Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications

Fabien Lotte

N° in order: D08-26

Thèse

Présentée devant

l'Institut National des Sciences Appliquées de Rennes

pour obtenir

Le grade de : DOCTEUR DE L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES
Mention INFORMATIQUE

par

Fabien LOTTE

Equipe : Bunraku - IRISA
Ecole doctorale : Matisse
Composante universitaire : INSA DE RENNES

Titre de la thèse :

Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications

soutenue le 4 décembre 2008 devant la commission d'examen

M. :	Pr. Pascal	GUITTON	President
M. :	Pr. Bruno	ARNALDI	Directeur de thèse
M. :	Dr. Anatole	LÉCUYER	Encadrant de thèse
MM. :	Pr. François	CABESTAING	Rapporteurs
	Pr. Touradj	EBRAHIMI	
M. :	Pr. Alain	RAKOTOMAMONJY	Examinateur

The brain is a wonderful organ: it starts working the moment you get up in the morning, and does not stop until you get into the office.
Robert Frost

À Marie…

Remerciements

Tout d'abord je tiens à remercier les différents membres de mon jury d'avoir accepté de juger mon travail et mon manuscript. Ainsi, je remercie Pr. Pascal Guitton d'avoir bien voulu présider mon jury de thèse, Pr. François Cabestaing et Pr. Touradj Ebrahimi d'avoir accepté de rapporter sur mon manuscript de thèse, et enfin Pr. Alain Rakotomamonjy d'avoir bien voulu examiner mon travail. Merci également pour vos commentaires pertinents sur mon travail et pour vos conseils.

Je souhaiterais remercier énormément Dr. Anatole Lécuyer et Pr. Bruno Arnaldi qui m'ont encadré au cours de cette thèse. Merci beaucoup pour tous vos nombreux conseils, votre soutient, vos encouragements, votre confiance et aussi pour votre sympathie. J'ai vraiment eu beaucoup de plaisir à travailler avec vous.

Merci aussi à toute l'équipe du projet OpenViBE pour vos conseils, votre aide et votre bonne humeur. Merci donc à Dr. Olivier Bertrand, Dr. Antoine Souloumiac, Dr. Christian Jutten, Dr. Jeremy Mattout, Dr. Denis Chêne, Dr. Bernard Hennion, Dr. Marco Congedo, Dr. Karim Gerbi, Ornella Plos, Cédric Gouy-Pailler, Dr. Bertrand Rivet, Yann Renard, Vincent Delannoy, Dr. Virginie Attina, Dr. Guillaume Gibert, Dr. Emmanuel Maby, Dr. Jean-Philippe Lachaux, Claude Dumas, Dr. Sophie Heinrich, Dr. Nathan Weisz (et j'en oublie surement !). Un merci spécial à Virginie, Guillaume et Manu pour m'avoir initié à l'électroencéphalographie et merci à Cédric pour avoir été le cobaye (encore pardon pour les dégats...). Un merci spécial également pour Marco, qui m'a initié aux interfaces cerveau-ordinateur avec Anatole, et qui a continué à m'aider par la suite. Enfin, un très gros merci à Yann, avec qui j'ai travaillé tout au long de ces trois années de thèse, et qui a été d'une aide inestimable.

Merci aussi à toutes les autres personnes avec qui j'ai pu collaborer et/ou échanger pendant cette thèse. Merci donc à Dr. Mingjun Zhong, Pr. Marc Girolami, Dr. Harold Mouchere, Dr. Hideaki Touyama, Rika Ito, Junya Fujisawa, Dr. Kunihiro Nishimura, Pr. Michitaka Hirose, Thomas Ernest, Jean-Baptiste Sauvan, Bruno Renier, Ludovic Hoyet, Dr. Fabrice Lamarche, Dr. Robert Leeb, Pr. Richard Reilly, Pr. Mel Slater, Dr. Ricardo Ron-Angevin, Dr. Areti

Tzelepi, Dr. Nicolas Brodu, Dr. Rémi Gribonval, Pr. Oscar Yanez-Suarez, Dr. Alois Schlogl, Dr. Stéphanie Gerbaud, Louis Mayaud et probablement d'autres que j'oublie mais que je remercie également.

Un grand merci à L'équipe Siames/Bunraku. Là je ne vais pas essayer de citer tout le monde car il y en a vraiment beaucoup et je suis sûr d'en oublier malgré moi, mais merci à tous pour ces années vraiment agréables, ces (nombreux) pots mémorables, ces soirées complètement folles, bref merci à tous pour cette ambiance inoubliable ! Mention spéciale pour mes deux "coburals" (ou peut être devrais-je dire "cobureaux" je ne sais pas trop ...), Yann Jehanneuf et Yann Renard pour "l'ambiance de folie" dans le bureau.

Merci à Dr. Jean Sreng de m'avoir supporté pendant plus d'un mois au Japon.

Merci à Angélique Jarnoux pour toute l'aide qu'elle m'a apporté pendant cette thèse.

Merci beaucoup à Morgane Rosendale pour avoir supporté que je rentre parfois tard le soir et que je l'abandonne régulièrement pour aller à des meeting ou à des conférences. Et merci également à Morgane pour avoir relu et corrigé l'anglais de tous mes articles, y compris l'anglais de cette thèse !

Merci à ma famille et à mes amis pour leur soutient et pour avoir supporter mes conversations de chercheur finalement pas palpitantes pour tout le monde...

Enfin, merci à tous ceux que j'ai oublié mais qui méritent quand même d'être remerciés !

Contents

Introduction

Since the first experiments of ElectroEncephaloGraphy (EEG) on humans by Hans Berger in 1929 [Ber29], the idea that brain activity could be used as a communication channel has rapidly emerged. Indeed, EEG is a technique which makes it possible to measure, on the scalp and in real-time, micro currents that reflect the brain activity. As such, the EEG discovery has enabled researchers to measure the human's brain activity and to start trying to decode this activity.

However, it is only in 1973 that the first prototype of a Brain-Computer Interface (BCI) came out, in the laboratory of Dr. Vidal [Vid73]. A BCI is a communication system which enables a person to send commands to an electronique device, only by means of voluntary variations of his brain activity [WBM+02, Bir06, PNB05, CR07, HVE07]. Such a system appears as a particularly promising communication channel for persons suffering from severe paralysis, for instance for persons suffering from amyotrophic lateral sclerosis [KKK+01]. Indeed, such persons may be affected by the "locked-in" syndrome and, as such, are locked into their own body without any residual muscle control. Consequently, a BCI appears as their only means of communication.

Since the 90's, BCI research has started to increase rapidly, with more and more laboratories worldwide getting involved in this research. Several international BCI competitions even took place in order to identify the most efficient BCI systems over the world [SGM+03, BMC+04, BMK+06]. Since then, numerous BCI prototypes and applications have been proposed, mostly in the medical domain [RBG+07, BKG+00], but also in the field of multimedia and virtual reality [KBCM07, EVG03, LLR+08].

Brain-Computer Interfaces

Naturally, designing a BCI is a complex and difficult task which requires multidisciplinary skills such as computer science, signal processing, neurosciences or psychology. Indeed, in order to use a BCI, two phases are generally required: 1) an offline training phase which calibrates the system and 2) an online phase which uses the BCI to recognize mental states and translates them into commands for a computer. An online BCI requires to follow a closed-loop process, generally composed of six steps: brain activity measurement, preprocessing, feature extraction, classification, translation into a command and feedback [MB03]:

1. **Brain activity measurement:** this step consists in using various types of sensors in order to obtain signals reflecting the user's brain activity [WLA+06]. In this manuscript we focus on EEG as the measurement technology.

2. **Preprocessing:** this step consists in cleaning and denoising input data in order to enhance the relevant information embedded in the signals [BFWB07].

3. **Feature extraction:** feature extraction aims at describing the signals by a few relevant values called “features” [BFWB07].

4. **Classification:** the classification step assigns a class to a set of features extracted from the signals [LCL+07]. This class corresponds to the kind of mental state identified. This step can also be denoted as “feature translation” [MB03]. Classification algorithms are known as “classifiers”.

5. **Translation into a command/application:** once the mental state is identified, a command is associated to this mental state in order to control a given application such as a speller (text editor) or a robot [KMHD06].

6. **Feedback:** finally, this step provides the user with a feedback about the identified mental state. This aims at helping the user controlling his brain activity and as such the BCI [WBM+02]. The overall objective is to increase the user’s performances.

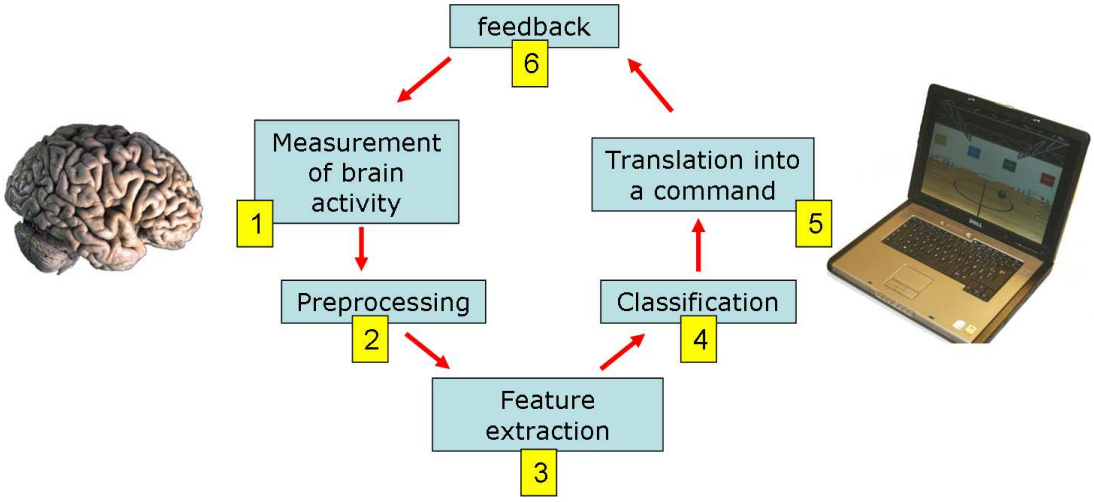


Figure 1: General architecture of an online brain-computer interface.

This whole architecture is summarized in Figure 1. These steps define an online BCI. However, as mentioned above, it should be noted that before operating such a BCI, a considerable calibration work is necessary. This work is generally done offline and aims at calibrating the classification algorithm, calibrating and selecting the optimal features, selecting the relevant sensors, etc. In order to do so, a training data set must have been recorded previously from the user. Indeed, EEG signals are highly subject-specific, and as such, current BCI systems must be calibrated and adapted to each user. This training data set should contain EEG signals recorded while the subject performed each mental task of interest several times, according to given instructions. The recorded EEG signals will be used as mental state examples in order to find the best calibration parameters for this subject.

Thesis objectives

The work presented in this PhD manuscript belongs to the framework of BCI research. More precisely, it focuses on the study of EEG signal processing and classification techniques in order to design and use BCI for interacting with virtual reality applications. Indeed, despite the valuable and promising achievements already obtained in the literature, the BCI field is still a relatively young research field and there is still much to do in order to make BCI become a mature technology. Among the numerous possible improvements, we are going to address three main points in this PhD thesis: improving the information transfer rate of current BCI, designing interpretable BCI systems and developping BCI systems for concrete real-life applications such as virtual reality applications. The BCI community have highlighted these points as being important and necessary research topics for the further development of BCI technology [MAM+06, KMHD06, WBM+02, AWW07, LSF+07].

1 - Improving the information transfer rate of BCI systems

Current BCI systems have a relatively low information transfer rate (for most BCI this rate is equal to or lower than 20 bits/min [WBM+02]). This means that with such BCI, the user needs a relatively long period of time in order to send only a small number of commands. In order to tackle this problem, we can address the following points:

- **Increasing the recognition rates of current BCI**. The performances of current systems remain modest, with accuracies, i.e., percentages of mental states correctly identified, which reach very rarely 100 %, even for BCI using only two classes (i.e., two kinds of mental states). A BCI system which is able to make less mistakes would be more convenient for the user and would provide a higher information transfer rate. Indeed, less mistakes from the system means less time required for correcting these mistakes.

- **Increasing the number of classes used** in current BCI. The number of classes used is generally very small for BCI. Most current BCI propose only 2 classes. Designing algorithms that can efficiently recognize a larger number of mental states would enable the subject to use more commands and as such to benefit from a higher information transfer rate [KCVP07, DBCM04a]. However, to really increase the information transfer rate, the classifier accuracy (percentage of correctly classified mental states) should not decrease too much due to the higher number of classes.

- **Designing asynchronous (self-paced) BCI**. Current BCI are mostly synchronous, which means their users can only interact with the application during specific time periods, instructed by the system. Contrary to synchronous BCI, self-paced BCI can issue commands at any time, and as such can issue more commands than synchronous BCI within the same time period [MKH+06]. Consequently, their resulting information transfer rate should also be higher.

2 - Designing interpretable BCI systems

Currently, the brain mechanisms are still far from being fully understood, and a considerable amount of neuroscience research is still required to achieve this goal, if ever. Research on BCI systems, which aim at decoding brain activity in real time, may be seen as a promising way of improving the understanding of the brain. Indeed, most current BCI systems can be trained to recognize various mental states using a set of training data. Consequently, the BCI community has recently stressed the need to develop signal processing and classification techniques for BCI from which a human could extract knowledge and gain insights on the brain dynamics [MAM+06]. Moreover, even since the very beginning of BCI research, it has been highlighted that the employed data analysis methods "should enable interpretation, so that the researchers can use the results for further improvement of the experimental setting" [KFP93]. However, current BCI systems generally behave as "black boxes", i.e., it is not possible to interpret what the algorithms have automatically learnt from data [MAM+06]. Designing interpretable BCI would make it possible to obtain systems that can recognize various mental states while providing knowledge on the properties and dynamics of these mental states. Such a system could potentially be used to improve the current neuroscience knowledge as well as to check and improve the designed BCI.

3 - Developping BCI systems for concrete virtual reality applications

With only a few exceptions [GEH+03, VMS+06], current BCI systems are mostly studied and evaluated inside laboratories, in highly controlled conditions. To further develop BCI technology, it is necessary to study BCI in real-life or close to real-life conditions, with a larger number of users/subjects. It is also essential to exploit efficiently the few available commands provided by current BCI systems. Indeed, by designing smart interfaces and appropriate interaction paradigms for BCI based applications, the amount of possible and available actions for the user could be increased, and the time necessary to select these actions could be decreased. In this thesis manuscript, we focus on Virtual Reality (VR) applications. Indeed we would like to develop, study and improve BCI-based VR applications, such as entertaining applications that could be used by the general public.

Approach and contributions

This manuscript describes the work we carried out in order to address the three objectives mentioned above. More precisely, the **first chapter** of this manuscript proposes an overview of current BCI designs and applications. The following chapters are dedicated to the scientific contributions we proposed. These contributions can be gathered into two parts: Part 1) gathers contributions related to EEG signal processing and classification whereas Part 2) gathers contributions related to virtual reality applications based on BCI technology. More precisely, Part 1 deals respectively with feature extraction and preprocessing, classification, interpretable BCI design and self-paced BCI design. Part 2 deals respectively with the study of BCI use for entertaining VR applications in close to real-life conditions, and with the use of a BCI for exploring and interacting with a Virtual Environment (VE), here a virtual museum. More details

are given in the following sections.

Part 1: EEG signal processing and classification

In order to increase the information transfer rates of current BCI systems and to design interpretable BCI, improvements can be brought at all processing levels: at the preprocessing level, at the feature extraction level and at the classification level. To improve the information transfer rate at the feature extraction level, we could design more robust and efficient features. To this end, we should design algorithms that can capture the relevant information related to each targeted mental state while filtering away noise or any unrelated information. Moreover, it is known that each subject is different from the other, regarding the spectral or spatial components of his brain activity for instance. Consequently, an ideal feature extraction algorithm for BCI should be trainable in the sense that it should be able to learn and use subject-specific features. Moreover, it is particularly important to design feature extraction methods that can be trained on multiclass data (e.g., [DBCM04b]).

In order to obtain an interpretable BCI, we can first obtain interpretable features. Features that are abstract mathematical information such as autoregressive coefficients (see section 1.5.1.2) are very unlikely to be interpreted by a human. To be able to extract relevant information about the brain dynamics from the features, the ideal features should convey physiological information.

We believe that inverse solutions are ideal candidates to address all these points. Indeed, inverse solutions are methods that make it possible to reconstruct the activity in the brain volume by using only scalp measurements and a head model (see section 1.4.4). As such they can localize the sources of activity within the brain, thus recovering a physiologically relevant information. Moreover, in works preceding this PhD thesis, we have shown that inverse solutions were promising and efficient (in terms of classification accuracy) spatial filters for BCI [CLL06]. Other pioneer studies performed by different groups have also found that inverse solutions were promising feature extraction methods for EEG-based BCI [GGP$^+$05, KLH05]. Consequently, in **Chapter 2**, we propose a trainable feature extraction algorithm for BCI which relies on inverse solutions and can deal with multiclass problems. The proposed algorithm, known as FuRIA (**Fu**zzy **R**egion of **I**nterest **A**ctivity), is assessed on EEG data sets from BCI competitions and compared with the algorithms of the competition winners.

To build an interpretable BCI system with a high information transfer rate it is also necessary to work at the classification level. To date, numerous classifiers have been tried and used to design BCI [MAB03, LCL$^+$07] (see also section 1.6). However, some classifiers that proved to be efficient in other contexts of pattern recognition have not been explored yet for BCI design [MAM$^+$06]. A category of classifiers appears as particularly attractive for BCI design: the fuzzy classifiers, which are classification algorithms based on the theory of fuzzy logic and fuzzy sets of Zadeh [Zad96a, Men95]. Indeed, Bezdek has highlighted that fuzzy classifiers were "perfectly suitable to deal with the natural fuzziness of real-life classification problems" [BP92].

Fuzzy Inference Systems (FIS) are fuzzy classifiers that can learn fuzzy “if-then” rules able

to classify data [Chi97a]. FIS have been successfully used in several other pattern recognition tasks such as hand-writing recognition [AL96b], ElectroMyoGraphy (EMG) classification [CYL+00] or even brain research, e.g., for EEG monitoring [BU03, HLS+01]. Moreover, Chan *et al* have stressed the suitability of FIS for classification of non-stationary biomedical signals [CYL+00]. Actually, the "fuzzyness" of FIS makes it possible to deal with the variability of such signals and to tolerate their possible contradictions [CYL+00]. In addition to these points, FIS exhibit several interesting properties that may address our objectives. First, FIS are universal approximators [Wan92]. Then, FIS are known to be interpretable, which means that it is possible to extract knowledge from what they automatically learnt [Gui01, Chi97a]. Finally, it is also possible to add a priori knowledge to FIS under the form of "hand-made" fuzzy rules [Chi97a]. All these points make FIS very interesting candidates for BCI-design.

Therefore, in **Chapter 3**, we study the use of a FIS for classification in an EEG-based BCI. More particularly, we study FIS by assessing their classification performances, their interpretability, the possibility to provide them with a priori knowledge. We also study their outlier rejection capabilities, i.e., their capabilities to reject data that do not correspond to any of the classes they learnt. For this study, we focus on the classification of EEG signals recorded during movement imagination.

As mentioned above, inverse solution-based features represent physiological knowledge and fuzzy inference systems can represent what they have learnt under the form of a set of rules. These two properties appear as particularly interesting to attain our objective of designing an interpretable BCI. However, the interpretability of these methods could be pushed further. Indeed, a BCI system would be more easily interpretable if it could express the knowledge it has extracted from EEG data using simple and clear words.

Therefore, in **Chapter 4**, we propose an algorithm, which is based on inverse solutions and fuzzy inference systems, to design fully interpretable BCI systems. This algorithm relies on the paradigm of "computing with words" of Zadeh [Zad96b] in order to express what has been learnt by the BCI system using only simple words, and not mathematical formulations.

Finally, in order to design BCI with higher information transfer rate and to use them in real applications, it is essential to design efficient Self-Paced BCI (SPBCI). Moreover, and independently from the information transfer rate, a SPBCI provides the user with a more natural and convenient mode of interaction. This point is also important as one of our objectives is to design BCI-based virtual reality applications for the general public. Consequently, **Chapter 5** deals with the design of EEG-based SPBCI. More precisely, this chapter considers SPBCI design as a pattern rejection problem. As such, it introduces new pattern rejection methods for SPBCI design and studies and compares various pattern rejection methods applied to various classification algorithms.

Part 2: Virtual reality applications based on BCI technology

In order to design practical BCI-based applications, and particularly virtual reality applications, it appears as essential to gather knowledge about the influence, role and needs of the user of the

system. To gather relevant and significant knowledge, the related studies should be performed with a sufficiantly high number of subjects in close to real-life conditions. Consequently, in the work presented in **Chapter 6**, we studied both the performances and the preferences of 21 naive subjects who used a BCI to interact with an entertaining VR application during an exhibition. More precisely, this chapter presents a simple self-paced BCI which uses a single electrode and a single feature. Thanks to this BCI, subjects could lift a virtual spaceship by using real or imagined foot movements. The correct recognition rates obtained were measured, and the users' feelings about their BCI experience were collected using a questionnaire.

BCI have been recently shown to be a suitable interaction device for VR applications [LSF+07]. Indeed, various prototypes have been proposed in order to perform simple navigation tasks in VE by thoughts [LFS+06, FLG+07, LFSP07, SLS+08] as well as a few virtual object manipulation tasks [LKF+05a, Bay03]. Despite these promising first prototypes, current BCI-based VR applications can only provide the user with few and limited interaction tasks. Indeed current BCI systems can only provide the user with a very small number of commands (only 2 for most BCI), and current BCI-based VR applications mostly rely on low-level interaction techniques limiting the possibilities offered to the user.

In **Chapter 7**, we present a BCI-based VR application which enables its users to visit a virtual museum by using thoughts only. In order to exploit efficiently the small number of commands provided by a BCI we proposed a novel interaction technique for BCI-based VR application. This interaction technique enables the user to send high-level commands, leaving the application in charge of most of the complex and tedious details of the interaction task. We also designed a self-paced BCI system which can provide its users with 3 different commands.

Finally, conclusions and perspectives of the work presented in this manuscript are given in the **last chapter**.

Chapter 1

Brain-Computer Interfaces Design and Applications

1.1 Introduction

This first chapter aims at reviewing the main BCI designs and their applications. This chapter first gives some definitions related to BCI. Then it reviews the methods and techniques used to design a BCI. As such, it details the different processing steps composing a BCI, that is, measurements of brain activity (Section 1.3), preprocessing (Section 1.4), feature extraction (Section 1.5) and classification (Section 1.6). Finally, Section 1.7 presents some BCI applications and prototypes already developped, by emphasising virtual reality applications.

1.2 Definitions

A BCI can formally be defined as a "communication and control channel that does not depend on the brain's normal output channels of peripheral nerves and muscles" [WBM$^+$02]. The messages and commands sent through a BCI are encoded into the user's brain activity. Thus, a BCI user "produces" different mental states (alternatively, we can say that a user is performing a mental task or is generating a given neurophysiological signal) while his brain activity is being measured and processed by the system. Traditionally, the different BCI systems are divided into several categories. Among these categories, researchers notably oppose dependent BCI to independent BCI, invasive BCI to non-invasive BCI as well as synchronous BCI to asynchronous (self-paced) BCI.

1.2.1 Dependent versus independent BCI

One distinction which is generally made concerns dependent BCI versus independent BCI [AWW07]. A dependent BCI is a BCI which requires a certain level of motor control from the subject whereas an independent BCI does not require any motor control. For instance, some BCI require that the user can control his gaze [LKF$^+$05a]. In order to assist and help severely disabled people who do not have any motor control, a BCI must be independent. However,

dependent BCI can prove very interesting for healthy persons, in order to use video games for instance [AGG07]. Moreover, such BCI may be more comfortable and easier to use.

1.2.2 Invasive versus non-invasive BCI

A BCI system can be classified as an invasive or non-invasive BCI according to the way the brain activity is being measured within this BCI [WBM+02, LN06]. If the sensors used for measurement are placed within the brain, i.e., under the skull, the BCI is said to be invasive. On the contrary, if the measurement sensors are placed outside the head, on the scalp for instance, the BCI is said to be non-invasive. Please refer to Section 1.3 for more details on the brain activity measurement methods employed in BCI.

1.2.3 Synchronous versus asynchronous (self-paced) BCI

Another distinction that is often made concerns synchronous and asynchronous BCI. It should be noted that it is recommended to denote asynchronous BCI as "self-paced" BCI. [PGN06, MKH+06]. With a synchronous BCI, the user can interact with the targeted application only during specific time periods, imposed by the system [KFN+96, PNM+03, WBM+02]. Hence, the system informs the user, thanks to dedicated stimuli (generally visual or auditory), about the time location of these periods during which he has to interact with the application. The user has to perform mental tasks during these periods only. If he performs mental tasks outside of these periods, nothing will happen.

On the contrary, with a self-paced BCI, the user can produce a mental task in order to interact with the application at any time [MKH+06, BWB07, SSL+07, MM03]. He can also choose not to interact with the system, by not performing any of the mental states used for control. In such a case, the application would not react (if the BCI works properly).

Naturally, self-paced BCI are the most flexible and comfortable to use. Ideally, all BCI should be self-paced. However, it should be noted that designing a self-paced BCI is much more difficult than designing a synchronous BCI. Indeed, with synchronous BCI, the system already knows when the mental states should be classified. With a self-paced BCI, the system has to analyse continuously the input brain signals in order to determine whether the user is trying to interact with the system by performing a mental task. If it is the case, the system has also to determine what is the mental task that the user is performing. For these reasons, the wide majority of currently existing BCI are synchronous [WBM+02, PNB05]. Designing an efficient self-paced BCI is presently one of the biggest challenge of the BCI community and a growing number of groups are addressing this topic [MKH+06, BWB07, SSL+07, MM03].

1.3 Measurements of brain activity

The first step required to operate a BCI consists in measuring the subject's brain activity. Up to now, about half a dozen different kinds of brain signals have been identified as suitable for a BCI, i.e., easily observable and controllable [WBM+02]. This section first describes the different available techniques for measuring brain activity. Then it describes the brain signals that can be used to drive a BCI.

1.3.1 Techniques for measuring brain activity

1.3.1.1 Overview of measurement techniques used for BCI

Numerous techniques are available and used, in order to measure brain activity within a BCI [WLA+06, dM03]. Among these techniques, we can quote MagnetoEncephaloGraphy (MEG) [MSB+07, BJL+08], functional Magnetic Resonance Imaging (fMRI) [WMB+04], Near InfraRed Spectroscopy (NIRS) [CWM07], ElectroCorticoGraphy (ECoG) [LMS+06] or implanted electrodes, placed under the skull [LN06]. However, the most used method is by far ElectroEncephaloGraphy (EEG) [WLA+06]. Indeed, this method is relatively cheap, non-invasive, portable and provides a good time resolution. Consequently, most current BCI systems are using EEG in order to measure brain activity. Thus, in this thesis work, we have focused on EEG-based BCI designs.

