016). Partially supervised p300 speller adaptation for eventual stimulus timing optimization: Target confidence is superior to error-related potential score as an uncertain label. *Journal of Neural Engineering*, *13*(2), 026008.

Zhang, Y., Dong, L., Zhang, R., Yao, D., Zhang, Y., & Xu, P. (2014). An efficient frequency recognition method based on likelihood ratio test for SSVEP-based BCI. *Comp. Math. Methods in Medicine*, *2014*, 908719:1–908719:7.

Zhu, D., Bieger, J., Molina, G. G., & Aarts, R. M. (2010). A survey of stimulation methods used in SSVEP-based BCIs. *Intell. Neuroscience*, *2010*, 1:1–1:12.

Zusammenfassung

In der vorliegenden Arbeit wird das EEG-basierte Brain-Computer Interface (BCI) zur Steuerung von autonomen Vorrichtungen vorgestellt. Das Ziel ist es solch ein System einzusetzen, um die Mobilität von Personen wiederherzustellen, die unter einer neuromuskulären Störung leiden und mechanisch keine elektronische Fortbewegungsmittel oder andere Geräte nutzen können. Dieses BCI wurde anhand eines autonomen Rollstuhls unter unterschiedlichen Umständen, einschließlich des öffentlichen Raums, untersucht.

Das BCI vereint P300 ausgelöste Potentiale und steady-state visual evozierte Potentiale (SSVEPs), um die Absicht des Nutzers zu entschlüsseln. Objektplatzierung und andere Steuerbefehle werden über ein P300-basiertes interaktives Menü ausgewählt, während das Gerät mittels der Modulation von SSVEP Amplituden angehalten wird. Um das unbeabsichtigte Auslösen eines Befehls zu vermeiden, wenn der Nutzer nicht das Interface beachtet, kann das P300-basierte BCI nur dann benutzt werden, wenn diese Funktion über das SSVEP-basierte BCI freigeschaltet wird. Der hauptsächliche Vorteil dieses Hybriddesigns im Vergleich zu rein P300-basierte Systeme ist, dass der Rollstuhl ohne kognitiven Aufwand schnell angehalten werden kann und, des Weiteren, falsch-positive Interaktionen weitestgehend ausgeschlossen werden können.

Die Eigenständigkeit des P300-basierten BCIs wird gewährleistet, in dem der Klassifikator während des normalen Betriebs des Interface angepasst wird. Kürzlich erfasste Messwerte werden genutzt, um den Klassifikator durchgängig anzupassen. Des Weiteren stellt diese Arbeit eine neuartige Methode zur auto-Kalibrierung des P300-basierten BCIs vor, die weder gekennzeichnete Daten noch den Eingriff des Nutzers benötigt. Die automatische Kalibrierung und die Anpassung des P300 Klassifikators erlauben den Nutzer das Interface auch ohne Beihilfe jederzeit in optimalen Bedingungen zu betreiben.

Die Vermeidung von falsch-positiven Ergebnissen des SSVEP-basierten BCI wird ebenfalls im Rahmen dieser Arbeit behandelt. In diesem Zusammenhang wird eine neue Methode vorgeschlagen, um die SSVEP-bezogene Aktivität zu detektieren, die eine geringe Einweisung benötigt und einfach zu adaptieren ist. Dieser neue Ansatz benötigt nur Daten ohne SSVEP Modulation, um die Parameter des Klassifikators zu finden, die die Quote an falsch-positiven Ergebnissen unter einem gewünschten Niveau hält. Während der online Probe mit dem Rollstuhl wurden, auch im schlimmsten Fall, nicht mehr als 0.067 falsch-positive Ergebnisse pro Minute detektiert.

Das vorgeschlagene BCI beinhaltet ebenfalls zwei passive Subsysteme, um kognitive Zustände und Fehler bezogene Potentiale (ErrPs, engl. error-related potentials) zu quantifizieren. Das Erste wird genutzt, um Gehirnschwingungen zu Messen, die mit der Auseinandersetzung mit Aufgabenstellungen und der Erschöpfung zusammenhängen. Die Quantifizierung beider Variablen bietet die Möglichkeit die Interaktion zwischen Nutzer und Gerät zu verbessern. Auf der anderen Seite wird das andere BCI verwendet, um maschinelle Fehler automatisch zu detektieren, sobald der Nutzer bemerkt, dass das P300-basierte BCI eine Option irrtümlich ausgewählt hat. Dieses automatische Feedback kann genutzt werden, um die Funktion des Hybrid-BCIs, in Fällen wo der P300 Klassifikator eine geringe Genauigkeit aufweist, zu verbessern. Wenn das System jedoch die P300 Signale in den meisten Fällen richtig detektiert, wird das ErrP-basierte BCI nicht benötigt und die Applikation dadurch verlangsamt.

Das abschließende Ergebnis ist ein Hybrid-BCI, das gleichzeitig unterschiedliche Komponenten der elektrischen neuronalen Aktivität analysiert. Die vorliegende Arbeit beinhaltet eine vollständige Untersuchung der vier implementierten BCIs und eine abschließende Prüfung des Hybridsystems anhand eines autonomen Rollstuhls. Die hiermit vorgestellten Ergebnisse zeigen, dass das vorgeschlagene Interface dem autonomen Gerät mit großer Effizienz die Absicht des Nutzers vermitteln kann. Es wird gezeigt, dass Testpersonen ohne apriori Kenntnisse der BCI Technologie das Interface nach wenigen Übungsminuten beherrschen und dabei eine Genauigkeit von 100% erreichen können.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich alle Hilfsmittel und Hilfe angegeben habe und auf dieser Grundlage die Arbeit selbstständig verfasst habe.

Ich erkläre weiterhin, dass die Arbeit nicht schon einmal in einem früheren Promotionsverfahren eingereicht wurde.

Berlin, den 22. Dezember 2017

Omar Mendoza Montoya