1.3.1.2 Invasive BCI

Although EEG is the most widely used technique, it should be noted that a large and rapidly growing part of BCI research is dedicated to the use of implanted electrodes which measure the activity of groups of neurons [LN06, FZO+04, HSF+06, Nic01, SCWM06]. Currently, most of this research has focused on the design and evaluation of invasive BCI for primates [LN06, Nic01]. However, recent results have shown the usability of such systems on humans [HSF+06, SCWM06]. Implanted electrodes make it possible to obtain signals with a much better quality and a much better spatial resolution than with non-invasive methods. Indeed, some invasive methods can measure the activity of single neurons while a non-invasive method such as EEG measure the resulting activity of thousands of neurons. As such, it is suggested that invasive BCI could obtain better results, in terms of performances (information transfer rate, accuracy, fiability, ...), than non-invasive methods, and especially than EEG. However, this statement still needs to be confirmed and is still a topic of debate within the BCI community. Indeed, even if EEG-based BCI are based on much noisier and coarser signals than those of invasive BCI, some studies have reported that they can reach similar information transfer rates [WM04, Wol07]. The main drawback of invasive BCI is precisely the fact that they are invasive, which requires that the subject endures a surgery operation in order to use the system. Moreover, implanted electrodes have a limited lifetime, which makes the subject endure regular surgery operations in order to replace the electrodes. Then, the use of implanted electrodes might be dangerous for the health of the subjects. Finally, implanting electrodes in a human's brain also raises numerous ethic problems. These points make non-invasive BCI, and most especially EEG-based BCI, the most used and the most popular BCI systems. In the following of this manuscript, we will focus exclusively on non-invasive BCI based on EEG.

1.3.1.3 Electroencephalography

Electroencephalography measures the electrical activity generated by the brain using electrodes placed on the scalp [NdS05]. EEG measures the sum of the post-synaptic potentials generated by thousands of neurons having the same radial orientation with respect to the scalp (see Figure 1.1). The first EEG measurements on a human subject have been conducted in 1924 by

Hans Berger. It is at that time that he worked out the name of “electroencephalogram”. His fundamental discovery was published in 1929 [Ber29].

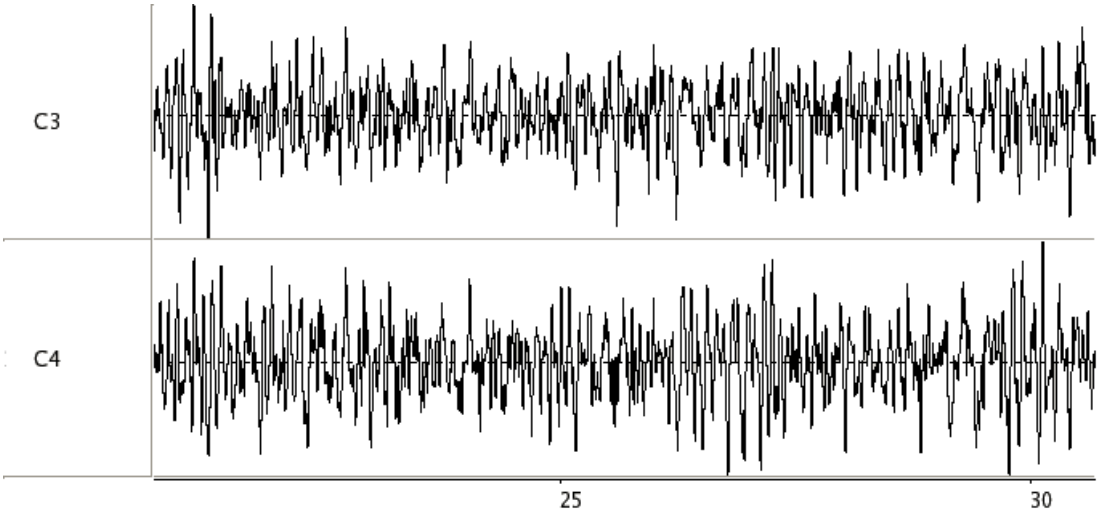


Figure 1.1: Examples of EEG signals, recorded using 2 EEG electrodes (C3 and C4) for a healthy subject (time is displayed in seconds).

Signals recorded by EEG have a very weak amplitude, in the order of some microvolts. It is thus necessary to strongly amplify these signals before digitizing and processing them. Typically, EEG signals measurements are performed using a number of electrodes which varies from 1 to about 256, these electrodes being generally attached using an elastic cap. The contact between the electrodes and the skin is generally enhanced by the use of a conductive gel or paste [Rei05]. This makes the electrode montage procedure a generally tedious and lengthy operation. It is interesting to note that BCI researchers have recently proposed and validated dry electrodes for BCI, that is, electrodes which do not require conductive gels or pastes for use [PFB$^+$07]. However, the performance of the resulting BCI (in terms of maximum information rate) were, on average, 30% lower than the one obtained with a BCI based on electrodes that use conductive gels or pastes.

Electrodes are generally placed and named according to a standard model, namely, the 10-20 international system [Jas58] (see Figure 1.2). This system has been initially designed for 19 electrodes, however, extended versions have been proposed in order to deal with a larger number of electrodes [AES91].

EEG signals are composed of different oscillations named “rhythms” [Nie05]. These rhythms have distinct properties in terms of spatial and spectral localization. There are 6 classical brain rhythms (see Figure 1.3):

- **Delta rhythm:** This is a slow rhythm (1-4 Hz), with a relatively large amplitude, which is mainly found in adults during a deep sleep.
- **Theta rhythm:** This a slightly faster rhythm (4-7 Hz), observed mainly during drowsiness and in young children.

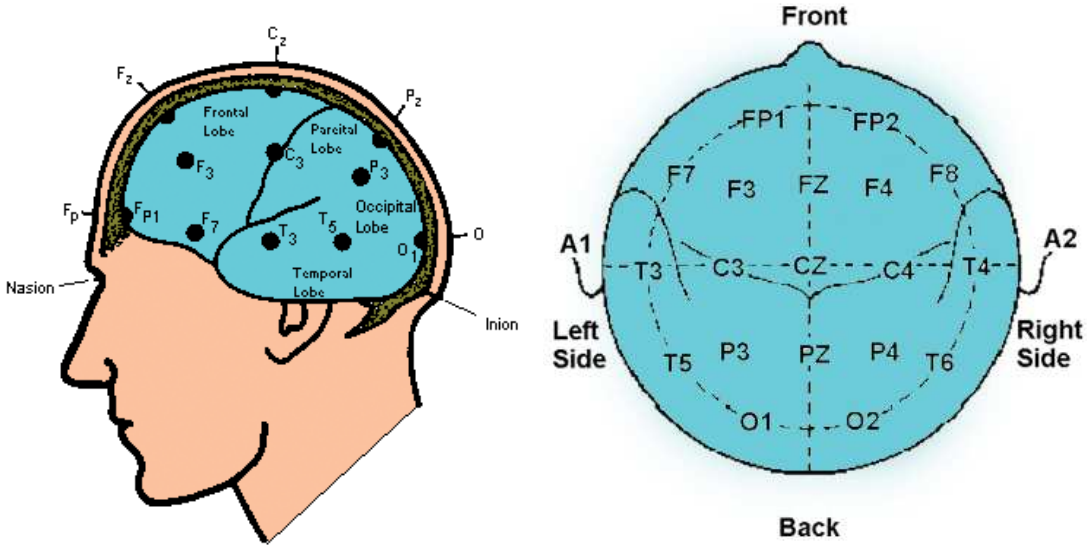


Figure 1.2: Positions and names of the 10-20 international system electrodes (pictures from www.wikipedia.org).

- **Alpha rhythm:** These are oscillations, located in the 8-12 Hz frequency band, which appear mainly in the posterior regions of the head (occipital lobe) when the subject has closed eyes or is in a relaxation state.
- **Mu rhythm:** These are oscillations in the 8-13 Hz frequency band, being located in the motor and sensorimotor cortex. The amplitude of this rhythm varies when the subject performs movements. Consequently, this rhythm is also known as the "sensorimotor rhythm" [PN01].
- **Beta rhythm:** This is a relatively fast rhythm, belonging approximately to the 13-30 Hz frequency band. It is a rhythm which is observed in awaken and conscious persons. This rhythm is also affected by the performance of movements, in the motor areas [PN01].
- **Gamma rhythm:** This rhythm concerns mainly frequencies above 30 Hz. This rhythm is sometimes defined has having a maximal frequency around 80 Hz or 100 Hz. It is associated to various cognitive and motor functions.

1.3.2 Neurophysiological signals used to drive a BCI

BCI aim at identifying, in the brain activity measurements of a given subject, one or several specific neurophysiological signals (i.e., brain activity patterns), in order to associate a command to each of these signals. Various signals have been studied and some of them were revealed as relatively easy to identify (automatically), as well as relatively easy to control for the user. These signals can be divided into two main categories [CS03, WBM$^+$02]:

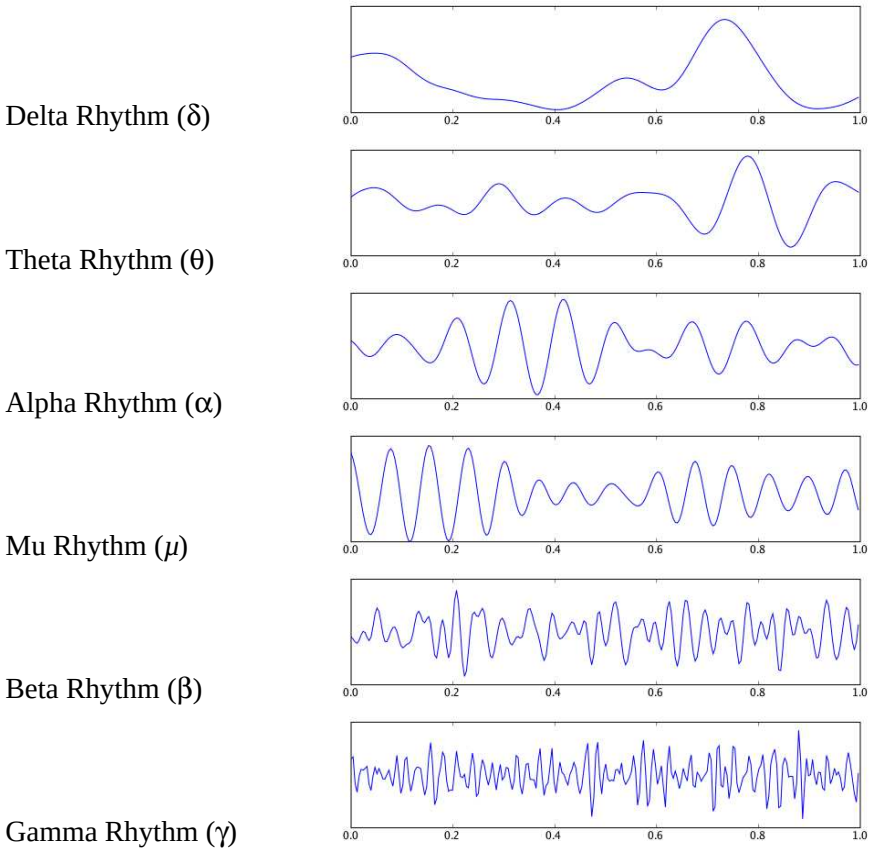


Figure 1.3: The different brain rhythms as measured by EEG (pictures from www.wikipedia.org).

- **Evoked signals** that are generated unconsciously by the subject when he perceives a specific external stimulus. Those signals are also known as **Evoked Potentials** (EP).

- **Spontaneaous signals** that are voluntarily generated by the user, without external stimulation, following an internal cognitive process.

In the following of this manuscript we will also denote a neurophysiological signal as a mental state or as a brain activity pattern. These three names will denote the same entity.

1.3.2.1 Evoked potentials

In this first category, the main signals are the Steady State Evoked Potentials (SSEP) and the P300 [WBM+02, CS03]. These two potentials are described further in this section. The main advantage of EP is that, contrary to spontaneous signals, evoked potentials do not require a specific training for the user, as they are automatically generated by the brain in response to a stimulus. As such, they can be used efficiently to drive a BCI since the first use [WBM+02,

CS03]. Nevertheless, as these signals are evoked, they require using external stimulations, which can be uncomfortable, cumbersome or tiring for the user.

Steady State Evoked Potentials: SSEP are brain potentials that appear when the subject perceives a periodic stimulus such as a flickering picture or a sound modulated in amplitude. SSEP are defined by an increase of the EEG signals power in the frequencies being equal to the stimulation frequency or being equal to its harmonics and/or sub-harmonics [LKF+05a, GPAR+07b, MPSNP06]. Various kinds of SSEP are used for BCI, such as Steady State Visual Evoked Potentials (SSVEP) [LKF+05a, MCM+95, TH07b, SEGYS07], which are by far the most used, somatosensory SSEP [MPSNP06] or auditory SSEP [GPAR+07b, GPAR+07a] (see Figure 1.4 for an example of SSVEP). These SSEP appear in the brain areas corresponding to the sense which is being stimulated, such as the visual areas when a SSVEP is used.

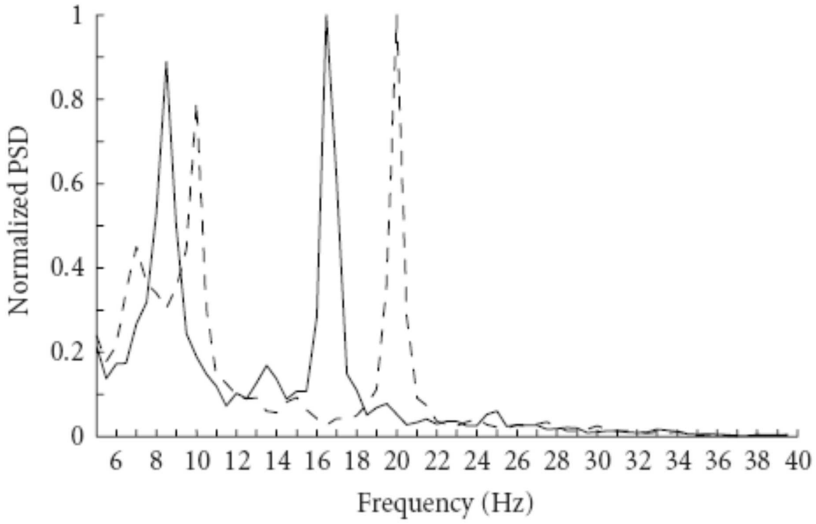


Figure 1.4: EEG spectrum showing SSVEP for stimulation frequencies of 17 Hz (plain line) or 20 Hz (dotted line). We can clearly notice the peak of power at the stimulation frequencies and their sub-harmonic (pictures from [LKF+05a]).

An advantage of this kind of signals is that they can be used within a BCI without training. Moreover, as stimuli with different stimulation frequencies will lead to SSEP with different frequencies, it is possible to use a large number of stimuli in order to obtain and use a large number of mental states for the BCI (e.g., see [GXCG03] were 48 stimuli were used). As such, it enables the user to have a large number of commands which makes the whole system more convenient. This explains the increasing interest of the BCI community for SSEP, and more especially for SSVEP. [MMCJ00, CGGX02, GXCG03, TRM06, NCdN06, MPP08]. For instance, a BCI application based on SSVEP can use several flickering buttons displayed on screen, each button having a unique flickering frequency. In such an application, the user should draw his attention on the button he wants to activate. Indeed, it is known that the SSVEP corresponding to a given button are enhanced when the user draws his attention on this

button. Detecting, within EEG signals, an SSVEP with a frequency f will then activate the button with a flickering frequency of f as this button is very likely to be the one the user was focusing on [CGGX02].

P300: The P300 consists of a Positive waveform appearing approximately 300 ms after a rare and relevant stimulus (see Figure 1.5) [FD88]. It is typically generated through the "odd-ball" paradigm, in which the user is requested to attend to a random sequence composed of two kind of stimuli with one of these stimuli being less frequent than the other. If the rare stimulus is relevant to the user, its actual appearance triggers a P300 observable in the user's EEG. This potential is mainly located in the parietal areas.

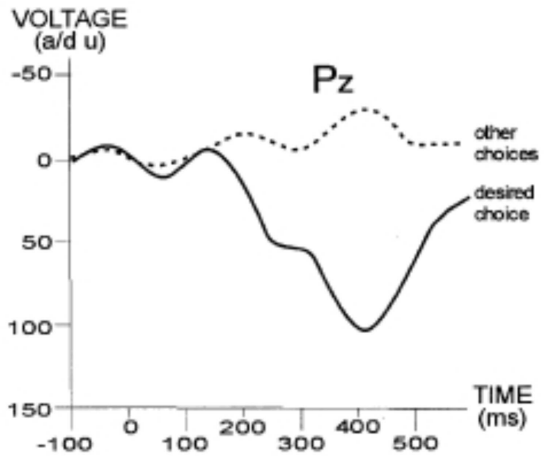


Figure 1.5: A P300 (enhanced by signal averaging) occuring when the desired choice appears (picture from [WBM+02]).

Generally, a P300-based BCI uses the fact that the P300 is present or missing from the input EEG signals in order to send, or not, a command to the application. Similarly to SSVEP-based BCI applications, in P300-based BCI applications, several buttons or objects are displayed on the user's screen. These buttons or objects are randomly higlighted, and the user is instructed to count, over a finite time period, the number of times that the button he wants to activate is highlighted. This aims at making the highlight of the desired button a rare and relevant stimulus in order to trigger the appearance of the P300. Thus, when a P300 is detected in the EEG signals, the system identifies the button desired by the user as the button which was highlighted 300 ms earlier, as this button is most likely to be the one for which the user was counting the number of highlights. The P300 is mostly used in a kind of "virtual keyboard" application known as the P300 speller [FD88, RS07b, KSC+06, SD06, PGT+06]. This application is described in more details in section 1.7.1.2. As other EP, the P300 has the advantage of requiring no training for the subject in order to be used. On the other hand, P300-based BCI applications require the user to constantly focus on fast and repetitive visual stimuli, which can be tiring and inconvenient.

1.3.2.2 Spontaneous signals

Within the category of spontaneous signals, the most used signals are undoubtly sensorimotor rhythms. However, other neurophysiological signals are used, such as slow cortical potentials or non-motor cognitive signals.

Motor and sensorimotor rhythms: sensorimotor rhythms are brain rhythms related to motor actions, such as arm movements, for instance. These rhythms, which are mainly located in the μ ($\simeq$ 8-13 Hz) and β ($\simeq$ 13-30 Hz) frequency bands, over the motor cortex, can be voluntarily controlled in amplitude by a user. When it comes to BCI, two different strategies have been proposed in order to make the user control these sensorimotor rhythms:

- **Operant conditioning:** A subject can learn to modify voluntarily the amplitude of his sensorimotor rhythms through a (very) long training known as "operant conditioning" [WM04, WMNF91, VMS$^+$06, Wol07] (see Figure 1.6).

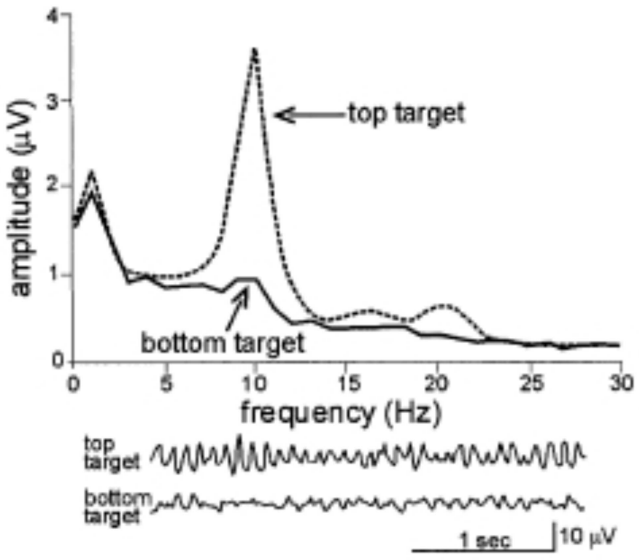


Figure 1.6: sensorimotor rhythm variations performed voluntarily by the subject between two conditions: "top target" and "bottom target" (picture from [WBM$^+$02]).

  In order to reach this goal, the user is free to select the mental strategy he is the most comfortable with. Motor imagery (see below) is one possible strategy which is often used. When using operant conditioning, the role of the feedback is essential, as it enables the user to understand how he should modify his brain activity in order to control the system. Generally, in BCI based on operant conditioning, the power of the μ and β rhythms in different electrode locations are linearily combined in order to build a control signal which will be used to perform 1D, 2D or 3D cursor control [WM04, Wol07]. The main drawback of this method is the very long training time which is necessary. Indeed, the training

of a given user can last several weeks or even several months [WM04, WMNF91]. However, once this training is completed, very good performances (in terms of information transfer rate) can be obtained.

- **Motor imagery:** For a user, performing motor imagery consists in imagining movements of his own limbs (hands or feet for instance) [PN01, PBSdS06, PNM+03]. The signals resulting from performing or imagining a limb movement have very specific temporal, frequential and spatial features, which makes them relatively easy to recognize automatically [PBSdS06, PNFP97, PNSL98]. For instance, imagining a left hand movement is known to trigger a decrease of power (Event Related Desynchronisation (ERD)) in the μ and β rhythms, over the right motor cortex [PdS99] (see Figures 1.7 and 1.8).

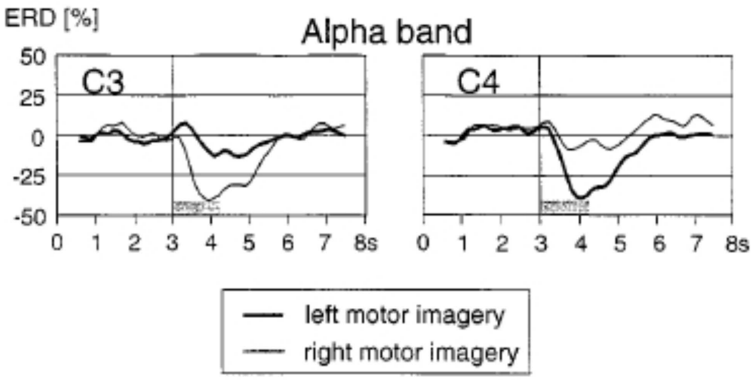


Figure 1.7: Time course of ERD following left hand and right hand motor imagery. The imagination starts at second 3 (picture from [PNG+00]).

A symmetric phenomenon appears when the user imagines a right hand movement. In motor imagery based BCI, the motor imagery task that has been identified (e.g., imagined left hand movement, imagined tongue movement, etc.) will be associated to a command, so as to control the movement of a cursor or the opening/closure of a prosthesis, for instance [PNM+03, SMN+04, GHHP99]. Using a motor imagery-based BCI generally requires a few sessions of training before being efficient [PGN06]. However, using advanced signal processing and machine learning algorithms enables the use of such a BCI with almost no training [BDK+07, BDK+06a].

Slow cortical potentials: Slow Cortical Potentials (SCP) are very slow variations of the cortical activity, which can last from hundreds of milliseconds to several seconds [BKG+00, KB05]. It is possible to learn to make these variations positive or negative using operant conditioning (see Figure 1.9).

Thus, SCP can be used in a BCI to generate a binary command, according to the positivity or negativity of the potential [BKG+00, KB05]. As the control of SCP is achieved by operant conditioning, mastering such a signal requires generally a very long training time. This training by operant conditioning is even longer for SCP than for motor rhythms [Bir06]. However, it seems that SCP would be a more stable signal [Bir06].

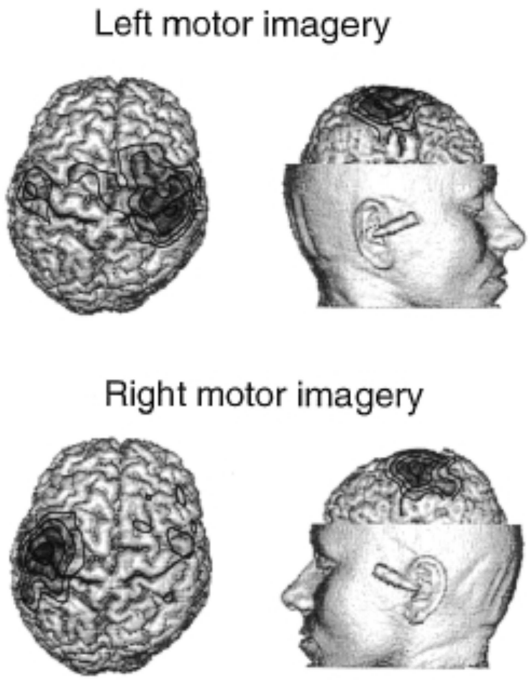


Figure 1.8: Spatial localization of ERD following left hand and right hand motor imagery (picture from [PNG+00]).

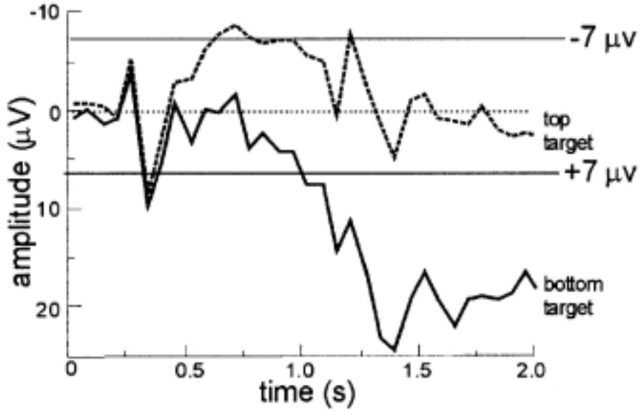


Figure 1.9: Voluntary variations of slow cortical potentials, between two conditions (reach the top target or reach the bottom target). (picture from [WBM+02])

Non-motor cognitive tasks: A relatively large number of non-motor cognitive processing tasks are also used in order to drive a BCI. These tasks are, for instance, mental mathematical computations, mental rotation of geometric figures, visual counting, mental generation of words, music imagination, etc. [CS03, dRMMC+00, BGM07b, CB04, KA90, ASS98]. All

these mental tasks generate specific EEG signal variations, in specific cortical regions and frequency bands, which makes them relatively easy to identify.

1.3.3 Conclusion

The neurophysiological signals presented in this section have all been used successfully in various applications. However, almost no comparisons of these signals have been performed so far. As such, it appears as difficult to select objectively the best one. All signals have their pros and cons. Evoked potentials can be used without subject training but require the use of external stimuli and can be tiring for the user. Spontaneous signals are more natural and comfortable to use, as they do not rely on external stimuli, but they generally require long training time. However, it seems that advanced machine learning and signal processing methods can reduce the need to train the subjects or even remove this training need [BDK+06a]. This is the reason why we focused on spontaneous signals in this thesis work. More specifically, we focused on motor imagery signals which are signals largely described in the literature and relatively natural to use for the subjects.

The three following sections are dedicated to the preprocessing, feature extraction and classification of EEG signals. These three BCI components could be gathered into a single and more general, higher level component, which could be denoted as "EEG processing". This component is a key element in the design of a BCI as it aims at transforming the input brain signals into a command for a given application. As such, the "EEG processing" component can be seen as the core of the BCI. Consequently, a wide majority of BCI research aims at improving this component to make the whole system more efficient.

It is important to note that the boundaries between the "preprocessing", "feature extraction" and "classification" components are not hard boundaries, and these boundaries may even appear as fuzzy. Furthermore, all these components are not necessarily used in all BCI [MBF+07]. Thus, the preprocessing and feature extraction components are sometimes merged into a single algorithm, whereas the classification algorithm can be missing or reduced to its simplest form, i.e., a decision threshold on the feature values. However, it is interesting to distinguish these components, as they have different inputs and outputs as well as different goals.

1.4 Preprocessing

Once the data have been acquired, they are generally preprocessed in order to clean (de-noise) the signals and/or to enhance relevant information embedded in these signals. Indeed, EEG signals are known to be very noisy, as they can be easily affected by the electrical activity of the eyes (EOG: ElectroOcculoGram) or of the muscles (EMG: ElectroMyoGram), e.g., face or jaw muscles [FBWB07]. These muscle artifacts are especially annoying as they have an amplitude which is much larger than the one of EEG signals. As such, it appears as difficult to remove these artifacts without accidentaly removing relevant information embedded in these EEG signals. Moreover, it is interesting to remove the background brain activity which is not related to the neurophysiological signals of interest. Overall, the preprocessing step can

be defined as a method which transforms a set of signals into a new set of signals which are supposedly denoised. In other words, the preprocessing step aims at increasing the signal-to-noise ratio of the input signals.

In order to perform this preprocessing, various spatio-spectro-temporal filters are used [MMDW97, RMGP00, BGM07a]. These filters can be simple frequency filters or more advanced filters such as independant component analysis [ZL06, NBL$^+$06, MEJS00, KASC08] or common spatial patterns [RMGP00, CLL06, BB04, WGG05, DBCM04a]. Such spatial filters are inscreasingly used in the BCI field as they were shown to be quite efficient. The remaining of this section describes the main preprocessing methods used for BCI design.

1.4.1 Simple spatial and temporal filters

Most BCI systems use simple spatial or temporal filters as preprocessing in order to increase the signal-to-noise ratio of EEG signals.

1.4.1.1 Temporal filters

Temporal filters such as low-pass or band-pass filters are generally used in order to restrict the analysis to frequency bands in which we know that the neurophysiological signals are. For instance, BCI based on sensorimotor rhythms generally band-pass filter the data in the 8-30 Hz frequency band, as this band contains both the μ and β rhythms, i.e., the sensorimotor rhythms [RMGP00]. This temporal filter can also remove various undesired effects such as slow variations of the EEG signal (which can be due, for instance, to electrode polarization) or power-line interference (50 Hz in France). Hence, temporal filters make it possible to reduce the influence of frequencies that are lying outside of the frequential regions of interest. Such a filtering is generally achieved using Discrete Fourier Transform (DFT) or using Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters.

Direct Fourier Transform filtering: DFT makes it possible to visualize a signal into the frequency domain, i.e., to see a signal as a sum of oscillations at different frequencies f. Thus, the DFT $S(f)$ of a signal $s(n)$, which is composed of N samples, can be defined as follows:

$$S(f) = \sum_{n=0}^{N-1} s(n) e^{\frac{-2i\pi fn}{N}} \tag{1.1}$$

Thus, filtering a signal using DFT simply consists in setting to 0 all coefficients $S(f)$ which do not correspond to targeted frequencies, and then to transform the signal back into the time domain, by using the inverse DFT:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{\frac{2i\pi nk}{N}} \tag{1.2}$$

When performing DFT filtering, a windowing step should be performed before applying DFT [Smi97]. DFT filtering can be used online and in real-time, thanks to the efficient and popular DFT implementation known as the Fast Fourier Transform (FFT) [Smi97]. As an

exemple, DFT filtering has been used for the classification of finger movement intention in several BCI [BCM02, KBCM07].

Filtering with Finite Impulse Response filters: FIR filters are linear filters which make use of the M last samples of a raw signal $s(n)$ in order to determine the filtered signal $y(n)$:

$$y(n) = \sum_{k=0}^{M} a_k s(n-k) \tag{1.3}$$

where the a_k are the filter coefficients, which values depend on the desired filter to be used [Smi97]. FIR filters are know to have excellent performances in the frequential domain. For instance, FIR filters have been used for BCI based on motor imagery [DBK+06] or on SSEP [GPAR+07b].

Filtering with Infinite Impulse Response filters: As FIR filters, IIR filters are linear filters. On the other hand, IIR filters are recursive filters, which means that, in addition to the M last samples, they make use of the outputs of the P last filterings:

$$y(n) = \sum_{k=0}^{M} a_k s(n-k) + \sum_{k=1}^{P} b_k y(n-k) \tag{1.4}$$

In this way, IIR filters can perform filtering with a much smaller number of coefficients than FIR filters. However, their performances in the frequential domain is slightly reduced [Smi97]. Among the IIR filters employed for EEG preprocessing in BCI, we can quote Butterworth, Tchebychev or elliptic filters [Smi97, MBC07, DBCM04a].

1.4.1.2 Spatial filters

Similarly to temporal filters, various simple spatial filters are used in order to isolate the relevant spatial information embedded in the signals. This is achieved by selecting or weighting the contributions from the different electrodes (and as such from the different spatial regions) [MMDW97]. The most simple spatial filter consists in selecting the electrodes for which we know they are measuring the relevant brain signals, and ignoring other electrodes. Indeed, these latter electrodes are likely to measure mostly noise or background activity not relevant for the targeted BCI. As an example, when using a BCI based on hand motor imagery, it is known that the neurophysiological signals of interest are mainly located over the motor or sensorimotor cortex areas [PN01, PK92]. Thus, it is interesting to focus on electrodes C3 and C4, which are located over the left and right motor cortex respectively (see Figure 1.2), or even to use only these electrodes [PNM+03]. Similarly, for BCI based on SSVEP, the most relevant electrodes are the electrodes O1 and O2, which are located over the visual areas [LKF+05a].

Other simple and popular spatial filters are the Common Average Reference (CAR) and the Surface Laplacian (SL) filters [MMDW97]. These two filters make it possible to reduce the background activity [MMDW97]. The CAR filter is obtained as follows:

$$\hat{V}_i = V_i - \frac{1}{N_e}\sum_{j=0}^{N_e} V_j \tag{1.5}$$

where $\hat{V}_i$ and V_i are the i^{th} electrode potential, after and before filtering respectively, and N_e is the number of electrodes used. Thus, with the CAR filter, each electrode is re-referenced according the average potential over all electrodes. The SL filter can be obtained as follows:

$$\hat{V}_i = V_i - \frac{1}{4}\sum_{j\in\Omega_i^4} V_j \tag{1.6}$$

where Ω_i^4 is the set of the 4 neighboring electrodes of electrode i. Thus, this filter can reduce localy the background activity. It should be noted that more advanced versions of this filter can be used, notably versions based on spline approximations [PBP87].

Naturally, numerous other preprocessing methods, which are more complex and more advanced, have been proposed and used. In the following, we describe two of the most popular methods, namely, independant component analysis and common spatial patterns. Then we evoke some other existing methods and notably methods known as inverse solutions.

1.4.2 Independant component analysis and blind source separation

Blind Source Separation (BSS) is a family of methods which are used to solve "cocktail party" like problems [Sto05, JH91]. Independent Component Analysis (ICA) is probably the best known member of this BSS family [HO00]. In a cocktail party problem, the measured signals m (recorded using several sensors) are resulting from an unknown linear mixing of several sources s. In a mathematical form, it reads:

$$m = As \tag{1.7}$$

where m is the matrix of measurements, with a sensor per row and a time sample per column; s is the source matrix, with a source per row and a time sample per column; and A is the unknown mixing matrix which represents the linear mixing. Performing BSS consists in determining an estimate $\hat{s}$ of s without knowing A [JH91]:

$$\hat{s} = Wm \tag{1.8}$$

where W is the demixing matrix. Typically we have $W = A^{-1}$, the problem being that A is unknown. To tackle this probem, ICA assumes that the sources s (also known as components) are statistically independent, which has been revealed as being a reasonable hypothesis for numerous problems [HO00, Sto05]. Numerous ICA algorithms have been proposed and proved to be useful, especially for EEG signal processing [DM04] and BCI design [ZL06, NBL$^+$06, MEJS00, KASC08, HHB$^+$03, QLC05, HdS07, EE04]. Indeed, EEG signals are resulting from the mixing of different signals coming from different brain regions. As such, ICA may unmix these signals and isolate the signals coming from different brain regions, representing different brain rhythms, or even separate artifacts from real brain signals. In this way,

it becomes possible to keep only the components corresponding to signals of interest and/or to remove components that are very likely to represent noise and/or artifacts. Then, the EEG signals can be reconstructed using only the selected components. This has been shown to increase the signal-to-noise ratio and as such the performances of the whole BCI [QLC05].

1.4.3 Common Spatial Patterns

Another spatial filtering method which is increasingly used for preprocessing in BCI, and has proved to be very efficient is the Common Spatial Patterns (CSP) method [RMGP00, WGG05, BB04, CLL06, MGPF99, PFB$^+$07, BMK$^+$06, BDK$^+$07, DBCM04b]. This method is based on the decomposition of the EEG signals into spatial patterns [RMGP00, DBCM04b, WGG05]. These patterns are selected in order to maximize the differences between the classes involved once the data have been projected onto these patterns. Determining these patterns is performed using a joint diagonalization of the covariance matrices of the EEG signals from each class [RMGP00, DBCM04b]. These filters have proved to be very efficient, especially during BCI competitions [SGM$^+$03, BMC$^+$04, BMK$^+$06]. During these competitions, various data sets were proposed to the participants, with the aim of evaluating the different pattern recognition algorithms for BCI. The goal of the participants was first to calibrate their algorithms using a data set known as the training set, in which EEG signals were labelled with their corresponding class. Then, the participants had to use their tuned algorithms in order to determine the classes of signals contained in a data set known as the testing set, in which the signals were unlabelled. The use of CSP have grown quickly during the different competitions, until they enabled several groups to win during the last competition, in 2005, on several data sets [BMK$^+$06]. Currently, CSP are used in the design of numerous BCI [RMGP00, WGG05, BB04, CLL06, MGPF99, PFB$^+$07, BMK$^+$06, BDK$^+$07, DBCM04b].

1.4.4 Inverse solutions

Relevant but much less used preprocessing methods for BCI are inverse solutions. Inverse solutions are methods that attempt to reconstruct the activity in the brain volume by using only scalp measurements and a head model [MML$^+$04, BML01]. When using EEG, the signals $m(t)$ ($m \in \Re^{N_e,1}$ with N_e being the number of electrodes used) recorded at time t on the scalp can be modeled by a linear combination of brain dipole activity $c(t)$ ($c \in \Re^{3*N_v,1}$ with N_v being the number of dipoles considered). This is called the *forward problem* [BML01]:

$$m(t) = Kc(t) \tag{1.9}$$

where K is a $N_e * (3 * N_v)$ matrix called the *leadfield* matrix which represents the physical properties (conduction) of the head. More precisely, this matrix is a head model in which each of one the N_v dipoles is modeled by a volume element called *voxel* (typically thousands of voxels are considered). The $c(t)$ vector holds the orientation and amplitude of each dipole, according to the three dimensions of the head model space. Inverse solutions aim at estimating the brain dipole activity $\hat{c}(t)$ by using only the scalp measurements $m(t)$ and the leadfield matrix (head model) K:

$$\hat{c}(t) = Tm(t) \tag{1.10}$$

where T is the generalized inverse of K. As $N_v >> N_e$, this problem has no unique solution and additional constraints must be added to solve it. Depending on the constraints used, different inverse solutions are obtained which leads to different T matrices [MML$^+$04]. Inverse solutions estimate the amplitude and/or the orientation of the dipoles.

There are two main kinds of inverse solutions: distributed solutions and equivalent dipole solutions [MML$^+$04, BML01]. Distributed solutions estimate the amplitudes and orientations of a large number of voxels distributed in all the cortex or in all the brain whereas equivalent dipole solutions estimate the position, amplitude and orientation of a few sources (typically one or two), each one modeled by an equivalent dipole.

From the point of view of BCI, inverse solutions give access to new information, i.e., to the activity in the brain volume. As this information has a strong physiological basis, it appears as a new and attractive method. Recently, a few studies have started to evaluate the efficiency of inverse solutions for BCI and have obtained promising first results [LLA07b, NKM08, KLH05]. In order to design BCI, inverse solutions are generally used in two different ways:

- As a preprocessing method which precedes feature extraction. In this case, the inverse solution is used to estimate $c(t)$ from which the features are extracted [GGP$^+$05, BCM$^+$07, NKM08].
- As a direct feature extraction technique. In this case, either the brain current density values, reconstructed in a number of Regions Of Interest (ROI) [CLL06] or the positions of the sources [QDH04, KLH05] are used directly as features so as to identify the mental tasks performed.

These methods have obtained very satisfying results, generally as good or even better than those in the literature. Moreover, it has been observed that extracting features from $c(t)$ (the source domain) would be more efficient than extracting them directly from $m(t)$ (the sensor domain) [GGP$^+$05, NKM08]. A possible interpretation is that the inverse solution acts as a spatial filter that removes the background activity and the noise not correlated with the targeted mental tasks.

1.4.5 Other methods

Numerous other preprocessing methods have been proposed and used for BCI design. Among these methods, we can quote various spatial filters such as invariant CSP [BKT$^+$08], Principal Component Analysis (PCA) [Smi02, LC03, TGW06] or Common Subspace Spatial Decomposition (CSSD) [WZL$^+$04, ZWG$^+$07] as well as numerous spectro-spatial filters [DBK$^+$06, LBCM05, TDAM06]. In addition to filtering methods, other relatively simple methods are used as preprocessing, such as moving average filtering, subsampling (in order to reduce the dimensionality of the problem) [KMG$^+$04, RGMA05b] or baseline correction [KCVP07]. Baseline correction consists in subtracting to the signals, or to their spectrum, an average amplitude level, estimated on a reference period. This aims at reducing the effects of the non-stationarity of EEG signals.

1.4.6 Conclusion

As higlighted in this section, numerous preprocessing methods have been used for BCI. However, no preprocessing method or combination of preprocessing methods have been identified as the best, due to a lack of comparisons. Nevertheless, studies performed so far have all highlighted the need to do preprocessing in order to improve the performance of the resulting BCI [HdS07]. More particularly, spatial filters and related methods have been shown to reduce the noise and dramatically improve the performance [BMK+06]. As such, if working with a sufficiently high number of electrodes, the use of spatial filters is now highly recommended.

1.5 Feature extraction

Measuring brain activity through EEG leads to the acquisition of a large amount of data. Indeed, EEG signals are generally recorded with a number of electrodes varying from 1 to 256 and with a sampling frequency varying from 100 Hz to 1000 Hz. In order to obtain the best possible performances, it is necessary to work with a smaller number of values which describe some relevant properties of the signals. These values are known as “features”. Such features can be, for instance, the power of the EEG signals in different frequency bands. Features are generally aggregated into a vector known as “feature vector”. Thus, feature extraction can be defined as an operation which transforms one or several signals into a feature vector.

Identifying and extracting good features from signals is a crucial step in the design of BCI. Indeed, if the features extracted from EEG are not relevant and do not describe well the neurophysiological signals employed, the classification algorithm which will use such features will have trouble identifying the class of these features, i.e., the mental state of the user. Consequently, the correct recognition rates of mental states will be very low, which will make the use of the interface not convenient or even impossible for the user. Thus, even if it is sometimes possible to use raw signals as the input of the classification algorithm (see section 1.6), it is recommended to select and extract good features in order to maximize the performances of the system by making easier the task of the subsequent classification algorithm. According to some researchers, it seems that the choice of a good preprocessing and feature extraction method have more impact on the final performances than the choice of a good classification algorithm [PFK93, HdS07].

Numerous feature extraction techniques have been studied and proposed for BCI [BFWB07, MAM+06]. These techniques can be divided in three main groups, which are: 1) the methods that exploit the temporal information embedded in the signals [SLP97, PR99a, ASS98, KMG+04, RGMA05b], 2) the methods that exploit the frequential information [PN01, Pal05, dRMMC+00, RTNS06, BGM07a] and 3) hybrid methods, based on time-frequency representations, which exploit both the temporal and frequential information [FBWB04, Bos04, WDH04]. A fourth category could have been added here, the category of methods that exploit the spatial information. However, this category would be limited to the use of inverse solutions which have already been described in the previous section. Indeed, the spatial information is generally used to perform a spatial filtering before extracting features based on the temporal and/or frequential information [BGM07a, MMDW97]. Thus, we only describe here the first three kinds of methods, as well as some marginal methods which do not fit into these main

categories.

1.5.1 Temporal methods

Temporal methods use as features the temporal variations of the signals. These methods are particularly adapted to describe neurophysiological signals with a precise and specific time signature, such as the P300 [KMG+04, RGMA05b, RS07a] or ERD, notably those triggered by motor imagery [OGNP01, SLP97]. Among these temporal feature extraction methods, we can find the amplitude of raw EEG signals, auto-regressive parameters or Hjorth parameters.

1.5.1.1 Signal amplitude

The most simple (but still efficient) temporal information that could be extracted is the time course of the EEG signal amplitude. Thus, the raw amplitudes of the signals from the different electrodes, possibly preprocessed, are simply concatenated into a feature vector before being passed as input to a classification algorithm. In such a case, the amount of data used is generally reduced by preprocessing methods such as spatial filtering or subsampling. This kind of feature extraction is one of the most used for the classification of P300 [RGMA05b, HGV+05, KMG+04, RS07b].

1.5.1.2 Autoregressive parameters

AutoRegressive (AR) methods assume that a signal $X(t)$, measured at time t, can be modeled as a weighted sum of the values of this signal at previous time steps, to which we can add a noise term E_t (generally a Gaussian white noise):

$$X(t) = a_1 X(t-1) + a_2 X(t-2) + \ldots + a_p X(t-p) + E_t \tag{1.11}$$

where the weights a_i are the autoregressive parameters which are generally used as features for BCI [AS96, GPAT03] and p is the model order. Several variants of autoregressive parameters have also been used such as multivariate AR parameters [ASS98], AR parameters with exogenous input [BKd+05] or Adaptive AR (AAR) parameters [HP04a, HP04b, SLP97, PNSL98]. AAR parameters assume that the weights a_i can vary over time, and are the most used variant of AR parameters. It seems that AAR parameters would give better results than AR parameters for motor imagery classification [SLP97, PNSL98], whereas they would give worse results for the classification of cognitive tasks such as mental computations, mental rotation of a geometric figure, etc. [HP04a, HP04b]. It should be noted that it is possible to derive a frequential information from the a_i coefficients [MW05].

1.5.1.3 Hjorth parameters

Hjorth parameters describe the temporal dynamics of a signal $X(t)$, by using three measures that are the activity, the mobility and the complexity [OGNP01]:

$$Activity(X(t)) = VAR(X(t)) \tag{1.12}$$

$$Mobility(X(t)) = \sqrt{\frac{Activity(\frac{dX(t)}{dt})}{Activity(X(t))}} \tag{1.13}$$

$$Complexity = \frac{Mobility(\frac{dX(t)}{dt})}{Mobility(X(t))} \tag{1.14}$$

Such features are mainly used for the classification of motor imagery [OGNP01, BM04, LC03, PN01].

1.5.2 Frequential methods

As mentioned in section 1.3.1.3, EEG signals are composed by a set of specific oscillations known as rhythms. Performing a given mental task (such as motor imagery or another cognitive task) makes the amplitude of these different rhythms vary. Moreover, signals such as steady state evoked potentials are defined by oscillations with frequencies synchronized with the stimulus frequency. Consequently, it appears as natural or even essential to exploit the frequential information embedded in the EEG signals. To this end, two main techniques, which are closely related, are used: band power features and power spectral density features.

1.5.2.1 Band power features

Computing a band power feature consists in band-pass filtering a signal in a given frequency band, then in squaring the filtered signal and finally in averaging the obtained values over a given time window [PN01, BGMP07]. It is also possible to log-transform this value in order to have features with a distribution close to the normal distribution [PN01]. Band power features are generally computed for several frequency bands previously determined according to the mental states to be recognized. Such features have been notably used with success for motor imagery classification [PN01, SLS+08, ZLGL08, LLLA07] but also for the classification of cognitive processing tasks [Pal05].

1.5.2.2 Power spectral density features

Power Spectral Density (PSD) features, sometimes simply called spectrum, inform on the distribution of the power of a signal between the different frequencies. PSD features can be computed, for instance, by squaring the Fourier transform of a signal [LKF+05a] or by computing the Fourier transform of the autocorrelation function of this signal [KA90]. PSD features are probably the most used features for BCI, and have proved to be efficient for recognizing a large number of neurophysiological signals [BFdM04, KA90, LKF+05a, BGM07b, dRMFM+02, MM03].

1.5.3 Time-frequency representations

Feature extraction methods that have been presented so far in this manuscript are probably the most used. However, and considering that neurophysiological signals used in a BCI have generally specific properties in both the temporal and frequential domain, other methods, which

can be seen as hybrid, have also been used to design BCI. These methods are based on various time-frequency representations such as the short-time Fourier transform or wavelets, and extract from the signals information that are both frequential and temporal. The main advantage of these time-frequency representations is that they can catch relatively sudden temporal variations of the signals, while still keeping frequential informations. On the contrary, pure frequential methods are assuming that the signal is in a stationary state.

1.5.3.1 Short-time Fourier transform

Short-Time Fourier Transform (STFT) simply consists in first multiplying the input signal by a given windowing function w which is non-zero only over a short time period, and then in computing the Fourier transform of this windowed signal. In discrete time, the STFT $X(n,w)$ of a signal $x(n)$ is as follows:

$$X(n,\omega) = \sum_{n=-\infty}^{+\infty} x(n)w(n)e^{-j\omega n} \tag{1.15}$$

The Time-Frequency (TF) representation is obtained by computing this Fourier transform along a sliding window, i.e., for different segments with a given level of overlapping. This method has been successfully used in several BCI studies [CPM05, HPM05]. Its main drawback is the use of an analysis window with a fixed size, which leads to a similar frequential and temporal resolution in all frequency bands. For instance, it would be more interesting to have a high temporal resolution for high frequencies which describe a fine scale. Wavelet analysis aims at overcoming this drawback.

1.5.3.2 Wavelets

Similarly to Fourier transform, wavelet transform decomposes a signal onto a basis of functions [SBRS99]. This basis of functions is a set of wavelets $\Phi_{a,b}$, each one being a scaled and translated version of the same wavelet Φ known as the mother wavelet:

$$\Phi_{a,b}(t) = \frac{1}{\sqrt{(a)}}\Phi(\frac{t-b}{a}) \tag{1.16}$$

The wavelet transform $W_x(s,u)$ of a signal x can be written as follows:

$$W_x(s,u) = \int_{-\infty}^{+\infty} x(t)\Phi_{u,s}(t)dt \tag{1.17}$$

where s and u are respectively the scaling and translating factor. The advantage of wavelets is that they make it possible to analyze the signal at different scales simultaneously. Moreover, the resolution depends on the scale. As such, high frequencies, which correspond to a fine scale, can be analyzed with a high temporal resolution whereas low frequencies, which correspond to a coarse scale, can be analyzed with a high frequential resolution. These points make wavelets a very interesting tool for analyzing EEG signals [SBRS99]. Various kinds of wavelets have been used for BCI, such as Daubechies wavelets [VHMM00, HdS07], Coiflet wavelets [YHS05],

Morlet wavelets [LSC04], bi-scale wavelets [MB00] or Mexican hat wavelets [Bos04]. They all made it possible to reach very promising results.

1.5.3.3 Other time-frequency representations

In addition to STFT and wavelet transform, a number of TF representations have been used for BCI. Such representations are generally based on different basis of functions and use different levels of adaptivity with respect to time and frequency. Among these methods, we can mention TF representations based on Wigner-Ville distributions [GEV03b], adaptive Gaussian representations [CJ00] or TF representations with Q-constant frequency decomposition [WDH04].

1.5.4 Other feature extraction methods

Other feature extraction methods have been used to design BCI, in a more marginal way. Among these methods, it is worth mentioning methods based on interactions between signals. Thus, measuring the coherency or phase synchronization between sensors has proved to be efficient for EEG feature extraction in BCI [GC04]. Similarly, describing the EEG signals thanks to brain connection graphs made it possible to discriminate different brain states [GPAR$^+$07b, GPAR$^+$07a]. Still exploiting the interactions between sensors, the fractal dimension of signals [BM04, BGMP07] or their multi-fractal spectrum [Bro08] have been used as features for BCI. Finally, several works have shown that using together features extracted using different methods could lead to increased performances [DBCM04a, GC04, BGMP07]. It is also interesting and efficient to create novel kinds of features, for instance by mixing existing features using genetic algorithms [BGMP07].

1.5.5 Feature selection and dimensionality reduction

BCI feature vectors are often of very high dimensionality (see for instance [RGMA05b]). Indeed, several features are generally extracted from several EEG channels (electrodes) and from several time segments, before being concatenated into a single feature vector. Moreovoer, the training sets, i.e., the example data for each class, are generally small, as the training process is time consuming and relatively uncomfortable for subjects. Consequently, BCI are often affected by a problem known as “curse-of-dimensionality”. This problem comes from the fact that the amount of data required to describe properly the different classes increases exponentially with the dimensionality of the feature vector. [JDM00, Fri97]. If the number of training data is small relatively to the number of features, the classification algorithm which will use these features and data will very likely give bad results. It is recommended to use at least 5 to 10 times more training data per class than the number of features [RJ91, JC82].

In order to tackle this problem, it is often necessary to use dimensionality reduction methods such as principal component analysis [BMBB04], or to use various feature selection methods [dRMFM$^+$02], among which genetic algorithms are the most popularly employed [GPAT03, PKK$^+$05, ETI02]. These different methods make it possible to work with a set of features with a much smaller size than the original set which generally leads to better performances.

1.5.6 Conclusion

Again, although numerous feature extraction methods have been proposed for BCI, it is very difficult to identify the most efficient ones due to a lack of comparisons. It seems also important to extract a small number of features which represents subject-specific information, in order to reach good performances. As such, it seems interesting to use features that can be tuned (e.g., band power features for which the frequency bands can be adapted to the subject) as well as dimensionality reduction or feature selection techniques in order to facilitate the subsequent work of the classifier. It is important to note that, even if a relatively large number of feature extraction techniques have been proposed, it is admitted by the BCI community that it is necessary to explore and study new feature extraction methods and concepts [MAM+06]. More precisely, it is important to find features which will lead to more efficient BCI, in terms of correct recognition rates, and to more interpretable BCI in order to have more insights on the mental processes employed by the BCI users to control the system [MAM+06].

1.6 Classification

The third key step for identifying neurophysiological signals in a BCI is translating the features into commands [MAM+06, MB03]. In order to achieve this step, one can use either regression algorithms [MW05, DHS01] or classification algorithms [PRCS00, LCL+07], the classification algorithms being by far the most used in the BCI community [BFWB07, LCL+07]. As such, in this section, we focus only on the classification algorithms.

The goal of the classification step is to assign automatically a class to the feature vector previously extracted. This class represents the kind of mental task performed by the BCI user. Classification is achieved using algorithms known as “classifiers”. Classifiers are able to learn how to identify the class of a feature vector, thanks to training sets. These sets are composed of feature vectors labeled with their class of belonging.

In this section, we first present a taxonomy of the different classification algorithms, and then the main classifier families that are used in the BCI field. These classifiers can be divided into five main categories which are: linear classifiers, neural networks, non linear bayesian classifiers, nearest neighbor classifiers and classifier combinations.

1.6.1 Classifier taxonomy

Several properties are commonly used to describe the different kinds of available classifiers:

Generative-discriminative:
Generative (also known as informative) classifiers, e.g., Bayes quadratic, learn the class models. To classify a feature vector, generative classifiers compute the likelihood of each class and choose the most likely. Discriminative ones, e.g., Support Vector Machines, only learn the way of discriminating the classes or the class membership in order to classify a feature vector directly [NJ02] [RH97];

Static-dynamic:
Static classifiers, e.g., MultiLayer Perceptrons, cannot take into account temporal infor-

mation during classification as they classify a single feature vector. On the contrary, dynamic classifiers, e.g., Hidden Markov Model, can classify a sequence of feature vectors and thus, catch temporal dynamics [Rab89].

Stable-unstable:
Stable classifiers, e.g., Linear Discriminant Analysis, have a low complexity (or capacity [Vap99]). They are said stable as small variations in the training set do not affect considerably their performances. On the contrary, unstable classifiers, e.g., MultiLayer Perceptron, have a high complexity. As for them, small variations of the training set may lead to important changes in performances [Bre98].

Regularized:
Regularization consists in carefully controlling the complexity of a classifier in order to prevent overtraining. A regularized classifier has good generalization performances and is more robust with respect to outliers [DHS01] [JDM00].

1.6.2 Linear classifiers

Linear classifiers are discriminant algorithms that use linear functions to distinguish classes. They are probably the most popular algorithms for BCI applications. Two main kinds of linear classifiers have been used for BCI design, namely, Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM).

1.6.2.1 Linear Discriminant Analysis

The aim of LDA (also known as Fisher's LDA) is to use hyperplanes to separate the data representing the different classes [DHS01] [Fuk90]. For a two-class problem, the class of a feature vector depends on which side of the hyperplane the vector is (see Figure 1.10).

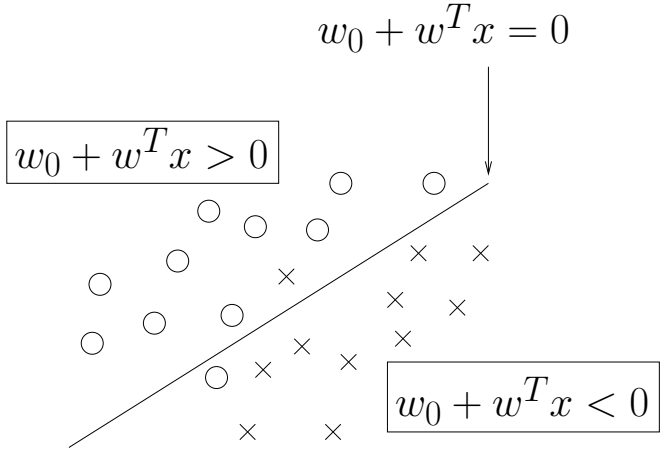


Figure 1.10: A hyperplane which separates two classes: the "circles" and the "crosses".

LDA assumes a normal distribution of the data, with equal covariance matrices for both classes. The separating hyperplane is obtained by seeking the projection that maximizes the distance between the two classes means and minimizes the interclass variance [Fuk90]. To solve an N-class problem ($N > 2$) several hyperplanes are used. The strategy generally used for multiclass BCI is the "One Versus the Rest" (OVR) strategy which consists in separating each class from all the others.

This technique has a very low computational requirement which makes it suitable for online BCI systems. Moreover this classifier is simple to use and generally provides good results. Consequently, LDA has been used with success in a great number of BCI systems such as motor imagery based BCI [Pfu99], P300 speller [Bos04], multiclass [GPAT03] or asynchronous [SMN+04] BCI. The main drawback of LDA is its linearity that can provide poor results on complex nonlinear EEG data [GEV03a].

A Regularized Fisher's LDA (RFLDA) has also been used in the field of BCI [BCM02, MKD+04]. This classifier introduces a regularization parameter C that can allow or penalize classification errors on the training set. The resulting classifier can accomodate outliers and obtain better generalization capabilities. As outliers are common in EEG data, this regularized version of LDA may give better results for BCI than the non-regularized version [MKD+04, BCM02]. Surprisingly, RFLDA is much less used than LDA for BCI applications.

1.6.2.2 Support Vector Machine

An SVM also uses a discriminant hyperplane to identify classes [Bur98, BC00]. However, concerning SVM, the selected hyperplane is the one that maximizes the margins, i.e., the distance from the nearest training points (see Figure 1.11). Maximizing the margins is known to increase the generalization capabilites [Bur98, BC00]. As RFLDA, an SVM uses a regularization parameter C that enables accomodation to outliers and allows errors on the training set.

Such an SVM enables classification using linear decision boundaries, and is known as linear SVM. This classifier has been applied, always with success, to a relatively large number of synchronous BCI problems [BCM02, GPAT03, RGMA05b, RG08]. However, it is possible to create nonlinear decision boundaries, with only a low increase of the classifier's complexity, by using the "kernel trick". It consists in implicitly mapping the data to another space, generally of much higher dimensionality, using a kernel function $K(x,y)$. The kernel generally used in BCI research is the Gaussian or Radial Basis Function (RBF) kernel:

$$K(x,y) = exp(\frac{-||x-y||^2}{2\sigma^2}) \tag{1.18}$$

The corresponding SVM is known as Gaussian SVM or RBF SVM [Bur98, BC00]. RBF SVM have also given very good results for BCI applications [KMG+04, GPAT03]. As LDA, SVM have been applied to multiclass BCI problems by combining together multiple two-class SVM [SLBP05, GEV03a].

SVM have several advantages. Actually, thanks to the margin maximization and the regularization term, SVM are known to have good generalization properties [BC00, JDM00], to be insensitive to overtraining [JDM00] and to the curse-of-dimensionality [Bur98, BC00]. Finally,

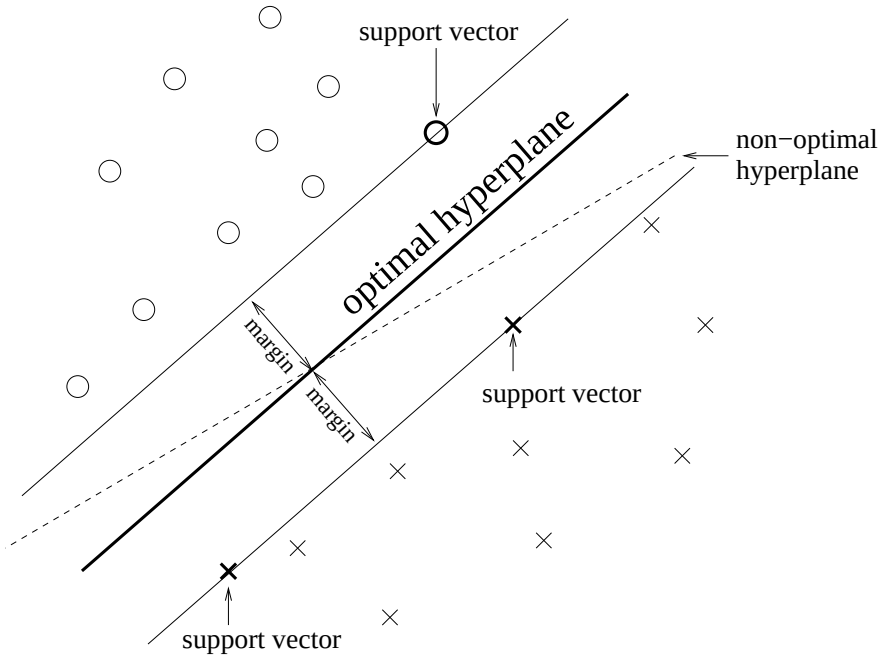


Figure 1.11: SVM find the optimal hyperplane for generalization.

SVM have a few hyperparameters that need to be defined by hand, namely, the regularization parameter C and the RBF width σ if using the kernel of Equation 1.18. These advantages are gained at the expense of a low speed of execution.

1.6.3 Neural Networks

Neural Networks (NN) are, together with linear classifiers, the category of classifiers mostly used in BCI research (see, e.g., [HST90, AS96]). Let us recall that a NN is an assembly of several artificial neurons which enables to produce nonlinear decision boundaries [Bis96]. This section first describes the most widely used NN for BCI, which is the MultiLayer Perceptron (MLP). Then, we briefly present other architectures of neural networks used for BCI applications.

1.6.3.1 MultiLayer Perceptron

An MLP is composed of several layers of neurons: an input layer, possibly one or several hidden layers, and an output layer [Bis96]. Each neuron's input is connected with the output of the previous layer's neurons whereas the neurons of the output layer determine the class of the input feature vector (see Figure 1.12).

Neural Networks and thus MLP, are universal approximators, i.e., when composed of enough neurons and layers, they can approximate any continuous function. Added to the fact that they can classify any number of classes, this makes NN very flexible classifiers that can adapt to a great variety of problems. Consequently, MLP, which are the most popular NN used

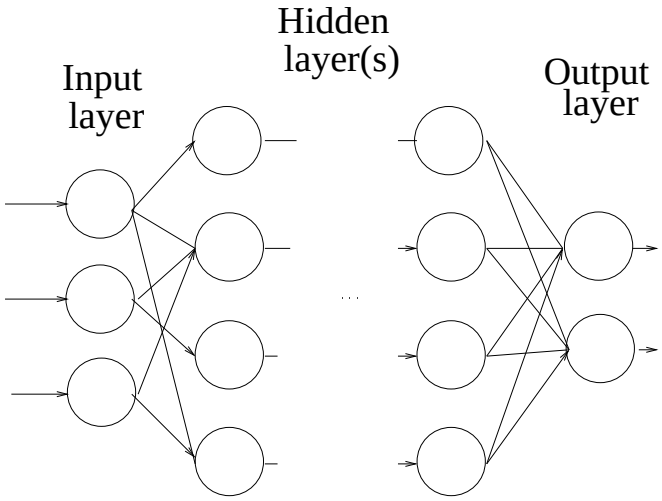


Figure 1.12: The architecture of a MultiLayer Perceptron, composed of an input layer of neurons, a number of hidden layers and an output layer.

in classification, have been applied to almost all BCI problems such as binary [Pal05] or multiclass [AS96], synchronous [HP00] or asynchronous [CB04] BCI. However, the fact that MLP are universal approximators makes these classifiers sensitive to overtraining, especially with such noisy and non-stationary data as EEG (e.g., see [BP05]). Therefore, careful architecture selection and regularization is required [JDM00].

A MultiLayer Perceptron without hidden layers is known as a perceptron. Interestingly enough, a perceptron is equivalent to LDA and, as such, has been sometimes used for BCI applications [WZL$^+$04]

1.6.3.2 Other Neural Network architectures

Other types of NN architectures are used in the field of BCI. Among them, one deserves a specific attention as it has been specifically created for BCI: the Gaussian classifier [dRMMC$^+$00, dRMRMG04]. Each unit of this NN is a Gaussian discriminant function representing a class prototype. According to its authors, this NN outperforms MLP on BCI data and can perform efficient rejection of uncertain samples [dRMMC$^+$00]. As a consequence, this classifier has been applied with success to motor imagery [SM04] and mental task classification [dRMMC$^+$00], particularly during asynchronous experiments [dRMMC$^+$00, CST$^+$03].

Besides the Gaussian classifier, several other NN have been applied to BCI purposes, in a more marginal way:

- Learning Vector Quantization (LVQ) Neural Network [Koh90, PFK93]
- Fuzzy ARTMAP Neural Network [CGN$^+$92, PPNS02];
- Dynamic Neural Networks such as the Finite Impulse Response Neural Network (FIRNN) [HP00], the Time-Delay Neural Network (TDNN) or the Gamma Dynamic Neural Network (GDNN) [BTV96];

- RBF Neural Network [DHS01, HHB+03];
- Bayesian Logistic Regression Neural Network (BLRNN) [PRCS00];
- Adaptive Logic Network (ALN) [KP00];
- Probability estimating Guarded Neural Classifier (PeGNC) [FF03].

1.6.4 Nonlinear Bayesian classifiers

This section introduces two Bayesian classifiers used for BCI: Bayes quadratic and Hidden Markov Model (HMM). Although Bayesian Graphical Network (BGN) has been employed for BCI, it is not described here as it is not common and, currently, not fast enough for real-time BCI [TR04, RTNS06].

All these classifiers produce nonlinear decision boundaries. Furthermore, they are generative, which enables them to perform more efficient rejection of uncertain samples than discriminative classifiers. However, these classifiers are not as widespread as linear classifiers or Neural Networks in BCI applications.

1.6.4.1 Bayes quadratic

Bayesian classification aims at assigning to a feature vector the class it belongs to with the highest probability [DHS01, Fuk90]. The Bayes rule is used to compute the so-called *a posteriori* probability that a feature vector has of belonging to a given class [Fuk90]. Using the Maximum A Posteriori (MAP) rule and these probabilities, the class of this feature vector can be estimated. Bayes quadratic consists in assuming a different normal distribution of data. This leads to quadratic decision boundaries, which explains the name of the classifier. Even though this classifier is not widely used for BCI, it has been applied with success to motor imagery [LSC04, SM04] and mental task classification [KA90, BFdM04].

1.6.4.2 Hidden Markov Model

Hidden Markov Models (HMM) are popular dynamic classifiers in the field of speech recognition [Rab89]. An HMM is a kind of probabilistic automaton that can provide the probability of observing a given sequence of feature vectors [Rab89]. Each state of the automaton can modelize the probability of observing a given feature vector. For BCI, these probabilities usually are Gaussian Mixture Models (GMM) [OGNP01].

HMM are perfectly suitable algorithms for the classification of time series [Rab89]. As EEG components used to drive BCI have specific time courses, HMM have been applied to the classification of temporal sequences of BCI features [OGNP01] [CST+03, ONGP00] and even to the classification of raw EEG [SNG05]. HMM are not much widespread within the BCI community but these studies revealed that they were promising classifiers for BCI systems.

Another kind of HMM which has been used to design BCI is the Input-Output HMM (IOHMM) [CB04]. Contrary to HMM, IOHMM is not generative but discriminative. The main advantage of this classifier is that one IOHMM can discriminate several classes, whereas one HMM per class is needed to achieve the same operation.

1.6.5 Nearest Neighbor classifiers

The classifiers presented in this section are relatively simple. They consist in assigning a feature vector to a class according to its nearest neighbor(s). This neighbor can be a feature vector from the training set as in the case of k Nearest Neighbors (kNN), or a class prototype as in Mahalanobis distance.

1.6.5.1 k Nearest Neighbors

The aim of this technique is to assign to an unseen point the dominant class among its k nearest neighbors within the training set [DHS01]. For BCI designs, these nearest neighbors are usually obtained using a metric distance [BCM02]. With a sufficiently high value of k and enough training samples, kNN can approximate any function which enables it to produce nonlinear decision boundaries.

kNN algorithms are not very popular in the BCI community, probably because they are known to be very sensitive to the curse-of-dimensionality [Fri97], which made them fail in several BCI experiments [SLBP05, BCM02, MKD$^+$04]. However, when used in BCI systems with low-dimensional feature vectors, kNN may prove to be efficient [BMBB04].

1.6.5.2 Mahalanobis distance

Mahalanobis distance based classifiers assume a Gaussian distribution $N(\mu_c, M_c)$ for each prototype of the class c. Then, a feature vector x is assigned to the class that corresponds to the nearest prototype, according to the so-called Mahalanobis distance $d_c(x)$ [CST$^+$03, BBS$^+$01]:

$$d_c(x) = \sqrt{(x-\mu_c)M_c^{-1}(x-\mu_c)^T} \tag{1.19}$$

This leads to a simple yet robust classifier, which even proved to be suitable for multiclass [SLBP05] or asynchronous BCI systems [CST$^+$03]. Despite its good performances, it is still scarcely found in the BCI literature.

1.6.6 Combinations of classifiers

In most papers related to BCI, the classification is achieved using a single classifier. A recent trend, however, is to use several classifiers, aggregated in different ways. The classifier combination strategies used in BCI applications are the following:

1.6.6.1 Voting

When using Voting, several classifiers are being used, each of them assigning the input feature vector to a class. The final class will be that of the majority [JDM00]. Voting is the most popular way of combining classifiers in BCI research, probably because it is simple and efficient. For instance, Voting with LVQ NN [PFK93], MLP [QLC05], regularized logistic regression [HdS07], or SVM [RGMA05b, RG08, HdS07] have been attempted. It should be noted that the vote could focus on the class labels but also on the classifier outputs, as in [RGMA05b, RG08].

1.6.6.2 Boosting

Boosting consists in using several classifiers in cascade, each classifier focusing on the errors committed by the previous ones [DHS01]. It can build up a powerful classifier out of several weak ones, and it is unlikely to overtrain. Unfortunalty, it is sensible to mislabels [JDM00] which may explain why it was not succesfull in one BCI study [BM04]. To date, in the field of BCI, boosting has been experimented with MLP [BM04, BGMP07], kNN [Sun07], or other classifiers such as decision trees [Sun07] or Ordinary Least Square (OLS) [HGV$^+$05].

1.6.6.3 Stacking

Stacking consists in using several classifiers, each of them classifying the input feature vector. These classifier are called level-0 classifiers. The output of each of these classifiers is then given as input to a so-called meta-classifier (or level-1 classifier) which makes the final decision [Wol92]. Stacking has been used in BCI research using HMM as level-0 classifiers, and an SVM as the meta-classifier [LC03] or with SVM and regularized logistic regression as level 0 and level 1 classifiers respectively [HdS07].

1.6.6.4 Random subspaces

The random subspace technique consists in generating new training sets from the original one and in training a different classifier for each one of these new training sets [Sun07]. The final decision is made thanks to majority voting. When using random subspaces, the new training sets are generated by using only a subset of the features from the original training set, these features being randomly selected. The main advantage of such a method is that it enables to reduce the dimensionality while still using all the available features, through several classifiers. This method has been used for BCI with decision trees and kNN [Sun07].

The main advantage of classifier combination techniques is that a combination of similar classifiers is very likely to outperform one of the classifiers on its own. Actually, combining classifiers is known to reduce the Variance of the classification system and thus the classification error [Fri97, Bre98].

1.6.7 Conclusion

A great variety of classifiers has been tried in BCI research. Their properties are summarized in Table 1.1. A crucial classifier property to obtain good results in a BCI is noise/outlier resistance or noise/outlier accomodation. Indeed, regularized classifiers or classifiers able to accomodate outliers in the training data tend to give the best results in terms of correct classification rates [LCL$^+$07]. It also seems that for synchronous BCI, exploiting the time information by using dynamic classifiers is rewarding [LCL$^+$07]. However, when it comes to asynchronous (self-paced) BCI, dynamic classifiers are not better than other classifiers. Indeed, as the start of the mental task is unknown in asynchronous BCI, dynamic classifiers have trouble in exploiting efficiently the time information. Finally, it also seems that combining multiple classifiers leads to increased performance in comparison with using a single classifier [HdS07, RG08, Sun07].

In spite of all these studies on classifiers, it is admitted that new classifiers should be explored and studied for BCI design [MAM+06]. The goal of these explorations is to increase the correct classification rates on one hand, and to enable researchers to interpret what the classifiers have learnt on the other hand. It would also be interesting to be able to provide a priori knowledge on the brain dynamics to the classifiers.

Table 1.1: Properties of classifiers used in BCI research

	Linear	Non Linear	Gene-rative	Discri minant	Dynamic	Static	Regu-larized	Stable	Un-stable	High dimension robust
FLDA	X			X		X		X		
RFLDA	X			X		X	X	X		
linear-SVM	X			X		X	X	X		X
RBF-SVM		X		X		X	X	X		X
MLP		X		X		X			X	
BLR NN		X		X		X			X	
ALN NN		X		X		X			X	
TDNN		X		X	X				X	
FIRNN		X		X	X				X	
GDNN		X		X	X				X	
Gaussian NN		X	X			X			X	
LVQ NN		X		X		X			X	
Perceptron	X			X		X		X		
RBF-NN		X	X			X			X	
PeGNC		X		X		X	X		X	
fuzzy ARTMAP NN		X		X			X		X	
HMM		X	X		X				X	
IOHMM		X		X	X				X	
Bayes quadratic		X	X			X			X	
Bayes graphical network		X	X			X			X	
k-NN		X		X		X			X	
Mahalanobis distance		X	X			X			X	

1.7 Feedback and applications of BCI

Once the class of the signals have been identified, the system can associate a specific command to this identified mental state and send this command to a given application. These applications can be divided into two main categories. The first and most important category is the medical domain [WBM+02, KKK+01, KMHD06, KPL+07]. Indeed, the main objective of BCI is to provide severely disabled people with a new communication channel which is not based on the traditional motor output channels. The second category is the multimedia and virtual reality domain. Thus, even if BCI are mainly designed for disabled people, they can also be of interest for healthy persons [AGG07], for instance by proposing video games based on BCI

[KBCM07, MBBB04, MPSP07], or by exploiting the amazing processing skills of the brain to perform image search in large data bases [PCG+08]. Still in this category of BCI applications is the virtual reality domain, which is becoming increasingly promising [LSF+07, LLR+06, LLR+08]. Finally, BCI have also been used for controlling robots [LLA07c, Mil08], or for artistic creations [MB04, KHFH08]. We describe more in details some of these applications in the remaining of this section.

During this step of interaction with the application, it is particularly essential to provide a feedback to the subject, concerning the mental state that has been recognized by the system. Indeed, this feedback enables the user to know whether he has correctly performed the mental task which enables him to learn how to control his brain activity. A carefully chosen feedback can reduce the time required by the user to learn how to control the system [WBM+02].In most BCI applications, this feedback is a visual feedback [WBM+02], but BCI that provide an auditory feedback [HNP+04] or a haptic feedback [KPJ+06, CAR+07, CKA+07] have also been proposed. Unfortunately, there is a relatively small number of BCI papers dedicated to feedback. Consequently, it is currently relatively difficult to select the most appropriate feedback for a given application.

1.7.1 Rehabilitation applications and applications for the disabled people

We describe here some of the main existing BCI applications in the field of rehabilitation and handicap.

1.7.1.1 The “Thought Translation Device”

The “Thought Translation Device” (TTD) is one of the very first BCI that was designed. It has been developped at the university of Tuebingen in the team of Pr. Niels Birbaumer [BKG+00]. This BCI is based on the SCP signal. It aims at enabling paralyzed persons to spell words by selecting letters in a binary tree, thanks to spontaneaous variations of the SCP amplitude. Indeed, in this application, the alphabet has been recursively divided into two parts, and distributed on the leaves of a binary tree, according to these divisions. At each node of this binary tree, the user (mentally) selects the left or right subtree, according to the desired letter and the alphabet subset associated to each subtree. In order to do so, the subject should trigger a positive or negative variation of its SCP. This user has previously learnt to control his SCP through operant conditioning. This TTD system enabled disabled people to communicate at the speed of approximately one letter every 2 minutes [BKG+00].

1.7.1.2 The P300 speller

The P300 speller is a BCI application which uses the P300 signal to spell words, as suggested by its name. This application was initally designed in 1988 by Farwell and Donchin [FD88, DSW00]. In this application, the subject is seated in front of a screen on which a 6 by 6 matrix is displayed. This matrix contains all the letters of the latin alphabet, as well as the digits from 1 to 9 and the space character (see Figure 1.13).

In this application, a row or a column of the matrix is highlighted every 125 ms. The user is asked to look at and draw his attention on the letter he wants to select, and to count the

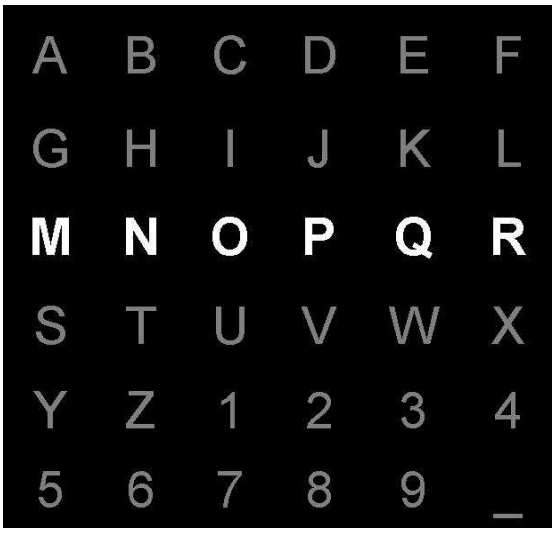


Figure 1.13: The P300 speller interface, as displayed on the user's screen [FD88].

number of times the desired letter is highlighted. The highlight of the desired letter being a rare and expected event, this triggers a P300 in the user's EEG signals. Detecting the absence or presence of the P300 makes it possible to find which are the line and column that contain the desired letter, and as such to find this letter. The P300 speller proposed by Donchin et al could enable its users to spell up to 7.8 letters per minute [DSW00].

As this application is based on the P300, it does not require training for being used, and have the advantage to be useful for anyone who can control his gaze. As such, this application is very popular [KSC+06, VMS+06] and is currently used in order to help paralyzed persons such as persons suffering from Amyotrophic Lateral Sclerosis (ALS) [Bir06, VMS+06].

It should also be mentioned that other BCI applications based on a similar principle have been developed and validated for ALS subjects. In these P300-based applications, the letter matrix was replaced by a matrix of directional arrows for controlling a cursor in 2D [PGT+06], a 4-choice matrix ("yes", "no", "pass", "end") [SD06], or a matrix of pictures of electronic devices [HVED08]. By using various signal processing methods, these three studies obtained positive results showing that P300-based BCI applications could be used by disabled people.

1.7.1.3 Cursor control through sensorimotor rhythms: the Wadsworth center BCI

The BCI of the Wadsworth center is based on the control of the sensorimotor rhythms μ and β, following a learning based on operant conditioning [WMNF91, WMV00, WM04]. In the standard application, a cursor displayed on a screen is moving horizontally, from left to right, with a constant speed. The user can control the cursor vertically, by making the amplitude of his sensorimotor rhythms vary. On the right side of the screen, several buttons (generally between 2 and 4) are displayed and are arranged vertically. The user has to adjust the vertical position of the cursor so as this cursor could hit the desired button, once it has reached the right side of the screen. For instance, each of these buttons can represent a set of letters that

the user will select. By distributing the selected letters among the buttons and repeating the selection procedure, the user will be able to select a specific letter [VMS+06]. This whole system has been shown to be useful and efficient for restoring communication for ALS persons [KNM+05].

1.7.1.4 Functional electric stimulation controlled by thoughts: the Graz BCI

The BCI of the Graz University of Technology is a BCI based on motor imagery. In this BCI, an imagined movement of the left hand, right hand, feet or tongue is associated to specific commands [KFN+96, PNG+00, PNM+03]. This BCI is used in numerous applications such as controlling a prosthesis [GHHP99] or a virtual keyboard [SMN+04], or interacting with virtual environments [LSF+07]. Here, we focus on an application of this BCI for controlling a Functional Electric Stimulation (FES) [PMP+03, PMPPR05]. In this application, a subject, with a complete paralysis of his left hand, is equiped with an FES system. Such a system uses electrodes, placed on the forearm of the subject, in order to send him an electrical current which forces his muscles to tense, a task that the subject is not able to perform voluntarily. In order to control this FES system, the subject is also equiped with an EEG cap which is part of the Graz BCI. The subject has to use foot motor imagery in order to activate the FES system for tensing or relaxing his muscles, i.e., for closing or opening his hand. The BCI used here is a self-paced BCI and as such is able to detect the imagined foot movements at any time.

1.7.1.5 Power wheelchair control by thoughts: the IDIAP BCI

The IDIAP BCI is an asynchronous BCI that can recognize 3 different mental tasks [MM03]. These mental tasks are, for instance, imagined left or right hand movements or mental cube rotation. Using these three mental tasks enables the BCI user to use three different commands, for controlling a power wheelchair for instance [VML+07]. In this wheelchair application, the three mental tasks were associated to the commands “turn left”, “turn right” and “move forward”. As this BCI is asynchronous, the control of the wheelchair was relatively natural for the users. Moreover, the wheelchair is based on ambient intelligence techniques in order to assist the user in his control task, according to a principle of “shared autonomy” [GGC+07]. Indeed, the wheelchair is equiped with sensors in order to obtain a internal representation of the environment. This representation makes it possible to combine the classifier output with the environment context, in order to compute the best possible movements. For instance, this makes it possible to perform more optimal and smooth trajectories or to avoid obstacles automatically [GGC+07, VML+07].

1.7.1.6 Hex-o-Spell: brain actuated spelling with the Berlin BCI

The last application that we present here is, as the P300 speller, a brain actuated spelling application. This application is known as Hex-O-Spell and is developped by the Berlin group [BDK+06b, BKD+07b, MB06]. In this application, the user has to control the rotation and length of an arrow by using motor imagery (right hand and foot motor imagery) in order to select a cell in a Hexagon (thus containing 6 cells), each cell containing a group of letters or a letter. Thus, imagined right hand movements are used to make the arrow rotate clockwise (see

Figure 1.14) whereas imagined foot movements are used to increase the length of this arrow. If

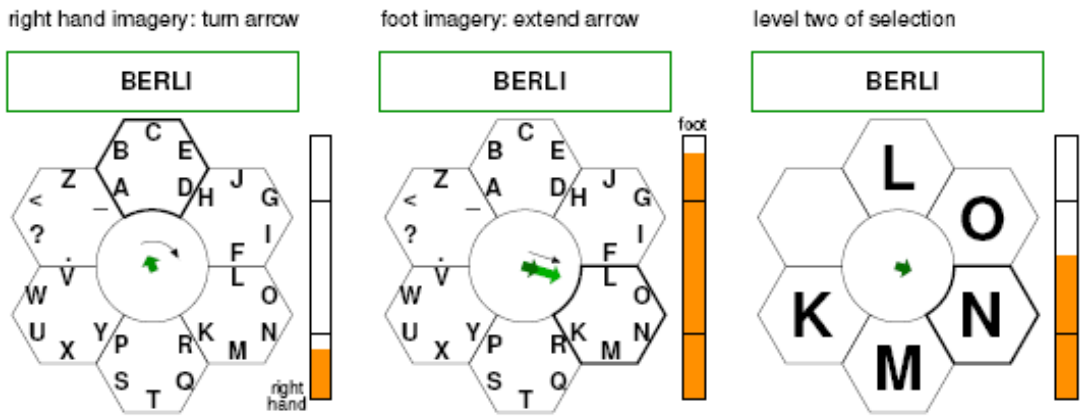


Figure 1.14: The interface of the "Hex-o-spell" application, as displayed on the BCI user's screen (picture from [BKD+07b]).

the arrow reaches a cell, two possible events may happen:

- the cell contains a group of letters. In this case, these letters are distributed among all the cells, and the selection process is repeated once;
- the cell contains a single letter. In this case this letter is selected.

This application, which is based on an asynchronous BCI, makes it possible to select a letter in only two steps, and as such has a good information transfer rate (up to 7.6 letters per minute).

Thus, there are already several medical applications of BCI dedicated to disabled people, and especially to paralyzed people. These applications have proved useful for these persons, which enhances the interest and potential of BCI. However, it should be noted that the BCI currently used in these applications may still be greatly improved, in order to reach higher information transfer rates, more robustness, more commands and comfort for the user, etc. Moreover, only a very small number of BCI are currently used outside laboratories whereas their ultimate goal for medical applications is to be used at the patient's home. These points also concern other applications of BCI, such as BCI applications for multimedia and virtual reality, which can be potentially used by healthy persons.

1.7.2 BCI applications for multimedia and virtual reality

In addition to medical and rehabilitation applications, there is an increasing number of BCI applications for multimedia [EVG03], such as for simple 2D video games [KBCM07, MBC07, MBBB04] to more advanced 3D video games and virtual worlds [LSF+07, LLR+08].

This section first presents the pioneer works related to the combination of Virtual Reality (VR) and BCI. Then it describes the main applications in which BCI are used as an interaction device for virtual environments. These works are divided into two main categories, according

to Bowman's classification [BKJP05]: systems used for navigating virtual worlds and systems used for selecting and/or manipulating virtual objects. Finally, this section presents how VR technology can be used to study and improve current BCI systems.

1.7.2.1 Pioneer works

Among the first works to use BCI for interacting with simulated environments, we can quote the work of Nelson *et al*, in the 90's [NHC+97]. In their work, Nelson *et al* have studied new interfaces for controlling a flight simulator [NHC+97]. To this end, they combined scalp EEG signals with EMG (ElectroMyoGraphy) signals recorded on the user's forehead. With this hybrid system, users were able to perform a 1-dimensional control of the flight simulator.

In 2000, Middendorf *et al* have also studied the use of brain signals for controlling a flight simulator [MMCJ00]. In this work, users can make a flight simulator (without immersive 3D rendering) roll towards the left or right by using SSVEP. Indeed, two flickering lamps were used to generate SSVEP in the user's EEG. The user had learnt to control the amplitude of their SSVEP by operant conditioning and could make the simulator roll according to this SSVEP amplitude.

Finally, still in 2000, Bayliss and Ballard specifically studied whether it was possible to combine BCI with VR [BB00]. To this end, subjects took part in a standard virtual driving stimulation. Subjects were asked to stop their car at red stop lights. The analysis of the subjects' EEG revealed that the appearance of a red stop light triggered a P300 [BB00]. This proved that even when users are immersed in a complex virtual world, it is still possible to record and identify relevant brain signals to drive a BCI. These pioneer works have been confirmed several years later by works which used BCI as VR interaction devices, as described in the following sections.

1.7.2.2 Navigating virtual environments by thoughts

Most existing works related to BCI and VR focus on navigating Virtual Environments (VE). The existing applications can be divided into two groups: 1) applications using the BCI to rotate the virtual camera and 2) applications using the BCI to travel in the VE.

Rotating the virtual camera: In order to enable BCI users to perform camera rotation in VE, two brain signals have been used: motor imagery and μ rhythm (through operant conditioning).

The University College London and the Graz University of Technology have used a synchronous BCI based on left and right hand motor imagery to enable users to rotate the virtual camera towards the left or right in a virtual conference room [LSL+04], or in a virtual bar [FLG+07] (see Figure 1.15). The subject's performances lied between 80% and 100% of accuracy, this accuracy being defined as the percentage of correctly classified mental states.

Touyama *et al* have used SSVEP to control the camera rotation within a CAVE^{TM} VE [TAH08]. To this end, two flickering buttons, with different flickering frequencies, were positioned on the left and right part of the VE. These buttons were used to trigger SSVEP which detection through a self-paced BCI made the camera rotate towards the left or the right. Their system enabled an accuracy between 70 % and 80 % for three naive subjects.

Figure 1.15: Rotating the virtual camera in a virtual bar by motor imagery (pictures from [FSS+04, FLG+07]).

Pineda *et al* have used a BCI based on the μ rhythm to interact with a "First Person Shooter" (FPS) 3D video game [PSVH03]. More precisely, a high μ level (higher than a threshold) triggered left rotations of the camera whereas a low μ level (lower than a threshold) triggered right rotations. A middle μ did not triggered anything, thus enabling a self-paced mode of operation. In this game, the other commands were classically issued using the keyboard.

Travelling and moving in a virtual environment: Several studies have demonstrated the possibility to travel a VE by using a BCI. The University College London and the Graz University of Technology have designed a synchronous BCI which enabled a user to move forward in a virtual street by imagining foot movements and to stop moving by imagining right hand movements [FLG+07, PLK+06].

The same groups have proposed an experiment in which a user could explore a virtual apartment using a synchronous BCI [LLK+07]. In this experiment, at each junction, the user had to select the next corridor (among two) he wanted to walk into. To achieve this, two arrows were displayed on the ground, indicating the two possible corridors. The user had to select the desired arrow by performing the associated motor imagery task during a period of 2 seconds. Once the corridor was selected, the user was moved automatically in the VE towards the next junction. The experiment showed that the users were able to explore the virtual flat and to reach a given room, with a classification accuracy which ranged from 67 % to 93 %, for 9 subjects with little BCI experience.

Ron-Angevin *et al* have also designed a synchronous BCI for moving in a VE (see Figure 1.16). In their system, the users, equipped with a Head Mounted Display (HMD), could control the left or right movement of a virtual car in order to avoid obstacles or reach ramps to make the car jump [RAERL05]. The car was continuously going forward and right hand motor imagery was used to steer towards the right and a relaxed mental state was used to steer towards the left.

More recently, the Graz group have performed several studies that used self-paced BCI to move in virtual environments [LFMP+07, LSF+07, SLS+08] (see Figure 1.17). Leeb *et al*

Figure 1.16: Steering of a virtual car through hand motor imagery (picture from [RAERL05]).

have shown that a paralyzed patient was able to move a wheelchair along a virtual street by using motor imagery of his paralyzed feet and a self-paced BCI based on a single EEG electrode [LFS$^+$06, LFSP07, LFMP$^+$07]. To this end, when the system detected an imagined foot movement, it triggered a forward movement in the VE whereas when no foot motor imagery was detected no movement was triggered. The performance obtained were quite good, with several sessions in which the patient reached an accuracy of 100%. It should be noted that the patient has been previously trained to motor imagery over several months.

Figure 1.17: Moving along a virtual street using a self-paced BCI based on foot motor imagery (pictures from: [LLK$^+$07]).

Scherer *et al* have designed a self-paced BCI for freely exploring a VE [SLS$^+$08]. In this application users should perform left or right hand motor imagery to turn towards the left or right respectively, and foot motor imagery to move forwards in the VE. This system proved relatively natural for the subjects. As an evaluation, 3 subjects had to collect three coins in the VE, within a given time. The results showed that 2 subjects out of 3 successfully completed the task. However, it should be mentioned that the performances of this system were relatively modest, with a relatively large false positive rate (i.e., the percentage of times that the system detected a mental command when no mental command was actually performed), 16.9 % on average and a relatively small true positive rate (i.e., the percentage of times that the system detected a mental command when a mental command was actually performed), 28.4 % on

average.

Finally, Leeb *et al* have proposed a self-paced BCI system for exploring a virtual representation of the national Austria library [LSFP07]. This BCI used a single motor imagery task to trigger a movement along a predefined pathway (see Figure 1.18). 5 subjects participated to this experiment. Results were shown to be quite good with true positive rates between 14 % and 50 % and false positive rates between 2 % and 7 %.

Figure 1.18: Exploring a virtual model of the national Austria library by thoughts (pictures from [LSF+07]).

1.7.2.3 Selecting and manipulating virtual objects

Three main studies have focused on selecting and/or manipulating Virtual Objects (VO) by using a BCI (see Figure 1.19). In the first study, Lalor *et al* have developed a 3D video game driven by a synchronous BCI [LKF+05a]. In this game, a monster went from platforms to platforms by walking along a tight rope. From time to time, the monster lost its balance, and the user had to restore it by using the BCI. To do so, two flickering checkerboard were placed on each side of the VE, in order to elicit SSVEP at different frequencies. When the system detected that the user was focusing on the left checkboard, it restored the monster's balance towards the left. A symmetric operation was performed with the right checkerboard. The users who participated in this study reached a classification accuracy of 89% on average.

Figure 1.19: From left to right: manipulating electronic devices in a virtual flat with a P300-based BCI (picture from [Bay03]), controlling a virtual avatar by motor imagery (picture from [FLD+07]), controlling the balance of a virtual monster using SSVEP (picture from [LKF+05a]).

In a second study, Bayliss has designed a P300-based BCI application to interact with a virtual flat. In this study, subjects were immersed in a virtual flat and had to turn On or Off various VO such as a lamp, a Hi-Fi or a TV [Bay03]. To do so, 3D spheres were randomly appearing over these VO, and users had to count the number of appearances of a sphere over the VO they wanted to use. After a while, the VO which was the most likely to have had trigerred the P300 was turned On or Off.

Finally, Friedman *et al* have proposed a system to control a virtual avatar using motor imagery [FLD+07]. This system used a synchronous BCI with which a foot motor imagery made the avatar walk, whereas a hand motor imagery made the avatar wave. For this experiment, the classification accuracy reached 87 % on average.

Table 1.2: Summary of BCI-based VR applications

	Number of commands	neuro-physiological signal	synchronous or self-paced	VE or context	Reference
Selecting-	2	SSVEP	synchronous	controlling a virtual monster	[LKF+05a]
manipulating	2	P300	synchronous	turning on/off VO in a virtual apartment	[Bay03]
VO	2	MI	synchronous	controlling a virtual avatar	[FLD+07]
	2	MI	synchronous	walking a virtual street	[PLK+06]
	2	MI	synchronous	exploring a virtual apartment	[LLK+07]
Travelling	2	MI	synchronous	virtual car steering	[RAERL05]
the VE	1	MI	self-paced	walking along a virtual street	[LFMP+07]
	3	MI	self-paced	exploring the “free-space”	[SLS+08]
	1	MI	self-paced	exploring a virtual library	[LSFP07]
Rotating	2	MI	synchronous	exploring a virtual bar	[FSS+04]
the virtual	2	MI	synchronous	exploring a virtual conference room	[LSL+04]
camera	2	μ rhythm	self-paced	playing a FPS game	[PSVH03]
	2	SSVEP	self-paced	exploring a virtual city	[TAH08]

Interestingly, it seems that in current applications, selecting and manipulating VO is mostly

achieved using evoked signals, whereas navigating VE is mostly achieved using spontaneous signals. This suggests that interesting combinations between those kinds of signals could be achieved to increase the number of available commands. It should also be noted that no current BCI application enables the user to select/manipulate VO and navigate VE at the same time. As such proposing a technique to perform both interaction tasks would clearly open very interesting possiblities for VR applications. Finally, it can be observed that the mapping between the mental states and the commands is still relatively simple for VR applications based on a BCI. As such, it appears as essential to seek innovative ways of using the few available commands in order to enlarge the number of actions available for the user.

1.7.2.4 Virtual reality for studying and improving brain-computer interfaces

So far, we have reported studies that have used BCI as a new interaction device for VR applications. Conversely, VR can also prove very interesting for BCI applications. Indeed, VR makes it possible to perform safe and carefully controlled BCI experiments towards a real-life use, can increase the motivation of subjects, reduce the time necessary to learn how to use a BCI, and, finally, can be used to study brain dynamics and behavior.

Safe and carefully controlled experiments: One of the numerous advantages of VR, is that it makes it possible to test and study various systems and tools in VE rather than in real-life, which is safer, cheaper, more convenient and which enables more controlled experiments. Thus, by using BCI in VE, several researchers have shown that humans can use BCI systems in complex and visually rich 3D environments [PSVH03, LFMP$^+$07]. Similarly, they have shown in VR that relatively complex tasks commonly performed in real-life could also be performed with a BCI despite the high mental workload generated by the task [LLK$^+$07]. VR also enabled Friedman *et al* to study different mapping between mental states and commands [FLD$^+$07]. The results showed that, surprisingly, the most natural mapping (e.g., using foot motor imagery to walk along the VE) does not give better results, in terms of classification accuracy, than non-natural mapping (e.g., walking into the VE by using hand motor imagery) [FLD$^+$07]. To sum up, VR makes it possible to study BCI in close to real-life conditions.

Improving BCI learning and performances: A second interest in using VR and videogames for BCI is related to the increased motivation it provides to users. Several studies have compared feedback consisting of classical 2D displays with feedback consisting of entertaining VR applications [LLK$^+$07, RAERL05]. These studies have shown that users' performances were higher with a VR feedback than with a simple 2D feedback. Moreover, there are evidences that the more immersive the VR display, the higher the performances and motivation of users [FLG$^+$07, LLK$^+$07]. Even if some observations need to be confirmed, VR appears as a way to shorten BCI learning and increase users' performances by increasing their motivation.

Studying brain dynamics with VR: Finally, another advantage of VR for BCI, is that VR can be seen as an ideal environment to study brain dynamics and behavior. For instance, Pfurtscheller *et al* have shown that visualizing a moving virtual object or a moving virtual hand triggered ERD/ERS over the sensorimotor areas of the brain, i.e., phenomenons similar to what

happened when a subject effectively performs a movement [PSL+07]. Another example is the work of Arrouet *et al* which used inverse solutions to enable a subject to navigate in his own brain, represented in 3D, and to visualize in real-time his own brain activity [ACM+05]. To this end multiple 3D objects were displayed on screen, each one corresponding to a specific brain region. The size and color of these objects represent the activity in the corresponding brain region (see Figure 1.20). Such a tool could be used to study, in real-time, what is happening in the brain following a given stimulus or mental task, as with the Brain TV application [LJB+07]. It is also expected that such an immersive neurofeedback would be more engaging and informative and would improve the user's abilities to control his brain activity.

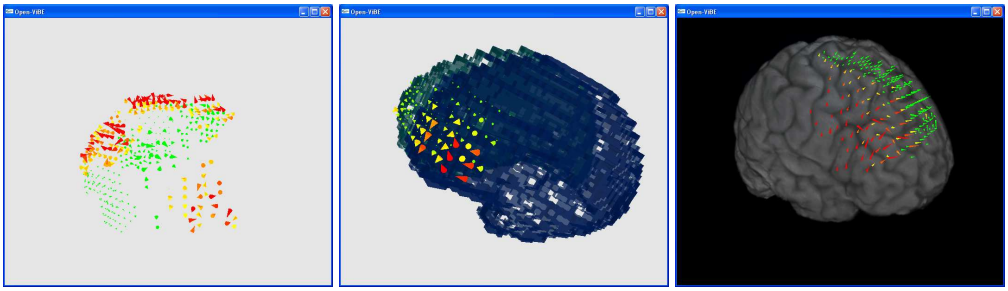


Figure 1.20: 3D visualization of brain activity in VR [ACM+05]

1.7.2.5 Conclusion

To sum up, it appears that not only BCI can prove useful for VR, but also VR can prove useful for BCI. Thus, the coupling between VR and BCI appears as very promising, but a lot of things still need to be done. For instance, it would be very interesting to study which kind of VR feedback will lead to the best performances and to the shortest training time for the subjects. Also, it would be interesting to study further the impact of motivation and presence as well as to quantify their effects.

1.7.3 Other BCI applications

Besides the VR and medical applications, some other applications of BCI have been proposed. Among these applications, we can mention brain-actuated robots [LLA07c, Mil08]. For instance, BCI have been used to control relatively simple mobile robots [TMK+03, dRMRMG03] or to control the dog robot AIBO from Sony [IKP06, CAB+06, COM+07]. BCI have also been used to control more advanced mobile robots such as robotic wheelchairs [TMW05, RBG+07, VML+07] or even humanoid robots [BSCR07, BSCR08]. A specificity shared by several of these robotic applications is that a lot of intelligence is placed into the robot. As such, the BCI user only uses a small number of commands (between 2 and 4), but these are very high-level commands, which will correspond to complex tasks achieved by the robots [BSCR07, dRMRMG03, RBG+07].

Other BCI applications include musical applications, in which a BCI is used to drive a musical composition. In such applications, some mental states, such as the prominence of a

given brain rhythm, are associated to a change of the musical style or to a change in the music tempo [MB04]. These signals can also be combined with other biological signals such as EMG [ABC+06]. Recently, a BCI has also been used to make paintings thanks to a P300-based application [KHFH08].

Naturally, the range of possible BCI applications is relatively large [Moo03], and there is no doubt that the number of applications of such a technology will rapidly increase.

1.8 Conclusion

This chapter aimed at reviewing the current methods used to design BCI, as well as the existing BCI applications. In this regard, we have first seen that, although several invasive or non-invasive methods were available to measure brain activity, EEG was by far the most convenient and the most popular one. Second, we have seen that EEG signals used to drive a BCI could be divided into 2 categories, namely, evoked potentials, such a P300 or SSVEP, and spontaneous signals, such as slow cortical potentials or motor imagery. Then, we have reviewed preprocessing techniques used in BCI. These techniques are generally temporal filters, such as direct Fourier transform filtering or spatial filters such as common spatial patterns and independant component analysis. We have also reviewed feature extraction techniques employed for BCI. These techniques could be divided in 3 categories, namely, temporal features such as Hjorth parameters, frequential features such as band power and time-frequency representation-based features such as wavelets. Then, we reviewed classification algorithms used in BCI. These algorithms can be gathered into 5 different categories. These categories are 1) linear classifiers, with, e.g., support vector machines, 2) neural networks, with, e.g., multilayer perceptrons, 3) nonlinear Bayesian classifiers, with, e.g., Bayes quadratic, 4) nearest neighbor classifiers, with, e.g., k nearest neighbors and 5) combination of classifiers with, e.g., boosting. Finally, we surveyed BCI applications, and we have seen that BCI were mostly used for medical applications such as spellers or prosthesis, but also for multimedia and virtual reality applications such as 3D video games.

This first chapter has highlighted that there have already been a relatively large number of studies related to BCI research that aimed at exploring and proposing various signal processing and classification algorithms for designing a BCI. Despite this large number of studies, the most appropriate algorithms, if any, have not been identified yet. Moreover, the BCI community has highlighted the need to explore and/or design more efficient algorithms (in terms of accuracy), from which it could be possible to gain insights about the brain dynamics [MAM+06]. These two points are specifically addressed in this manuscript and more particularly in the four next chapters, gathered within Part 1: EEG signal processing and classification.

Similarly, several prototypes of applications based on BCI have been proposed, mainly in the medical and rehabilitation field but also in the multimedia and virtual reality field. Concerning this last field, the interest and potential of VR for BCI have been particularly stressed. However, there is still a lot to study regarding BCI-based VR applications, such as the preferences of the BCI user, the impact of the feedback provided to this user, or the design of appropriate interaction paradigms. These points are addressed in Part 2 of this manuscript:

virtual reality applications based on BCI technology.

Part 1:
EEG signal processing and classification

Chapter 2

Preprocessing and Feature Extraction: FuRIA, an Inverse Solution-based Algorithm using Fuzzy Set Theory

2.1 Introduction

One objective of this PhD thesis is to study methods which could lead to 1) more efficient BCI in terms of information transfer rate and 2) interpretable BCI. To fulfill these objectives, we can focus at the level of preprocessing and feature extraction, or at the level of classification. In this chapter, we address the problem of preprocessing and feature extraction.

In this chapter, we propose a trainable feature extraction algorithm for BCI which relies on inverse solutions as well as on the fuzzy set theory. This algorithm is called FuRIA which stands for **Fu**zzy **R**egion of **I**nterest **A**ctivity. FuRIA can automatically identify what are, for a given subject, the relevant Regions Of Interest (ROI) and frequency bands for the discrimination of mental tasks, even for multiclass BCI. The activity in these ROI and frequency bands can be used as features for any classifier.

The chapter is organized as follows: Section 2.2 provides additional details on the concept of inverse solution and on its current uses for BCI design. Section 2.3 describes in details the FuRIA algorithm we propose. Section 2.4 describes the evaluations of FuRIA achieved on binary and multiclass EEG data sets. Finally, Section 2.5 discusses the results and concludes.

2.2 Inverse solutions and BCI

This section begins with a brief recall about what are inverse solutions and presents a way of formulating inverse solutions as quadratic forms, i.e., as computationaly efficient forms. Then, this section focuses on inverse solution-based BCI and highligths the main limitations of current systems.

2.2.1 Inverse solutions as a quadratic form

As mentioned in Chapter 1, inverse solutions aim at estimating the brain dipole activity $\hat{c}(t)$ by using only the scalp measurements $m(t)$ and the leadfield matrix (head model) K (see section 1.4.4):

$$\hat{c}(t) = Tm(t) \tag{2.1}$$

where T is the generalized inverse of K. Congedo has shown that any linear and discrete inverse solution could be formulated as a quadratic form [Con06]:

$$\gamma_v(t) = m(t)^T Q_v m(t) \tag{2.2}$$

where Q_v is an $N_e * N_e$ matrix (N_e being the number of electrodes used) denoted as the *inverse operator* for voxel v [Con06] and γ_v is the current density in voxel v. The superscript T denotes transpose. Typically $Q_v = T_v^T T_v$ with T_v being the v^{th} row of T. The current density in a given ROI Ω, i.e., in a set of voxels, can be computed as follows:

$$\gamma_\Omega(t) = \sum_{v\in\Omega} \gamma_v(t) = m(t)^T Q_\Omega m(t) \qquad with \qquad Q_\Omega = \sum_{v\in\Omega} Q_v \tag{2.3}$$

This notation is very convenient as it allows very fast computations, whatever the number of voxels in the ROI Ω. Actually the Q_Ω matrix can be computed offline, and the size of this matrix depends only on the number of electrodes used.

2.2.2 Inverse solution-based BCI

As mentioned in section 1.4.4, a few recent studies have started to evaluate the efficiency of inverse solutions as feature extractors for BCI [LLA07b, NKM08, KLH05]. Indeed, there are increasing evidences that the use of inverse solutions would improve the performances of the system, in terms of correct recognition rates. Inverse solutions such as ELECTRA [GGP$^+$05], equivalent dipole analysis [QDH04, KLH05] or minimum norm estimates [BCM$^+$07, BMG08] have been applied to non-invasive BCI designs. All these methods obtained very promising results. These good results can be explained if we consider inverse solutions as spatial filters based on physiological a priori. Hence, they make it possible to focus on relevant information while removing the noise coming from other brain regions and not related with the mental tasks performed by the subject.

In spite of these promising results, some limitations remain. Indeed, current methods are either general-purpose, i.e., they have the ability to deal with any kind of mental task, or generator of few features but rarely both at the same time. Several methods require strong a priori knowledge on the neurophysiological mechanisms involved by the mental tasks used, and hence, are not general-purpose at all [QDH04, KLH05, BCM$^+$07]. With these methods, the ROI to be used must be defined beforehand and by hand. These methods are currently limited to the use of mental tasks that involve the motor and sensorimotor areas of the brain. Using predefined ROI also raises the problem of specialization. Indeed, it is well known that each subject has his own specificities, in terms of spatial (involved brain regions) or frequential (involved frequency bands) features [WBM$^+$02]. Hence, a non-general-purpose method, exclusively based

on a priori knowledge, will not be able to adapt to each subject's specificities, and will have, most probably, non-optimal performances.

A few general-purpose methods, based on distributed inverse solutions, have been proposed [GGP+05, NKM08, BMG08]. The main limitation of these methods is that they must extract one or several features per voxel, which generates a very large number of features. Actually, the head models generally used are composed of hundreds or thousands of voxels. This requires the use of feature selection techniques [GGP+05, NKM08, BMG08]. Even though this solution gives good results, the number of features used remains generally relatively high, particularly in comparison with the number of features extracted by non-general-purpose methods [QDH04, KLH05]. A high number of features also reduces the interpretability of the resulting model. Moreover, in these methods, all voxels are processed independantly, whereas a number of them are dependant of each other and as such should be gathered in brain regions.

Congedo *et al* have proposed a method which is both general-purpose and generator of few features as voxels are gathered into ROI [CLL06]. This method, which combines data-driven spatial filters and an inverse solution, has obtained results comparable to those of the winner of the BCI competition 2003. However, this method still needs improvements as it is not completely automatic and limited to the use of two ROI whose spatial extension is hard to define [CLL06].

2.3 The FuRIA feature extraction algorithm

FuRIA is a trainable feature extraction algorithm based on an inverse solution for non invasive BCI. It can learn and use subject specific features even from multiclass data sets. It uses the concept of ROI in order to generate a relatively small number of features. A feature extracted with FuRIA is related to a clear physiological information as it corresponds to the activity in a given brain region and its associated frequency band. Moreover, and contrary to existing methods, FuRIA can automatically identify these relevant ROI, as well as the frequency bands in which these ROI current densities are discriminant. Finally, FuRIA also introduces the concepts of fuzzy ROI and fuzzy frequency bands, which concepts are used to obtain increased classification performances.

FuRIA aims at being modular in the sense that various kinds of inverse solutions could be used within it. This section briefly describes the inverse solutions that could be used within FuRIA and the specific one that we used in our implementation. It then describes in details the FuRIA feature extraction algorithm.

2.3.1 Inverse solutions for FuRIA

FuRIA aims at being used with any kind of linear and distributed inverse solution. Actually, distributed solutions enable the use of a large number of dipoles rather than a few equivalent dipoles. As such they provide more information and are more flexible. On the other hand, the use of linear inverse solutions appears as essential for BCI applications. Indeed, the strong real-time constraints that are imposed when using a BCI online prevent the use of non-linear inverse solutions as they are computationally demanding. Several linear and distributed inverse solutions have been used for BCI, such as ELECTRA [GGP+05], LORETA/sLORETA

[LNM06, CLL06] or the depth-weighted minimum norm technique [BCM$^+$07]. FuRIA can be used with any of these inverse solutions.

2.3.2 The sLORETA inverse solution

In our implementation of FuRIA, we used sLORETA (standardized low resolution electromagnetic tomography) which is an instantaneous, discrete and linear inverse solution proposed by Pascual-Marqui [PM02]. sLORETA is known to have very good localization properties [PM02] including no localization bias in the presence of measurement and biological noise [PM07]. Moreover, it has been proved experimentally that sLORETA was suitable for the design of EEG-based BCI [CLL06].

To solve the inverse problem, sLORETA relies on a regularized least mean square solution to equation 2.1:

$$T = K^T(KK^T + \alpha_R X)^+ \tag{2.4}$$

where α_R is a positive regularization parameter and X is the centering matrix which is used to re-reference the data to common average reference. The exponent $+$ denotes the Moore-Penrose matrix pseudo-inverse. In order to reach a high localization capability, sLORETA is based on a standardization of this solution. As such, the activity (current density) γ_v of a voxel v is obtained as follows with sLORETA:

$$\gamma_v = \hat{c}_v^T S_v^{-1} \hat{c}_v \tag{2.5}$$

where $S = TK$ is the resolution matrix, with S_v being the v^{th} 3*3 block of S. Similarly, $\hat{c}_v$ is the v^{th} triplet of $\hat{c}$. Consequently, to express the current density with sLORETA as a quadratic form (see equation 2.2), one should use $Q_v = T_v^T S_v^{-1} T_v$. The interested reader can find more details about sLORETA in [PM02, PM07]. However, it should be reminded that the principle of the FuRIA algorithm is not dependent on the inverse solution chosen. Consequently, any other distributed and linear inverse solution could be used instead of sLORETA.

2.3.3 Overview of the FuRIA algorithm

2.3.3.1 Training of FuRIA

In order to be used, FuRIA has first to be trained using a set $\Theta = \{(m(t), C)_{1..N_t}\}$ of labelled training data, with C being the class of $m(t)$, i.e., the mental task performed by the subject while $m(t)$ was recorded. The goal of this training phase is to find subject-specific ROI Ω_l and frequency bands Φ_l that contain the most relevant information for mental task discrimination. This training phase is accomplished offline, in three main steps:

- **Identification of statistically discriminant voxels and frequencies:** The goal of this step is to identify the ordered pairs $w_k = (f_i, v_j)$ $(k \in [1, N_k])$, with f_i being a frequency and v_j being a voxel, with the largest discriminative power. In order to do so, we rely on a statistical analysis for comparing, between the different classes, the mean current density in each frequency f_i $(i \in [1, N_f])$ and in each voxel v_j $(j \in [1, N_v])$.

- **Creation of ROI and frequency bands:** This step aims at gathering the voxels v_j selected at the previous step into several ROI Ω_l, as well as at gathering the selected frequencies f_i into several frequency bands Φ_l. In order to do so, we rely on a clustering algorithm for finding clusters of voxels and frequencies and for transforming these clusters into ROI and frequency bands. Thus, by the end of this step, we have created a set of N_w ordered pairs $W_l = (\Phi_l, \Omega_l)$ $(l \in [1, N_w])$ (one pair per cluster) in which each ROI is associated to a single frequency band. This frequency band should gather the frequencies in which the activity of the ROI voxels is discriminant.

- **Fuzzification of ROI and frequency bands:** The previously found ROI Ω_l are turned into fuzzy ROI $\tilde{\Omega}_l$ and the frequency bands Φ_l are turned into fuzzy frequency bands $\tilde{\Phi}_l$. This aims at giving more importance to the more discriminant voxels and frequencies, while still using the information contained in the less discriminant ones. The overall objective is to increase the discriminative power of the W_l pairs.

2.3.3.2 Use of FuRIA for feature extraction

Once the fuzzy pairs $\tilde{W}_l = (\tilde{\Phi}_l, \tilde{\Omega}_l)$ have been identified, FuRIA can be used for feature extraction. The features extracted are the current densities in the fuzzy ROI $\tilde{\Omega}_l$ after band-pass filtering EEG signals in the associated fuzzy frequency bands $\tilde{\Phi}_l$.

All these steps are detailled hereafter. In the following, the description of each training step is divided into two parts: 1) a section "algorithm" which describes the general algorithm we propose and 2) a section "implementation" which describes the specific implementation of FuRIA that we evaluate in this chapter. In this regard, it is worth noting that other implementations could be used (e.g., different clustering algorithms, different statistical tests, ...) as long as they are consistent with the algorithms proposed.

2.3.4 First training step: identification of statistically discriminant voxels and frequencies

2.3.4.1 Algorithm

The first step of the training of FuRIA aims at identifying the pairs w_k of voxels v_j and frequencies f_i which are the most discriminant, i.e., the pairs of voxels and frequencies whose current density is the most different between classes. In order to do so, we perform a statistical analysis that compares the mean current densities between classes for each pair $w_k = (f_i, v_j)$. To this end, each training EEG record $m(t)$ passes through the following procedure (see Fig. 2.1):

1. $m(t)$ is decomposed into frequency bands by using a set of filters h_i. Each filter h_i is a 2-Hz wide band-pass filter centered on frequency f_i. We denote as $m_i(t)$ the signal resulting from the filtering of $m(t)$ by h_i.

2. the current density $\gamma_{i,j}(t)$ in voxel v_j, for frequency f_i is computed using the inverse solution:

$$\gamma_{i,j}(t) = m_i(t)^T Q_{v_j} m_i(t) \tag{2.6}$$

3. finally, $\gamma_{i,j}(t)$ is averaged over a time window of interest, which starts at sample t_0 and is N_s samples long. The obtained value is then log-transformed:

$$< \gamma_{i,j} >= log(\frac{1}{N_s} \sum_{t=t_0}^{t_0+N_s} \gamma_{i,j}(t)) \tag{2.7}$$

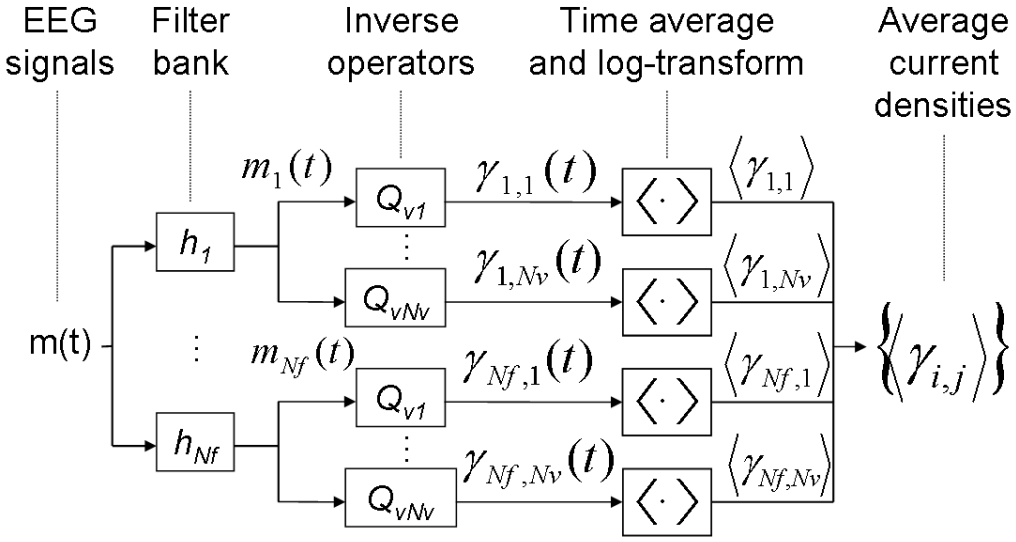


Figure 2.1: Computation of the average current densities $< \gamma_{i,j} >$ in all frequencies f_i and voxels v_j, from a training EEG record m(t).

Then, the $< \gamma_{i,j} >$ are gathered into statistical samples according to the label of their corresponding training record $m(t)$. As such, we obtain one statistical sample per class. These samples are then compared using the statistical analysis. In other words, this statistical analysis compares the mean $< \gamma_{i,j} >$ between classes and hence gives the discriminative power of each pair w_k. Pairs w_k which obtained a p-value higher than a given threshold α (this threshold is a hyperparameter of FuRIA) are not considered anymore in the remaining of the training process. The other pairs are denoted as "significant". This procedure should remove numerous voxels and frequencies and should only keep the ones which are specific to the mental tasks performed by the subject and to the physiology of this subject.

2.3.4.2 Implementation

In our implementation of FuRIA, the h_i filters were either Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters. We used windowed sinc filters as FIR filters [Smi97]

and Yule-Walker filters as IIR filters [FP84]. These filters were chosen as they enable the design of custom filters, which is needed for the fuzzification of frequency bands step (see section 2.3.6). Concerning the statistical test used, we employed Multiple Comparisons Randomization (MCR) tests as described by Holmes [HBWF96]. More precisely, for BCI with only 2 classes, we used MCR t-tests whereas for multiclass BCI (with 3 or more classes) we used MCR ANOVA (ANalysis Of VAriance), as ANOVA can compare multiple conditions, and as such, can deal with more than 2 classes. Naturally, depending on the need of the FuRIA user, different kinds of band-pass filters or statistical analysis could be used instead.

2.3.5 Second training step: creation of ROI and frequency bands

2.3.5.1 Algorithm

This step aims at gathering significant voxels and frequencies into ROI Ω_l ($l \in [1, N_w]$) associated with frequency bands Φ_l, each ROI being associated to a single frequency band. Basically, a given ROI would gather significant voxels, and the frequency band associated to this ROI would gather the frequencies in which these voxels activity is discriminant. Creating such ROI and frequency bands aims at obtaining a compact feature representation. Indeed, using the activity in a few ROI and frequency bands as features should lead to much less features than when considering voxels and frequencies alone, as done frequently [GGP$^+$05, NKM08]. Moreover, the activity in neighboring voxels and frequencies tends to be statistically correlated [MML$^+$04]. As such it should be more appropriate to use these voxels and frequencies together rather than independently.

Ideally, we would like to gather voxels belonging to the same neural source into the same ROI. We would like, at the same time, to gather into a single frequency band the frequencies at which a similar ROI appears. This means it is desirable to find clusters gathering both voxels and frequencies. In order to find these clusters, we associate to each significant pair $w_k = (f_i, v_j)$ the feature vector $[x_j, y_j, z_j, f_i]$ in which x_j, y_j, z_j are the spatial coordinates of v_j in the head model used. The vector elements are normalized to zero mean and unit variance in order to deal with the different ranges between space and frequencies. Then, we apply a given clustering algorithm to all these vectors. Finally, for each obtained cluster, we gather all voxels whose associated vector belongs to this cluster into the same ROI Ω_l and we associate to this ROI the frequency band $\Phi_l = [f_{min}, f_{max}]$. Here, f_{min} and f_{max} are respectively the minimal and maximal value of the coordinate f_i among all the vectors belonging to this cluster. This clustering gives a set of ordered pairs $W_l = (\Phi_l, \Omega_l)$ which are expected to be discriminant.

2.3.5.2 Implementation

When using sLORETA, the neural sources tend to appear as local maximums of the current density [PM02] and hence as local maximums of the statistics obtained in output of the statistical analysis. Consequently, we used Mean Shift as the clustering algorithm since it gathers vectors attracted by the same local maximum of the underlying density function [CM02]. However, as the voxels coordinates and frequencies considered are regularly spaced within their numerical domain, the underlying density function for the vectors $[x_j, y_j, z_j, f_i]$ will be relatively flat and thus will prevent a proper use of Mean Shift. To cope with this problem, we used a slightly

modified version of the Mean Shift algorithm for clustering. This slight modification simply consists in replacing $\hat{D}$, the standard density estimate at point P used in Mean Shift:

$$\hat{D}(P) = \frac{1}{N_k H^d} \sum_{k=1}^{N_k} \chi\{\frac{1}{H}(P - P_k)\} \tag{2.8}$$

by $\tilde{D}$, a weighted density estimate at point P :

$$\tilde{D}(P) = \frac{1}{N_k H^d} \sum_{k=1}^{N_k} s_k * \chi\{\frac{1}{H}(P - P_k)\} \tag{2.9}$$

with H being the smoothing parameter, χ a kernel, here the Epanechnikov kernel (the optimal kernel for Mean Shift, see [CM02]), P the current vector, P_k, the k^{th} vector from the data set, and d the dimensionality. Finally $s_k = 1 - p_k$, with p_k being the p-value obtained by the w_k pair, during the statistical analysis performed in the previous step. This leads to the following form for the *sample mean shift* vector :

$$M_H(P) = \frac{1}{n_P} \sum_{P_i \in S_H(P)} s_k * (P_k - P) \tag{2.10}$$

Here, $S_H(P)$ is the d-dimensional sphere of radius H centered at P, with n_P vectors inside. This weighted version of Mean Shift will gather into the same cluster all the vectors attracted by the same local maximum of the statistics. As such, this version of Mean Shift is expected to gather altogether the vectors corresponding to the same neural source, as these sources should be local maximums of the statistics (see above).

It should be noted that making clusters according to the local maximums of the statistics is essential when dealing with multiclass BCI based on sLORETA. Actually, when the number of mental tasks performed increases, the number of brain regions involved increases as well. Added to the fact that sLORETA is low resolution, this may lead to a high overlapping of these regions, making classical clustering fail to recover the different sources. We observed this problem experimentally, which led to the conception of this clustering strategy.

2.3.6 Third training step: fuzzification of ROI and frequency bands

The last training step of FuRIA consists in fuzzifying the previously obtained ROI Ω_l and frequency bands Φ_l. Actually, a ROI can be seen as a conventional (or "crisp") set of voxels whereas a frequency band can be seen as a crisp set of frequencies. However, it is clear that in a ROI or in a frequency band, not all the voxels or frequencies have the same discriminative power. Nevertheless, all these elements still carry more or less information that could be used, making it hard to choose which of them should be kept. Moreover, electrophysiologically, the brain regions related to specific brain functions are not well defined for a specific user: their boundaries are naturally "fuzzy". Consequently, rather than selecting a few voxels and frequencies within a ROI, we believe that all significant voxels and frequencies should be used, but the voxels and frequencies that are less discriminant should belong "less" to their ROI and frequency band than the others. Thus, we propose to consider ROI and frequency bands as

fuzzy sets [Zad96a] of voxels and frequencies, in which all voxels and frequencies are given a degree of membership into the ROI or frequency band to which they belong. We denote such kinds of ROI and frequency bands as fuzzy ROI and fuzzy frequency bands. Thus, a fuzzy membership function μ is associated to each ROI and each frequency band. To sum up, two fuzzifications are performed: the fuzzification of space and the fuzzification of frequency.

2.3.6.1 Design of fuzzy ROI and fuzzy frequency bands from a given fuzzy membership function

Algorithm: As mentionned above, a crisp ROI Ω_l is a set of voxels which current density is computed according to Eq. 2.3. A fuzzy ROI $\tilde{\Omega}_l$ is not defined by a set of voxels anymore but by a fuzzy membership function μ_{S_l}. This function provides the degree of membership, in $[0,1]$, of any existing voxel to the fuzzy ROI $\tilde{\Omega}_l$. Contrary to crisp ROI for which all its voxels are used equally (see Eq. 2.3), we define the current density $\gamma_{\tilde{\Omega}_l}(t)$ in a fuzzy ROI as follows:

$$\gamma_{\tilde{\Omega}_l}(t) = \sum_{j=1}^{N_v} \mu_{S_l}(v_j)\gamma_{v_j}(t) \tag{2.11}$$

This leads to:

$$\gamma_{\tilde{\Omega}_l}(t) = m(t)^T Q_{\tilde{\Omega}_l} m(t) \qquad with \qquad Q_{\tilde{\Omega}_l} = \sum_{j=1}^{N_v} \mu_{S_l}(v_j) Q_{v_j} \tag{2.12}$$

Similarly, we associate a fuzzy membership function μ_{F_l} to each frequency band Φ_l. The function μ_{F_l} provides the degree of membership, in $[0,1]$, of any existing frequency to the fuzzy frequency band Φ_l. We can note that this function has exactly the same form as the magnitude response of a digital filter. This means that to band-pass filter a signal in a given fuzzy frequency band we have to design a custom digital filter that have the desired fuzzy membership function μ_{F_l} as magnitude response.

Implementation: We used the window technique to automatically design FIR filters from the desired magnitude response, i.e., from a fuzzy membership function associated to a frequency band [Smi97]. To automatically design the IIR filters, we used the Yule-Walker method [FP84]. This explains why we used a windowed-sinc FIR filter or a Yule-Walker IIR filter for the first training step of FuRIA (see section 2.3.4).

2.3.6.2 Setup of the fuzzy membership functions

Algorithm: In order to determine the kind of fuzzy membership functions to be used as well as their parameters, we first compute the discrimination scores d_{v_j} and d_{f_i} of each voxel v_j and frequency f_i respectively, for each pair $W_l = (\Phi_l, \Omega_l)$:

$$d_{v_j} = \frac{1}{N_f} \sum_{f_i \in \Phi_l} s_{v_j, f_i} \qquad and \qquad d_{f_i} = \frac{1}{N_v} \sum_{v_j \in \Omega_l} s_{v_j, f_i} \tag{2.13}$$

where $s_{v_j,f_i} = 1 - p_{j,i}$ with $p_{j,i}$ the p-value obtained during the statistical analysis step, for voxel v_j at frequency f_i. In order to highly emphasize the contribution of the most discriminative voxels and frequencies we chose exponential fuzzy membership functions:

$$\mu_{S_l}(v_j) = \begin{cases} exp(-\frac{1}{2}(\frac{d_{v_j} - d_{vmax}}{\sigma_v})^2) & v_j \in \Omega_l \\ 0 & otherwise \end{cases} \quad with \quad \sigma_v = \frac{1}{N_v} \tag{2.14}$$

$$\mu_{F_l}(f_i) = \begin{cases} exp(-\frac{1}{2}(\frac{d_{f_i} - d_{fmax}}{\sigma_f})^2) & f_i \in \Phi_l \\ 0 & otherwise \end{cases} \quad with \quad \sigma_f = \frac{1}{N_f} \tag{2.15}$$

where d_{vmax} and d_{fmax} are the maximal scores among voxels and frequencies respectively. This means that the voxel and frequency with the highest discrimination scores obtain the highest degree of membership, i.e, 1.0, while other voxels and frequencies obtain a score that decreases exponentially with their discrimination score. However, the value of the σ parameter given above is only an initial value which may not be the optimal value to maximize the discriminative power of the pairs $\tilde{W}_l = (\tilde{\Phi}_l, \tilde{\Omega}_l)$. Consequently, we then optimize the σ parameters of each pair $\tilde{W}_l$ by using the adaptive gradient ascent procedure described in Algorithm 1.

Here λ_f and λ_v are positive learning rates, and ε_v and ε_f are small positive increments used to estimate the derivatives of the function F. This function F is the fitness function that we want to maximize and that evaluates the discriminative power of a given pair $\tilde{W}_l$. This fitness function is equal to the statistics obtained with a statistical test that compares the current density in $\tilde{\Omega}_l$ and $\tilde{\Phi}_l$ between the different classes. More precisely, for each training record $m(t)$, this record is first band-pass filtered in the $\tilde{\Phi}_l$ frequency band by using the corresponding IIR or FIR filter. Then, the current density in $\tilde{\Omega}_l$ is computed using Eq. 2.11, and averaged over a given time window and log-transformed as in Eq. 2.7. One should note that the obtained values $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$ depend on the values of σ_v and σ_f which are used to compute the band-pass filter and the ROI current density. These $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$ are then arranged by class label. The statistical analysis finally compares the mean value of these $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$ between the different classes, the null hypothesis H_0 being "the mean value of the $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$ is not different between the classes". The obtained statistics is used as the value of the fitness function F. Thus, algorithm 1 selects the values of σ_f and σ_v that maximize the discriminative power of a given $\tilde{W}_l$. Naturally, this procedure is performed for each pair W_l. It is worth noting that a gradient ascent optimization seems appropriate as we experimentally observed that the fitness function F was not monotonic. Rather, this fitness function generally had an optimum for small values of the σ parameter.

Implementation: For the fitness function F, we used as the statistical analysis a t-test for binary BCI and an ANOVA for multiclass BCI. These tests are the same ones as those used during the first training step (see 2.3.4). Concerning the values of $\varepsilon_f, \varepsilon_v, \lambda_f$ and λ_v, we performed extensive experimental tests and found that $\varepsilon_f = \varepsilon_v = 0.0001$ and $\lambda_f = \lambda_v = 10^{-5}$ were appropriate values. We used these values in all our experiments.

Algorithm 1 Adaptive gradient ascent algorithm (ε_f,ε_v,λ_f,λ_v)

$newF \leftarrow F(\sigma_v, \sigma_f)$
repeat
  {dealing with the frequency domain}
  $oldF \leftarrow newF$
  $\Delta F_f \leftarrow F(\sigma_v, \sigma_f) - F(\sigma_v, \sigma_f + \varepsilon_f)$ {estimating the partial derivative with respect to σ_f}
  $\sigma_f \leftarrow \sigma_f - \lambda_f \frac{\Delta F_f}{\varepsilon_f}$ {gradient ascent}
  $newF \leftarrow F(\sigma_v, \sigma_f)$
  {Adaptation of the learning rate λ_f}
  if $newF < oldF$ **then**
    $\lambda_f \leftarrow \frac{\lambda_f}{2}$
  else
    $\lambda_f \leftarrow \lambda_f + 0.1\lambda_f$
  end if
  {dealing with the space domain}
  $oldF \leftarrow newF$
  $\Delta F_v \leftarrow F(\sigma_v, \sigma_f) - F(\sigma_v + \varepsilon_v, \sigma_f)$ {estimating the partial derivative with respect to σ_v}
  $\sigma_v \leftarrow \sigma_v - \lambda_v \frac{\Delta F_v}{\varepsilon_v}$ {gradient ascent}
  $newF \leftarrow F(\sigma_v, \sigma_f)$
  {Adaptation of the learning rate λ_v}
  if $newF < oldF$ **then**
    $\lambda_v \leftarrow \frac{\lambda_v}{2}$
  else
    $\lambda_v \leftarrow \lambda_v + 0.1\lambda_v$
  end if
until $|\frac{\Delta F_f}{\varepsilon_f}| \leq 0.001$ and $|\frac{\Delta F_v}{\varepsilon_v}| \leq 0.001$

At the end of this offline training, a set of fuzzy ROI $\tilde{\Omega}_l$ associated to fuzzy frequency bands $\tilde{\Phi}_l$ has been identified. They can now be used for feature extraction, possibly online.

2.3.7 Feature Extraction with FuRIA

Once the training is achieved, feature extraction with FuRIA consists in computing the current density in each fuzzy ROI and fuzzy frequency band and in using these current density values as features. More formally, it consists in filtering the EEG signals $m(t)$, once for each one of the N_w fuzzy ROI $\tilde{\Omega}_l$ obtained, using the FIR or IIR filter corresponding to $\tilde{\Phi}_l$. Then, $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$, the current density in $\tilde{\Omega}_l$, is computed using Eq. 2.11 and averaged over a given time window and log-transformed as in Eq. 2.7. The N_w current densities $< \gamma_{\tilde{\Omega}_l,\tilde{\Phi}_l} >$ are then concatenated into a N_w dimensional feature vector $[< \gamma_{\tilde{\Omega}_1,\tilde{\Phi}_1} >, < \gamma_{\tilde{\Omega}_2,\tilde{\Phi}_2} >, \ldots, < \gamma_{\tilde{\Omega}_{N_w},\tilde{\Phi}_{N_w}} >]$. Such a feature vector can then be used as an input of any classifier, e.g., a Support Vector Machine, this classifier being in charge of estimating the class of $m(t)$. Figure 2.2 summarizes the principle of feature extraction using FuRIA.

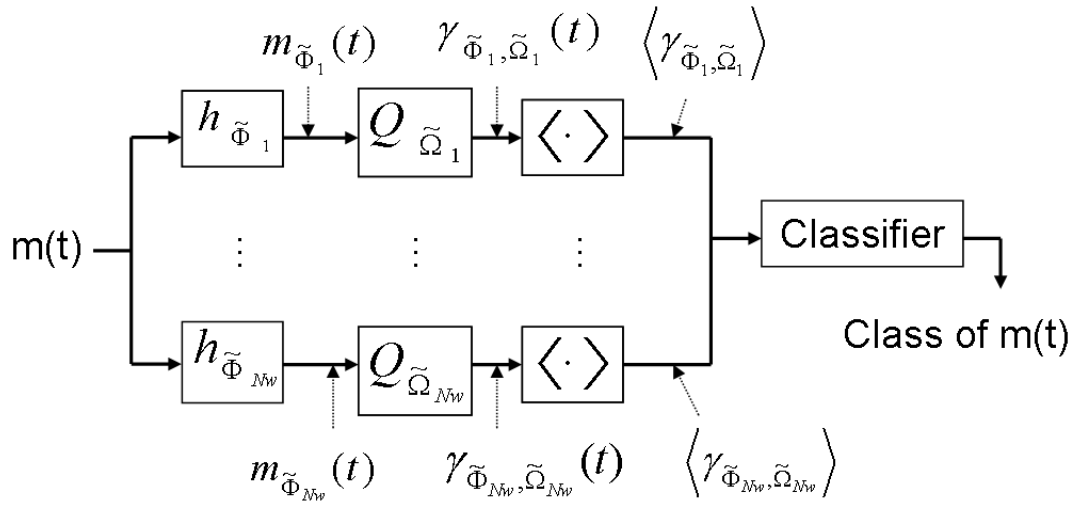


Figure 2.2: Feature extraction with FuRIA.

2.3.8 Model selection

The presented implementation of FuRIA has two hyperparameters: the threshold α used for the statistical analysis and the smoothing parameter H of the Mean Shift clustering algorithm. The values of these hyperparameters have an impact on the number of features N_w and on the extension and shape of the ROI and frequency bands. As this impacts the performance of the recognition algorithm, we must select the most appropriate hyperparameters. A solution could be to test several values for H and α and select the couple that enables the best classification on a training set, estimated using a given classifier and cross validation. However, we noticed that this method favors models with numerous features, which is not desirable. Furthermore, we observed experimentally that there were generally models with a classification accuracy only slightly lower than the best one, but with much less features. Indeed, if we plot the number of features versus the classification accuracy, the resulting curve tends to be relatively flat for large numbers of features, and suddendly decreases for a smaller number of features (see Figure 2.3).

Ideally, we would like to use the model corresponding to the point of the curve situated just before this sudden decrease of classification accuracy (point A, in blue, on Figure 2.3), as it would be the best tradeoff between a high classification accuracy and a small number of features. Indeed, we believe that such a model with few features should be prefered as it is probably more robust, less computationally demanding, more easily interpretable and it should ease the training of the classifier. Consequently, we propose a simple model selection criterion C_p:

$$C_p = 2 * Acc_{cv} - N_w \tag{2.16}$$

where Acc_{cv} is the accuracy (in percent) obtained using cross validation on a training set. The model with the highest C_p is the one that should be prefered. Thus, this criterion is still based

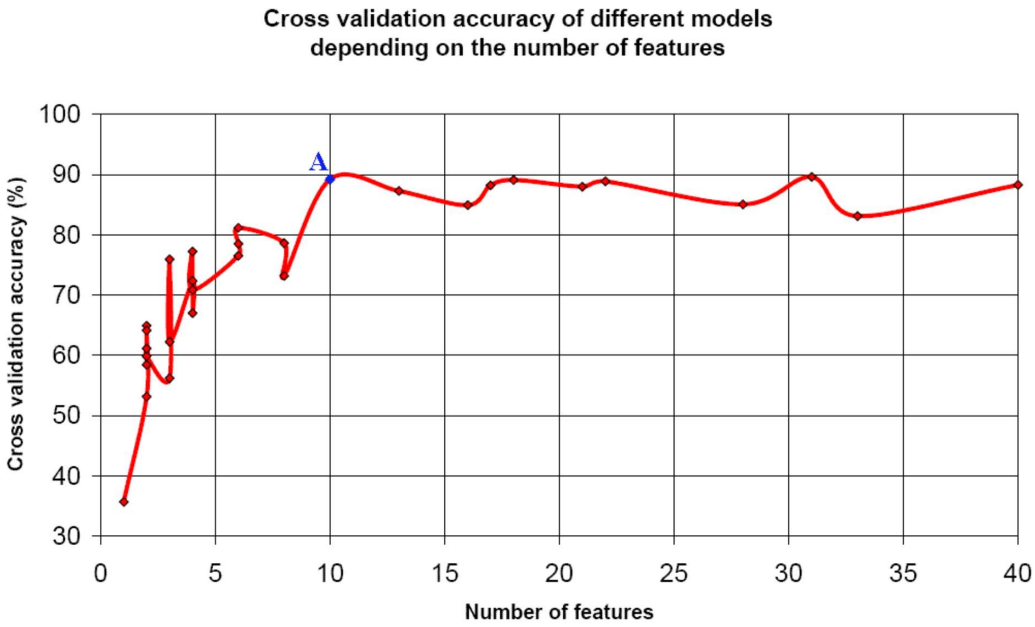


Figure 2.3: Example of a plot of the number of features versus the 10*10 fold cross validation accuracy of several models obtained with FuRIA. The point A corresponds to the model we would like to automatically select, as this model has the best tradeoff between a small number of features and a good classification accuracy. This plot corresponds to the data of subject S1 from the BCI competition 2005 (see Section 2.4.1.2).

on cross validation but it penalizes models with many features. We also consider that models with a number of features lower than the number of mental states should be avoided. Actually, we consider that a mental state is generated by at least one brain region. It should be noted that we also tested model selection criterions such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) [Zuc00]. Unfortunatly, they were not able to select the desired model as described above. If needed, the terms in the C_p criterion could be weighted in a different way so as to favor either models with a high cross-validation accuracy or sparse models. However, we observed that, as such, this criterion gave satisfactory results in practice. This is shown in the next section which is devoted to the evaluation of FuRIA.

2.4 Evaluations of FuRIA

These evaluations have two objectives. First we want to assess the impact of the different hyper-parameters as well as the contribution of the fuzzification processes on the performances. Then we also want to globally assess the efficiency of FuRIA, by comparing a BCI based on FuRIA with other state-of-the-art BCI systems used during BCI competitions.

In order to assess FuRIA, we evaluated it on four different subjects, available from two

data sets of the BCI competitions 2003 and 2005. For all these evaluations, we worked with a standard head model, composed of three concentric spheres and containing 2394 voxels. This head model has been obtained using LORETA-Key, a software dedicated to inverse solutions which has been developed by Pascual-Marqui (see [PM] for more details on this software). For the training of FuRIA, we considered the frequencies located in the 3-45 Hz frequency band with a step of 1 Hz between two consecutive frequencies. Concerning the MCR tests, we used a value of 1000 for the number of random permutations. The features were classified using a gaussian Support Vector Machine (SVM), as SVM is one of the most popular and efficient classifiers used for BCI [LCL$^+$07] (see also section 1.6.2.2). When dealing with multiclass problems, several SVM were combined using the One-Versus-the-Rest (OVR) scheme in order to design a multi-class classifier. The optimal SVM hyperparameters were selected using 10*10 fold stratified cross validation. The description of the data sets and the results obtained are presented in the following sections.

2.4.1 EEG data sets

2.4.1.1 BCI competition 2003 - data set IV

The first data set used was the EEG data set IV of the BCI competition 2003 [BMC$^+$04], provided by the Berlin group [BCM02]. These data contain EEG signals recorded while a subject was performing self-paced left and right finger tapping tasks. EEG signals were sampled at 100 Hz, recorded using 28 electrodes and comprised the 500 ms before the actual movement. 314 trials were available for training and 100 for testing. The goal of the competitors was to forecast, for each trial, the hand that was used. For this data set, we used FuRIA to learn and extract features on the last 250 ms time window of the data, i.e., we used $t_0 = 25$ and $N_s = 25$ in Eq. 2.7. According to several studies, this time window should be the most informative [BCM02, WZL$^+$04, CLL06]. According to a previous study on the same data set [CLL06], we chose a sLORETA regularization parameter $\alpha_R = 1000$. Here, we used FIR and IIR filters with 24 points and an order 8 respectively.

2.4.1.2 BCI competition 2005 - data set IIIa

The second data set used was the EEG data set IIIa of the BCI competition 2005 [BMK$^+$06], provided by the Graz group [SLBP05]. These data were recorded while three subjects S1, S2 and S3, were performing a 4-class motor imagery task. They were instructed to imagine left hand, right hand, foot or tongue movements. For both training and testing, 60 trials were available per class. Trials were sampled a 256 Hz and were recorded using 60 electrodes. Each trial lasted 7 seconds, without taking into account the inter trial periods of random lengths. The subjects were instructed to perform the motor imagery tasks during the last 3 seconds of each trial. For all subjects, we subsampled the data at 128 Hz, and used as time window for FuRIA these last 3 seconds, i.e., we used $t_0 = 512$ and $N_s = 384$ in Eq. 2.7. For training FuRIA, we ignored all trials contaminated by artifacts, leaving approximatly between 25 and 45 trials per class, depending on the subject. We used $\alpha_R = 100$ as the sLORETA regularization parameter. Here, we used FIR and IIR filters with 50 points and an order 10 respectively.

2.4.2 Evaluation of the influence of hyperparameters and fuzzification processes

In our implementation of FuRIA, two hyperparameters should be defined by the user: 1) α the significance threshold used in the statistical analysis of the first training step (see Section 2.3.4) and 2) H, the smoothing parameters used in Mean Shift during the second training step (see Section 2.3.5). In this section, we evaluated FuRIA for different values of the hyperparameter α, among {0.01, 0.05, 0.1, 0.25, 0.5}, and different values of the hyperparameter H, among {0.75, 1, 1.25, 1.5, 1.75, 2}. The goal was to assess the impact of these hyperparameters on the classification performances and on the number of generated features. Actually, we wanted to assess how FuRIA would behave and how it would accomodate voxels and frequencies with a very low discriminative power and how the fuzzification concept could use the available information. For all these data sets we also compared the results obtained when using FuRIA without the fuzzification process (i.e., using only crisp ROI and frequency bands), with only the spatial fuzzification, with only the frequential fuzzification and with the full (spatial and frequential) fuzzification. In the following, these four conditions will be denoted as "Raw", "Freq", "Space" and "All" respectively. We also computed the results for both FIR and IIR filters. For a matter of consiceness, only the results for the kind of filter that gave the best results are presented in this chapter. However, the complete and detailled results for all methods, subjects, conditions and filter types are displayed in Annex C. For each data set, the SVM classifier was trained on the features learnt and extracted by FuRIA on the training set, and used to classify the available test set using these features.

2.4.2.1 BCI competition 2003 - data set IV

Tables 2.1 and 2.2 display the mean classification accuracies obtained on the test set when using FuRIA with FIR filters, for different values of α and H respectively. The mean number of features (denoted as "NbFeat") is also displayed. In each table of this section, the best result for each condition and each subject is displayed in bold figures. The differences between the Raw condition and the fuzzy conditions, as revealed by a paired t-test, showed that all the fuzzy conditions performed better than the raw condition on average. However, this difference is only significant for the "All" ($p < 0.05$) and "Space" ($p << 0.01$) conditions.

Table 2.1: Data set IV, BCI competition 2003, test set: classification accuracy (%) and number of features for different values of α, averaged over the different values of H.

α	Average number of features	Raw	Freq	Space	All
0.01	10.17	**80.5**	**81**	82	80.33
0.05	10.17	79.5	80.67	82.67	**82.5**
0.1	9.17	77	78.17	**84.67**	81.83
0.25	9.67	76.67	77.17	83.17	76.17
0.5	8.83	72.17	74.83	78	77.33

Table 2.2: Data set IV, BCI competition 2003, test set: classification accuracy (%) and number of features for different values of H, averaged over the different values of α.

H	Average number of features	Raw	Freq	Space	All
0.75	31.8	**83.6**	**82**	**83.4**	**82.6**
1	12.6	80.2	79.6	82.6	80.8
1.25	6.4	78.2	79	82.6	79.2
1.5	2.8	74.4	76.8	80	78.4
1.75	2	73.2	76.6	82	78.2
2	2	73.4	76.2	82	78.6

2.4.2.2 BCI competition 2005 - data set IIIa

Tables 2.3 and 2.4 display the mean classification accuracies obtained for each of the three subjects on the test set, for different values of α and H respectively. Please note that, for subject S2, no results are presented for $\alpha = 0.01$ as no pair w_k was found significant with this threshold. Here again, only the results obtained with the best filter are displayed, that is IIR filters for S1 and S2 and FIR filters for S3. The complete results can be found in Annex C. The statistical differences between the Raw condition and the fuzzy conditions, obtained using a paired t-test, are displayed in Table 2.5.

Table 2.3: Data set IIIa, BCI competition 2005, test set: classification accuracy (%) and number of features for different values of α, averaged over the different values of H.

Subject	α	Average number of features	Raw	Freq	Space	All
S1	0.01	9.17	75.28	76.02	74.26	73.15
	0.05	9.33	**77.96**	**79.63**	76.76	**78.46**
	0.1	10.67	75.93	77.22	**79.17**	77.78
	0.25	7.83	60.93	67.96	72.31	68.98
	0.5	9	62.13	66.76	75.19	71.39
S2	0.05	7.83	**54.58**	44.31	**58.19**	55.278
	0.1	11.17	53.33	**50.97**	57.78	**56.11**
	0.25	11.67	50.97	43.89	47.22	45.69
	0.5	14.67	50.83	47.92	50.69	51.39
S3	0.01	6.17	71.67	69.72	71.94	68.75
	0.05	9.83	**73.06**	**76.94**	70.42	**77.78**
	0.1	9.17	70.42	72.08	**72.92**	72.5
	0.25	11.83	66.39	75	67.92	68.75
	0.5	12.5	64.17	71.39	68.47	67.08

Table 2.4: Data set IIIa, BCI competition 2005, test set: classification accuracy (%) and number of features for different values of H, averaged over the different values of α.

Subject	H	Average number of features	Raw	Freq	Space	All
S1	0.75	25.4	**86.22**	**88.67**	**87.44**	**89.33**
	1	12.2	84.22	85.67	86.56	85.89
	1.25	7.2	77.44	79.33	79.11	77.66
	1.5	5	70.89	73.67	76.11	72.66
	1.75	3.2	58.56	62.33	68.11	63.56
	2	2.2	45.33	51.44	55.89	54.56
S2	0.75	23.6	**61.39**	49.44	**60.83**	**56.94**
	1	13	58.96	51.46	58.33	56.25
	1.25	8.6	57.50	**55.28**	56.67	55.83
	1.5	5.2	54.44	53.06	55.56	55
	1.75	2.6	55.56	53.89	56.67	55
	2	1.4	56.11	45.83	54.44	50
S3	0.75	26	**81.67**	**80.17**	**83.5**	**81**
	1	13.4	78.33	80	80	80.17
	1.25	7.6	76	78.5	78.17	79.5
	1.5	5.6	74.33	78.5	77.83	79.83
	1.75	4	55.5	63.83	51.83	58.33
	2	2.8	49	57.17	50.67	47

2.4.2.3 Discussion

We performed a paired t-test to investigate the overall statistical differences between all the conditions, across all subjects and all hyperperparameter values. This revealed that globally, all fuzzy conditions performed better than the "Raw" one, and that this difference was significant ($p < 0.001$). This suggests that for a given set of ROI and frequency bands, fuzzifying them is likely to increase their classification performances. However, both the "Space" and "All" conditions performed significantly better than the "Freq" one ($p < 0.05$). There was no significant difference between these two conditions ($p > 0.05$), even though the "Space" condition gave slightly better results than the "All" one, on average. An interpretation is that the spatial and full fuzzifications are quite robust, as they increase performances on all subjects tested (except subject S2 for the "All" condition), even though this increase was sometimes not significant. On the contrary, the frequential fuzzification increases the performances on some subjects, sometimes more than the "Space" and "All" conditions (e.g., subject S3) and decreases them on other subjects. As such, the frequential fuzzification appears as less robust. We then suggest to use, by default, the spatial fuzzification only, or to try the different fuzzifications and select the most appropriate one for a given subject.

Concerning the effects of the threshold α, it can be noticed that the "Raw" condition

Table 2.5: Data set IIIa, BCI competition 2005: comparison of the raw versus the fuzzy conditions. A negative t-value indicates that the fuzzy condition is more efficient than the raw one.

subject	value	Freq	Space	All
S1	t-value	-3.56	-2.81	-2.20
	p-value	< 0.001	< 0.01	< 0.05
S2	t-value	4.15	-0.69	0.20
	p-value	< 0.001	> 0.05	> 0.05
S3	t-value	-2.82	-1.03	-1.61
	p-value	< 0.01	> 0.05	> 0.05

reached its best performances using the traditional values $\alpha = 0.01$ or $\alpha = 0.05$ whereas the fuzzy conditions generally reached their best results for the values $\alpha = 0.05$ or $\alpha = 0.1$. This suggests that the fuzzification process enables to use efficiently the information contained in less discriminant voxels and frequencies in order to improve the performances. Regarding the results for the hyperparameter H, the best accuracy is almost always obtained for the lowest value of H. However, accuracies very close to these ones can be obtained with higher value of H and hence, much less features. This supports the use of the previously proposed model selection criterion which penalizes models with a large number of features (see section 2.3.8).

We noticed that, depending on the subject, either FIR or IIR filters gave the best results. This highlights the well known inter-subject variability in the BCI field. Hence, preliminary tests must be performed in order to find the best filter for a given subject.

2.4.3 Comparison with BCI competition results

In the previous section, we have assessed the effects of the different fuzzification processes. In this section, we assess the global efficiency of the FuRIA features by comparing a BCI based on FuRIA features with BCI used by the BCI competition participants. To perform this comparison, we needed to select some parameters without the knowledge of the test set. Consequently, we only relied on cross validation scores on the available training sets. Thus, we selected the kind of filter (FIR or IIR) and the kind of fuzzification (spatial, frequential or both) according to their average cross validation score on the training set. To select the optimal hyperparameters H and α, we relied on the model selection criterion proposed in section 2.3.8, Equation 2.16.

2.4.3.1 BCI competition 2003 - data set IV

Based only on the training set, the parameter selection procedure described above found that $H = 1.75$, $\alpha = 0.05$, FIR filters and only the spatial fuzzification was the most appropriate configuration. This resulted in only 2 features. These parameters led to an accuracy of 84% on the test set, that is, exactly the same score as the winner of the competition [WZL$^+$04].

This suggests that the method is efficient, especially when considering the fact that only two features were used.

Fig. 2.4 displays the two ROI and frequency bands learnt by FuRIA. Interestingly, these ROI lie in the left and right motor areas, and the frequency bands lie in the β ($\simeq$ 13-30 Hz) band, which is consistent with the literature on the subject [WZL$^+$04, CLL06, PdS99]. This suggests that the FuRIA features are interpretable features, which can be used to check what has been learnt or even to extract knowledge about the brain dynamics.

14-31 Hz 14-28 Hz

Figure 2.4: The fuzzy ROI (in red) and their corresponding frequency bands that were automatically obtained by using FuRIA for data set IV of BCI competition 2003. The brighter the red color of the voxel, the higher the voxel degree of membership $\mu_{Sl}(v_j)$. The brain is seen from the top, nose up. These pictures were obtained with the LORETA-Key software [PM].

2.4.3.2 BCI competition 2005 - data set IIIa

Table 2.6 sums up the parameters used for each subject and selected using only the training set. The resulting number of features is also displayed. Concerning this data set, the goal of the

Table 2.6: Parameters used for data set IIIa from BCI competition 2005 (%)

	H	α	filter kind	fuzzification	feature number
S1	1	0.5	IIR	fuzzy space	11
S2	1	0.05	IIR	fuzzy space	12
S3	1.25	0.1	FIR	all fuzzy	5

participants was to provide a continuous classification, i.e., a class label for each time point.

However, this was still a synchronous BCI classification task, as the motor imagery tasks were performed during known time periods, here from second 4 to second 7 of each trial. Hence, the FuRIA features were first learnt on the time window from second 4 to second 7 of each trial. Concerning the classification of the test set trials, we classified each time point from second 4 to second 7 of each trial, as requested by the competition rules. We classified each point by using the FuRIA features extracted from the 1 second window preceding this point. Classification outputs were also aggregated across time, which is known to improve accuracy [LCL$^+$07, LSC06]. In order to do so we used a different multiclass Gaussian SVM (made of several binary SVM combined using the one-versus-the-rest scheme) for each single time point of the trial period from second 4 to second 7. These SVM were trained on the FuRIA features extracted on the 1 second window preceding their corresponding time point. Then, to classify a given time point, the final output $O_f(t)$ at time t was estimated by using the individual outputs $O_i(t)$ of the SVM corresponding to the previous time points situated from second 4 to the present time point:

$$O_f(t) = \sum_{k=0}^{t} w_k O_i(k) \tag{2.17}$$

where the outputs O_i are vectors containing the output of each SVM used in the one-versus-the-rest scheme (i.e., one output per class). Here, $t = 0$ corresponds to the second 4 of the trial. We defined $w_k = Acc_{cv} - 25$ with Acc_{cv} being the 10*10 fold cross validation accuracy (in percent) on the training set. Hence, this method is a weighted combination of classification outputs across time. These weights were chosen so as to ignore contributions from randomly performing classifiers (with a cross validation error of 25 %, as there are 4 classes) and emphasize contributions of well performing classifiers. The final class C_f attributed to a given point t was the one for which $C_f = argmax(O_f(t))$.

As continuous classification was used, a classification accuracy could be computed for each time point. The performance measure used was the maximal classification accuracy. Performances obtained using our methods as well as performances obtained by BCI competition 2005 participants on data set IIIa [Sch05] are reported for comparisons in table 2.7. All these participants also used SVM as classifiers (combined with other classifiers for the 2*nd* participant). For feature extraction, they all used a frequential information (band-pass filters or amplitude spectra) combined with spatial filters such as common spatial patterns, independant component analysis, principal component analysis and/or surface Laplacian [BMK$^+$06]. Results obtained by Schlögl *et al* when using an SVM as classifier and adaptive autoregressive parameters as features on the same data sets are also displayed [SLBP05]. However, these results are the leave-one-out cross validation accuracy over all the data, and not the classification accuracy on the test set. As such they should just be used for information, and not for comparison.

As shown by table 2.7, our method outperformed the one of the winner of the competition on 2 subjects out of 3 and even reached the best score among all participants on subject 3. However, the winner reached a really impressive score on subject 2 (the "worst" subject according to the general performances), leading him to the best overall results. We had to withdraw a large part of the EEG data from subject 2, as they were contamined by artifacts. This small amount of training data used may explain why FuRIA has been outperformed by the winner on

Table 2.7: Maximal classification accuracy for the test sets of data set IIIa from BCI competition 2005 (%)

	S1	S2	S3	Mean
Winner	86.67	**81.67**	85.00	**84.44**
2^{nd}	92.78	57.50	78.33	76.20
3^{rd}	**96.11**	55.83	64.17	72.04
FuRIA	90.56	69.17	**88.33**	82.68
Schlögl *et al* [SLBP05]	77.2	52.4	53.9	61.17

this subject. Globally, our method reached the second position, with a score only slightly lower than the one of the winner. This shows the efficiency of FuRIA, especially when considering the small number of features used.

Fig. 2.5 displays the fuzzy ROI and corresponding fuzzy frequency bands automatically learnt by FuRIA for subject 3. What can be noticed here is that the fuzzy ROI identified as relevant are located in the left and right motor areas, for frequency bands clearly within the μ (8-13 Hz) and β (13-30 Hz) rhythms. This is consistent with the literature on motor imagery [PdS99, PBSdS06, EGN03] which, again, enhances the interpretability of the extracted features. It should be noted that we have used, within sLORETA, a standard and non-realistic head model. Indeed, this model represents the head as three concentric spheres. In order to reach a more accurate and more exact source localization, and, as such, a better interpretability, it would be interesting to use a realistic head model which corresponds to each subject's anatomy. More precisely, it would be interesting to work with a head model generated from MRI (Magnetic Resonance Imaging) scans of each subject's head. Unfortunately, such scans were not available for these data.

2.5 Conclusion

This chapter has presented FuRIA (Fuzzy Region of Interest Activity), a trainable feature extraction algorithm for Brain-Computer Interfaces which is based on inverse solutions. This algorithm can be trained to automatically identify relevant regions of interest and their associated frequency bands for the discrimination of mental tasks, in binary as well as in multiclass BCI. To our best knowledge, FuRIA is currently the only method which can automatically identify relevant brain ROI and frequency bands for mental state classification in noninvasive BCI. This chapter also introduced the concepts of fuzzy ROI and fuzzy frequency bands which enabled to use efficiently the available information and, thus, to increase the classification performances.

The evaluation of the proposed method, using sLORETA as the inverse solution and an SVM as classifier, showed its efficiency. Actually, the obtained results were comparable with those of BCI competition winners. A possible interpretation is that the inverse solution, combined with the FuRIA training, acts as a spatial filter that removes the background activity and the noise not correlated with the targeted mental tasks. As such it focuses on relevant, subject-

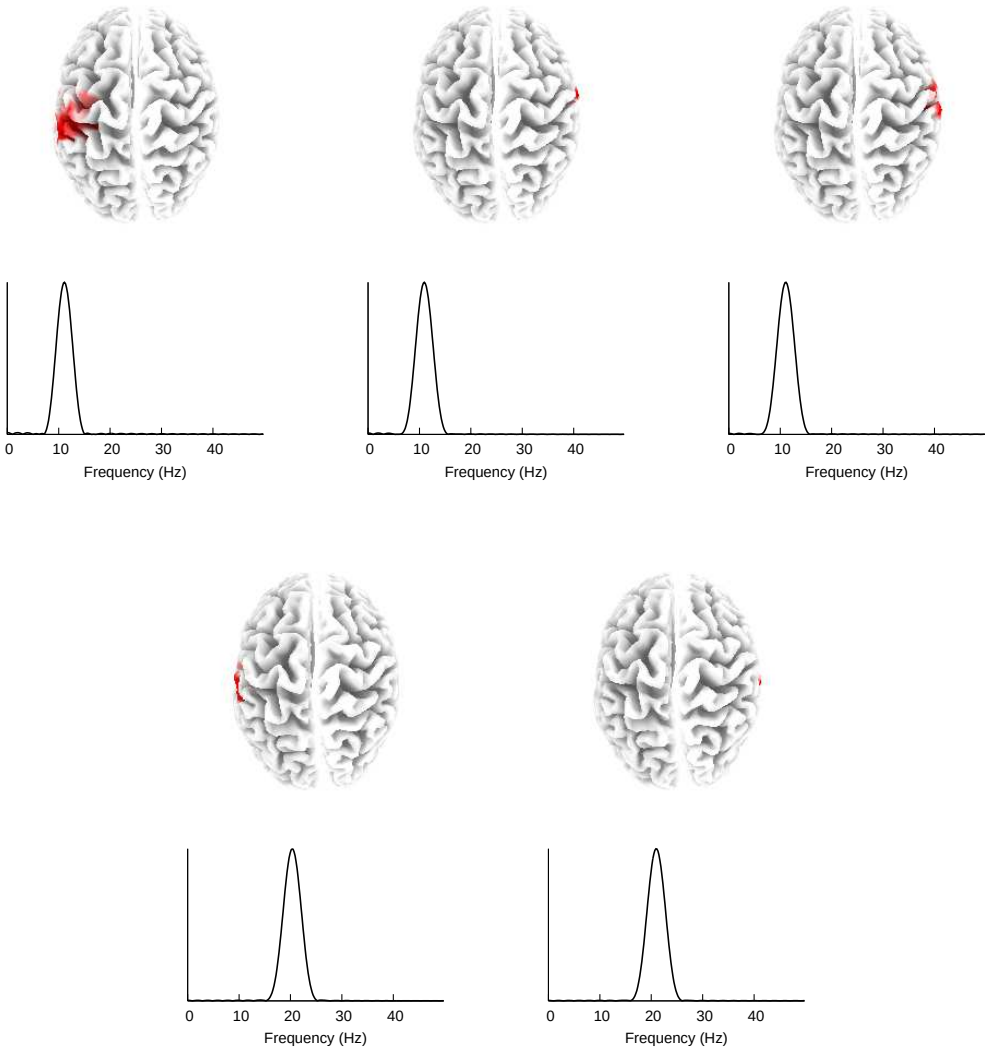


Figure 2.5: The fuzzy ROI (in red) and their corresponding fuzzy frequency band (which is equivalent to the filter magnitude response) that were automatically obtained by using FuRIA, on subject 3 from data set IIIa of BCI competition 2005. The brighter the voxel red color, the higher the voxel degree of membership $\mu_{Sl}(v_j)$. The brain is seen from the top, nose up. These pictures were obtained with the LORETA-Key software [PM].

specific, brain activity features. An additionnal advantage of FuRIA is the interpretability of the learnt and extracted features, which simply correspond to the activity in specific brain regions and frequency bands. As such FuRIA is a possible solution to a problem raised by the BCI community, namely, the lack of insights and interpretation that can be gained from currently

employed features [MAM+06]. The main drawback of FuRIA is its long training process. Indeed, the discriminative power of several voxels and frequencies should be investigated, which can be long if the number of voxels, electrodes or training data is large, and if time consuming statistical analysis such as the ones based on permutation tests are used. However, as this training is performed offline, this point does not seem critical. It should also be mentioned that in order to use an inverse-solution based method such as FuRIA, a relatively large number of electrodes is necessary.

By designing FuRIA, we have proposed a solution for building more efficient and interpretable BCI systems by working at the preprocessing and feature extraction level. In order to build a fully interpretable BCI system and possibly an even more efficient one, it is also necessary to work at the classification level. The next chapter of this manuscript is dedicated to this point.

Chapter 3

Classification: Studying the Use of Fuzzy Inference Systems for Motor Imagery-based BCI

3.1 Introduction

The BCI community has stressed the need to explore signal processing techniques which could lead to more efficient BCI from which we can extract knowledge [MAM+06]. The previous chapter of this manuscript tackled the problem at the preprocessing and feature extraction levels. In this chapter we focus on the classification part. More precisely, this chapter focuses on a classifier known as a Fuzzy Inference System (FIS) and study its use for classification in EEG-based BCI. Indeed, as mentioned in the introduction of this manuscript, FIS have several advantages which make them promising classifiers for BCI design. In this chapter, we study FIS for classifying popular EEG signals: motor imagery. More precisely, we study FIS on four points: (1) the performances of FIS (Section 3.4), for which we conducted a comparative analysis with other popular classifiers, (2) the interpretability of FIS (Section 3.5), (3) the possibility to add a priori knowledge to the FIS (Section 3.6), and (4) the ability of FIS to reject outliers (Section 3.7). The next two sections of this chapter describe respectively the specific FIS that we used in these studies as well as the data (EEG signals and features) employed. The four studies mentioned above are described in the four remaining sections.

3.2 Fuzzy Inference System employed: the FIS of Chiu

In the literature, numerous kinds of Fuzzy Inference Systems are available [Gui01, Nau97]. Among them, we chose the Chiu's FIS (CFIS) [Chi97a, Chi97b]. Indeed, CFIS is robust to noise, which is fundamental when dealing with such noisy data as EEG signals. Moreover, according to Chiu, the CFIS is generally more efficient than neural networks which are classifiers that have been succesfully used in numerous BCI studies [LCL+07]. Finally, it is a clustering-based FIS, making it suitable for dealing with small training sets [Gui01], which is also very relevant for BCI design [LCL+07].

3.2.1 Extraction of fuzzy rules

The fuzzy rules of a FIS are generally designed by hand or automatically extracted from data (or both). With the CFIS, three steps are required to automatically extract the fuzzy "if-then" rules from data: clustering of training data, generation of initial fuzzy rules and optimization of fuzzy rules.

3.2.1.1 Clustering of training data

A clustering algorithm known as "substractive clustering" [Chi97b] is applied to the training data of each class separately. The training data must be normalized in order to be bounded by a unit hypercube. The first step of the clustering algorithm consists in computing the potential P_i of each data point X^i, using the following equation:

$$P_i = \sum_{j=1}^{n} e^{-\frac{4}{R_a^2}||X^i - X^j||^2} \tag{3.1}$$

where n is the number of training data for the considered class, R_a represents the normalized radius of the clusters and must be specified by the user, and $||.||$ is the euclidean distance.

Then, the point X^k with the highest potential P_k^* is defined as the center of the first cluster. This computation ensures that the center (i.e., a point with a high potential) is a point that has many close neighbors. As such, it prevents outliers from being a center and/or modifying the position of the center.

The second step consists in revising the potential of all the data points according to the cluster center X^k obtained previously:

$$P_i \Leftarrow P_i - P_k^* e^{-\frac{4}{(1.25R_a)^2}||X^i - X^k||^2} \tag{3.2}$$

The point with the highest revised potential is then selected to be the next cluster center. The revising of the potentials prevents centers of clusters from being too close. This process of revision/selection is repeated until the potentials of all the data points are below a given threshold. Such a method enables to find automatically the number of clusters and their positions.

3.2.1.2 Generation of the fuzzy rules

A fuzzy "if-then" rule is generated for each cluster found previously. For a given cluster j, belonging to class Cl_i, the generated fuzzy rule is:

$$\textit{if } X_1 \textit{ is } A_{j1} \textit{ and } \ldots \textit{ and } X_N \textit{ is } A_{jN} \textit{ then class is } Cl_i$$

where N is the dimensionality of the data, X_k is the k^{th} element of a feature vector X and A_{jk} is a Gaussian membership function:

$$A_{jk}(X_k) = e^{-\frac{1}{2}(\frac{X_k - x_{jk}}{\sigma_{jk}})^2} \tag{3.3}$$

where x_{jk} is the k^{th} element of the vector representing the center of the cluster, and σ_{jk} is a positive constant which is initially the same for all A_{jk}. To increase accuracy, the membership functions can be "two-sided" Gaussians with a plateau and a different standard deviation on each side (see Fig. 3.1).

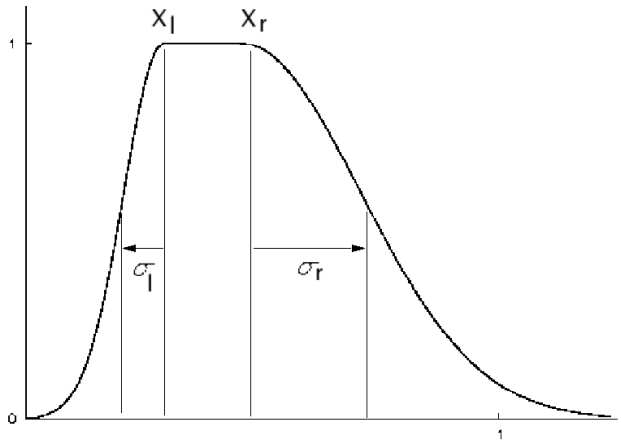


Figure 3.1: A two-sided Gaussian function, with means x_l and x_r and standard deviations σ_l and σ_r.

The degree of fulfillment $\mu_j(X)$ of a rule j is computed as follows:

$$\mu_j(X) = \prod_{k=1}^{N} A_{jk}(X_k) \tag{3.4}$$

Thus, the standard multiplication is used here as the *and* operator.

3.2.1.3 Optimization of the fuzzy rules

Last, each membership function A_{jk} is tuned according to gradient descent formulas [Chi97a]:

$$x_{jk} \Leftarrow x_{jk} \pm \lambda \mu_j \frac{(1 - \mu_{c,max} + \mu_{\neg c,max})(X_k - x_{jk})}{\sigma_{jk}^2} \tag{3.5}$$

$$\sigma_{jk} \Leftarrow \sigma_{jk} \pm \lambda \mu_j \frac{(1 - \mu_{c,max} + \mu_{\neg c,max})(X_k - x_{jk})^2}{\sigma_{jk}^3} \tag{3.6}$$

where λ is a positive learning rate which must be defined by the user, c is the class of the feature vector X, $\mu_{c,max}$ is the highest degree of fulfillment among the rules that assign X_k to the class c, and $\mu_{\neg c,max}$ the highest degree of fulfillment among the rules that do not assign X_k to the class c. Only the fuzzy rules corresponding to $\mu_{c,max}$ and $\mu_{\neg c,max}$ are optimized. In Equations 3.5 and 3.6, the "+" sign is used for the rule corresponding to $\mu_{c,max}$ and the "-" sign for the one corresponding to $\mu_{\neg c,max}$.

3.2.2 Classification

Once trained, the FIS can use its set of fuzzy rules to classify any new feature vector X. The class assigned to X corresponds to the class associated with the rule j for which $\mu_j(X)$ of Equation 3.4 is the highest.

3.3 Motor imagery EEG data

In order to evaluate the use of Fuzzy Inference Systems for BCI, we used popular EEG signals: motor imagery data. This section describes both the data sets of EEG motor imagery that we used and the features that we extracted from them.

3.3.1 EEG data

The data used for this study correspond to the EEG data set IIIb of the BCI competition 2005 [BMK+06]. This data set gathers the EEG signals recorded from three subjects who had to imagine left or right hand movements. Hence, the two classes to be identified were "Left" and "Right".

The EEG signals were recorded by the Graz group [PN01, SNP02], using bipolar electrodes at positions C3 and C4, and were filtered between 0.5 Hz and 30 Hz. We did not perform additionnal preprocessing before feature extraction. Subject 1 took part in a virtual reality experiment [LSL+04] in which the detection of left or right imagined hand movements triggered a camera rotation towards the left or right respectively, in a virtual room. Subjects 2 and 3 took part in a "basket" experiment in which the detection of left or right hand movements made a falling ball displayed on the screen, move towards the left or the right. The aim was to reach one of the two baskets located at the bottom left and bottom right of the screen [VSCP04].

For subject 1, 320 trials were available in the training set, whereas the test set was composed of 159 trials. For subjects 2 and 3, both the training and the test sets were composed of 540 trials. Each trial was 8 seconds long, and was divided as follows: during the first two seconds, a blank screen was presented to the subject. At second 3, a visual cue was presented to the subject in order to tell him which imagined hand movement he should start performing immediately. Finally, the data from second 4 to 8, for subject 1, or from second 4 to 7 for subjects 2 and 3, were used to provide feedback to the subject, according to the imagined hand movement detected. This feedback was either the rotation of the virtual environment, for subject 1, or the movement of the ball for subjects 2 and 3. More details about this data set can be found in [BMK+06].

3.3.2 Feature extraction method

For further classification, it is first necessary to extract features from these EEG signals. In order to do so, we chose to use Band Power (BP) features since such features are known to be efficient for motor imagery classification (see section 1.5.2.1) [PN01].

The main drawback of such features is that subject-specific frequency bands, in which the BP is to be computed, must be identified before use. Actually, the optimal frequencies for

discriminating between left and right hand movements vary from subject to subject [PN01]. Moreover, and independently from the features used, it is necessary to identify, for each subject, the optimal time window in which to extract the features in order to achieve maximal discrimination. This time window is located, for each trial, after the start of the feedback presentation, i.e., after second 4. It is indeed the period in which the subject is performing motor imagery.

3.3.2.1 Selection of optimal time window and frequency bands

In order to find the relevant time window and frequency bands, we used a method based on statistical analysis. It should be noted that these calibration steps were performed before entering the classification procedures with the aim of identifying the frequency bands and the time window to be used. Once identified, these frequency bands and the time window were used without modification in the classification procedure.

To identify the subject-specific frequency bands, we used a paired t-test which compared the BP means between both classes, for every 2 Hz wide frequency band between 1 Hz and 30 Hz, with a step of 1 Hz. As expected from the litterature [PN01], the frequencies for which the BP achieved the best discrimination were found in the μ and β bands, which supports the use of such features (see Figures 3.2 and 3.3).

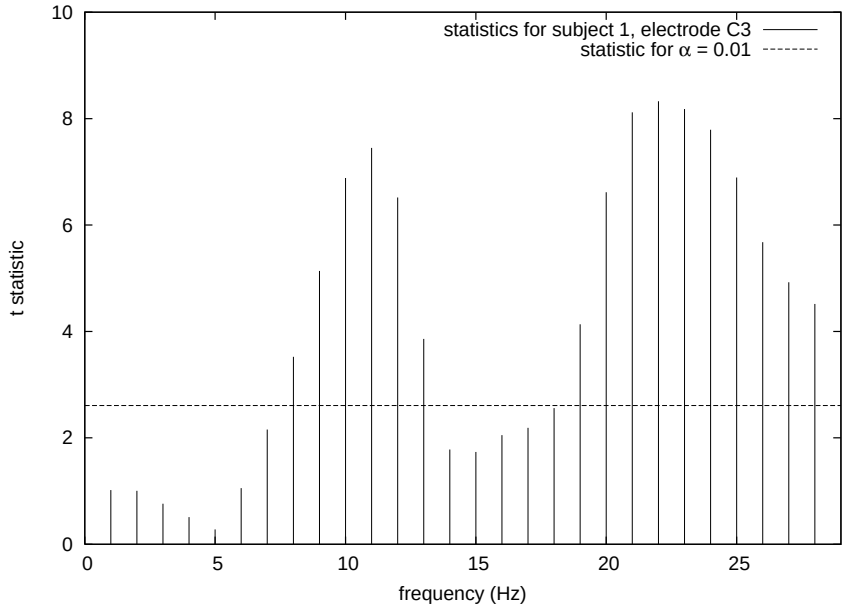


Figure 3.2: T statistics obtained with the BP features extracted for each frequency, from electrode C3 with Subject 1, in the optimal time window (see below for the determination of this time window). The dashed line represents the significance threshold for $\alpha = 0.01$.

Adjacent significant frequencies (with probability of type I error below $\alpha = 0.01$) were gathered into a single frequency band. Then, for every frequency band, a *shrinking* step was

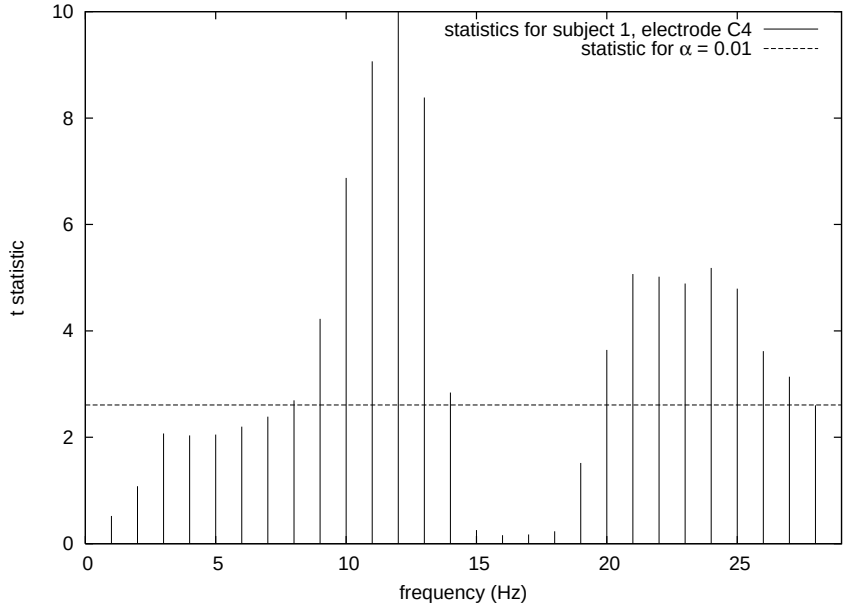


Figure 3.3: T statistics obtained with the BP features extracted for each frequency, from electrode C4 with Subject 1, in the optimal time window (see below for the determination of this time window). The dashed line represents the significance threshold for $\alpha = 0.01$.

performed which consisted in reducing the frequency band (making it 1 Hz shorter) and computing a new statistics for this band. If the new statistics was higher than the previous one, the shrunk frequency band was selected. The shrinking process was repeated until the statistics could not be increased any further.

To identify the optimal time window in which to extract the BP features, we performed the statistical analysis mentioned above for several time windows, and selected the one with the highest mean value of significant statistics.

3.3.2.2 Features extracted

The parameters used for BP feature extraction are summed up in Table 3.1. In this table, the window start value is given in seconds after the start of the feedback presentation.

Thus, this BP feature extraction method represents each trial by a four dimensional feature vector: $[C3_\mu, C3_\beta, C4_\mu, C4_\beta]$ in which Cp_y is the BP value for electrode Cp in the y band. These feature vectors will be used as input data for the following classification step.

3.4 First study: Performances

In this section we study the performance of the CFIS for motor-imagery classification. In order to do so, we compared the performance of CFIS with that of three classifiers widely used in the BCI community. Two performance measures were computed: 1) accuracy, i.e.,

Table 3.1: Parameters of band power feature extraction for each subject.

Subject	C3 μ band (Hz)	C3 β band (Hz)	C4 μ band (Hz)	C4 β band (Hz)	window start (s)	window length (s)
1	11	21-29	11-13	21-27	0.4	2.5
2	8-13	20-24	11-14	20-29	1.4	1.5
3	9-12	21-22	11-12	18-25	1.4	1.5

the percentage of correctly classified feature vectors and 2) the mutual information, which compares the amount of information carried by the classifier output and by the true class labels [SNP02, SKSP03].

3.4.1 Classifiers used for comparison

To assess the performance of the CFIS algorithm, we selected three popular classifiers widely used in the BCI community and which were shown to be the most accurate classifiers in several BCI experiments (see Section 1.6) [LCL$^+$07]. The first classifier was a non-linear SVM using a Gaussian kernel [Bur98]. The second classifier was a MLP with one hidden layer and sigmoid activation functions [Bis96]. The third classifier was a Perceptron, i.e, a Linear Classifier (LC) equivalent to LDA [DHS01].

The implementation of these three classifiers was achieved using the Torch C++ library [CBM02]. The optimal values for the hyperparameters (radius R_a for the CFIS, regularization parameter C for the SVM, etc.) of MLP, LC, SVM and FIS were chosen using 10-fold Cross Validation (CV). Two-sided Gaussians were used as membership functions for the CFIS. These functions were found to give the best generalization performance on the training data. This generalization was also estimated using 10-fold CV. The four classifiers were trained on the training sets of the motor imagery data described in Section 3.3, using the features aforementioned.

3.4.2 Accuracy and Mutual Information

Table 3.2 displays the average accuracy and Mutual Information (MI) [SNP02] obtained by each classifier on each subject's test set.

In terms of accuracy, the results showed that CFIS outperformed LC and reached similar results as SVM and MLP. Concerning MI, the results showed that CFIS performed better than SVM and LC and was outperformed by MLP.

3.4.3 Conclusion

The accuracy reached by the CFIS algorithm makes it suitable for motor imagery-based BCI applications. Indeed, concerning accuracy, the CFIS reached the same level of performance as

Table 3.2: Accuracy (%) and Mutual Information (MI) of classifiers

	Subject	CFIS	SVM	MLP	LC
Acc.	1	86.7±1.6	**86.8±0.0**	86.6±0.3	84.1±0.9
	2	74.7±1.5	**75.9±0.5**	75.5±0.1	71.8±1.8
	3	**75.7±0.6**	75.4±0.5	74.6±0.1	72.7±2.0
	Mean	79±1.2	79.4±0.3	78.9±0.2	76.2±1.6
MI	1	0.49±0.07	0.37±0.12	**0.63±0.03**	0.45±0.03
	2	0.17±0.00	0.20±0.00	**0.29±0.02**	0.19±0.00
	3	0.26±0.01	0.24±0.03	**0.29±0.04**	0.20±0.03
	Mean	0.30±0.03	0.27±0.05	0.40±0.03	0.28±0.02

the most popular classifiers used for BCI design. It should also be noted that, due to its relative computational simplicity, the CFIS is suitable for real-time use.

3.5 Second study: Interpretability

As seen previously, FIS are classifiers that are known to be interpretable, i.e., the rules they automatically learn to classify data can be read and interpreted by the user. In this section, we study this interpretability in to order to assess if it can be of practical use for BCI purposes. In order to do so, we will focus on the fuzzy rules extracted by the CFIS on our motor imagery data.

3.5.1 Extracted fuzzy rules

The rules automatically extracted by the CFIS from the EEG data of subject 1 are displayed in Figure 3.4. In this Table, each row represents a fuzzy rule and each column represents a feature. As such, the function displayed in row j and column k is the fuzzy membership function A_{jk}, i.e., the membership function for rule j and feature X_k. Interestingly enough, only two fuzzy rules were extracted for each subject.

3.5.2 Interpretation

The interpretation of the extracted rules displayed on Figure 3.4 is that the power for electrode C3 in the μ and β bands during imagined right hand movements is smaller than that during imagined left hand movements. A symmetric behaviour can be observed for electrode C4. In EEG research, this phenomenon is known as contralateral Event Related Desynchronisation (ERD) [PK92]. Actually, it is known that when a subject imagines a movement of one of his hands, there is a decrease of energy in his motor cortex from the opposite side of the hand used [PK92].

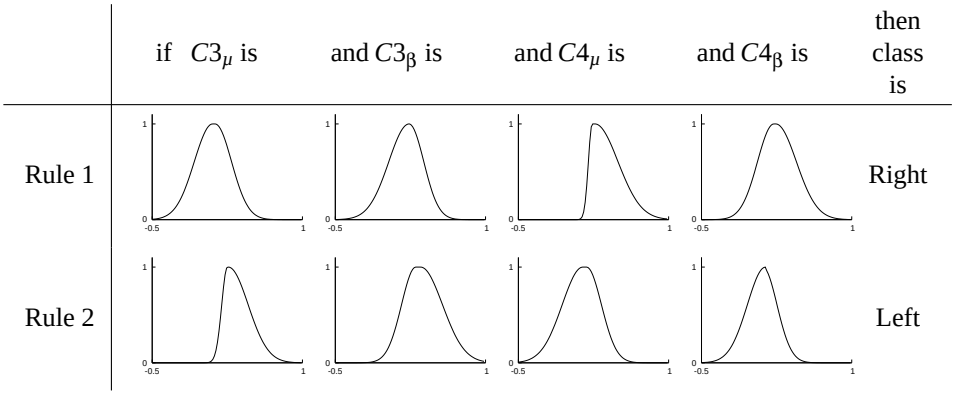


Figure 3.4: Fuzzy rules automatically extracted by CFIS for subject 1.

3.5.3 Conclusion

The interpretability of the CFIS has been studied here on properties of brain dynamics that have been known for a long time. Even though no *a priori* knowledge was used for the training phase, the CFIS was still able to automatically extract relevant information about ERD and to display it in an understandable way. Such a possibility could be used to automatically extract novel knowledge about the brain dynamics in BCI experiments. As the brain is still far from being fully understood, this property appears to be very interesting [MAM+06].

3.6 Third study: Adding a priori knowledge

As mentioned earlier, an interesting property of FIS is the possibility they offer to design and use Hand-Made Fuzzy Rules (HMFR) [Men95]. Indeed, as FIS algorithms are made of a set of fuzzy rules, it is possible to add any kind of supplementary fuzzy "if-then" rule to them. For instance, hand-made rules designed by experts in the field can be used as *a priori* information and added to the set of rules extracted automatically. This section reports the study we conducted on this point.

3.6.1 Conception of "hand-made" fuzzy rules

One *a priori* knowledge concerning hand motor imagery EEG data concerns the presence of contralateral ERD in the μ and β bands [PK92]. Specific hand-made rules that reflect the meaning of ERD could state that if the activity in the μ and β bands is higher in one electrode (C3 or C4) than in the other, then it means that the subject is imagining a movement of the hand of the same side. Such rules could be formalized by a human expert as follows:

Rule 1: if $C4_\mu > C3_\mu$ and $C4_\beta > C3_\beta$ then class is Right
Rule 2: if $C3_\mu > C4_\mu$ and $C3_\beta > C4_\beta$ then class is Left

To describe $(Cp_y - Cq_y) > 0$ (i.e., $Cp_y > Cq_y$) using membership functions, we could use the following function h:

$$h(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \tag{3.7}$$

However, h is a crisp function, i.e., it is not a fuzzy function. To use a fuzzy function which range is in [0,1] and not in {0,1} it is possible to introduce the well-known sigmoid function g:

$$g(x) = \frac{1}{(1+e^{-\lambda x})} \tag{3.8}$$

This function describes the same relationship as h, but in a "fuzzier" way. Besides, when $\lambda \to \pm\infty$ then $g \to \pm h$. Finally, two HMFR using g as membership functions can be designed to discriminate left or right imagined hand movements as displayed on Fig. 3.5.

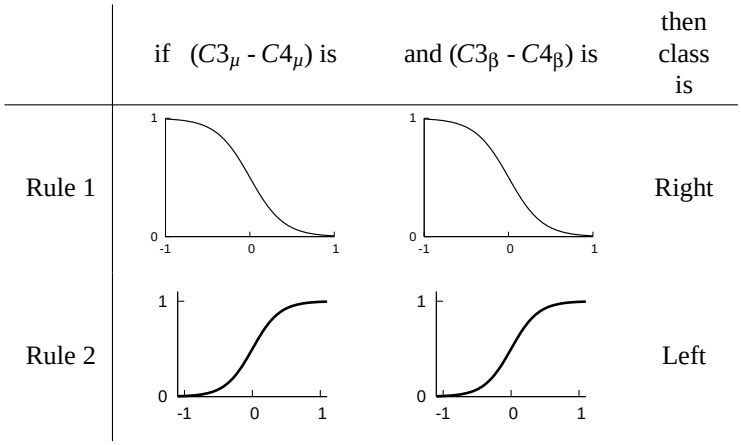


Figure 3.5: Hand-made fuzzy rules to classify motor imagery data

Using schemes of trials and errors on the training sets, the optimal value for λ was chosen to be ± 21500 in the four membership functions. It should be noted that such rules cannot be learnt by the CFIS as they describe relationships between features and not the properties of the features.

3.6.2 Performance

We computed the accuracy of the proposed HMFR, as well as the average accuracy of a CFIS that contains both automatically extracted rules and HMFR (see Table 3.3). This last classifier is denoted as CFIS+HMFR in the following. One should note that the accuracy obtained with HMR is not an average value. There is no need to train the classifier when using HMR. Consequently, the FIS classifier with hand made rules will always reach the same accuracy on a given data set.

Table 3.3: Accuracy (%) of trained CFIS versus FIS made of HMFR

Subject	1	2	3
HMFR	87.4%	66.5%	72.6%
CFIS	86.7±1.6	**74.7±1.5**	**75.7±0.6**
CFIS+HMFR	**88.1±0.4**	72.4±1.0	75.6±0.6

These results first show that the HMFR accuracy is much higher than chance, which means that efficient HFMR using a priori knowledge on motor imagery can be designed and can classify imagined hand movements.

The performance of the CFIS+HMFR classifier are contrasted. Indeed, it reached a higher accuracy than CFIS on Subject 1. This score is even higher than any other classifier used in Section 3.4.2 such as MLP or SVM. The result is opposite on Subject 2, as accuracy of CFIS+HMR is smaller than that of CFIS. No significant difference can be observed on Subject 3. These results can be related to the accuracy obtained by the HMFR alone. Actually, it seems that the HMFR giving good results alone can lead to an even better classifier when combined with automatically extracted rules, as seen with Subject 1 for instance. On the other hand, the HMR giving relatively poor results alone, would probably reduce the performance of the classifier when combined with automatically extracted rules, e.g., for Subject 2.

3.6.3 Conclusion

This study suggested that it was possible to design HMFR that represent a priori knowledge on motor imagery and use them within a BCI. It also seems that if the designed HMFR are efficient, they may increase the performance of a FIS by being added to automatically learnt rules. This result should be relevant to BCI research as it means FIS may be able to exploit knowledge that is present in the vast literature about EEG [MAM+06].

3.7 Fourth study: rejection of outliers

Outliers are feature vectors that do not correspond to any of the targeted mental tasks. They are commonly recorded during a BCI experiment. These outliers should not be classified since they are likely to be associated with a wrong class label. To reduce the error rate, outliers should be rejected, i.e., identified as not belonging to any one of the targeted classes. Rejecting outliers can also be particularly important in asynchronous BCI experiments in which the subject can think about anything but the targeted mental tasks during the so-called “non-control” state (also known as “idle” state) [TGP04, MKH+06].

In this section, we study the outlier rejection capabilities of the CFIS algorithm as compared to other classifiers used in previous BCI experiments: SVM, MLP and LC classifiers described in Section 3.4.1.

3.7.1 Method

To evaluate the CFIS outlier rejection capabilities, we generated artificial outliers by randomly placing new feature vectors at a large distance from the feature vectors of the test sets. This distance was selected randomly from 2 to 4 times the standard deviation of all feature vectors. This technique ensures that outliers lied outside the pattern of each class, as a reference to the definition of an outlier given by Moore *et al* [MM99]. We added 25% of such outliers to each test set. A third class label was assigned to the outliers to ensure their classification would increase the error rate.

For the CFIS, a feature vector was rejected if the highest degree of fulfillment was smaller than a given threshold. For both SVM and LC, rejection occured if the absolute value of their output was smaller than the given threshold. For the MLP, rejection was performed if the largest output value was smaller than the threshold. All classifiers were trained on the training set of each subject. Then, we computed the error-reject curves for these classifiers [JDM00] on each subject's test set with added outliers. These curves were computed by gradually increasing the value of the rejection thresholds and computing the error and reject rate for all these values. The classification error rate was defined as being the percentage of vectors assigned to a wrong class.

3.7.2 Results

The error-reject curves for each subject are displayed on Figures 3.6, 3.7 and 3.8. On these curves, the X-absciss corresponds to the reject rate and the Y-ordinate to the classification error rate.

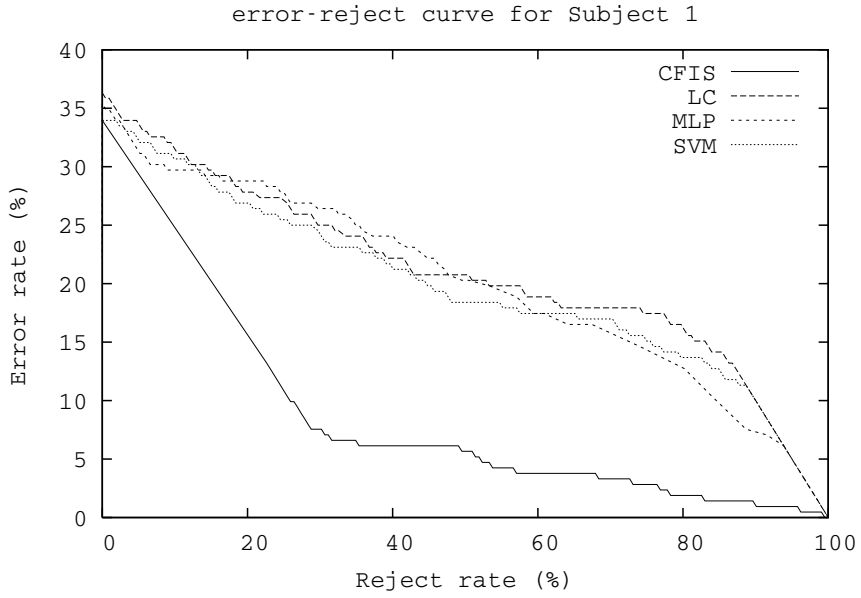


Figure 3.6: The error-reject curves for each classifier, on data of subject 1.

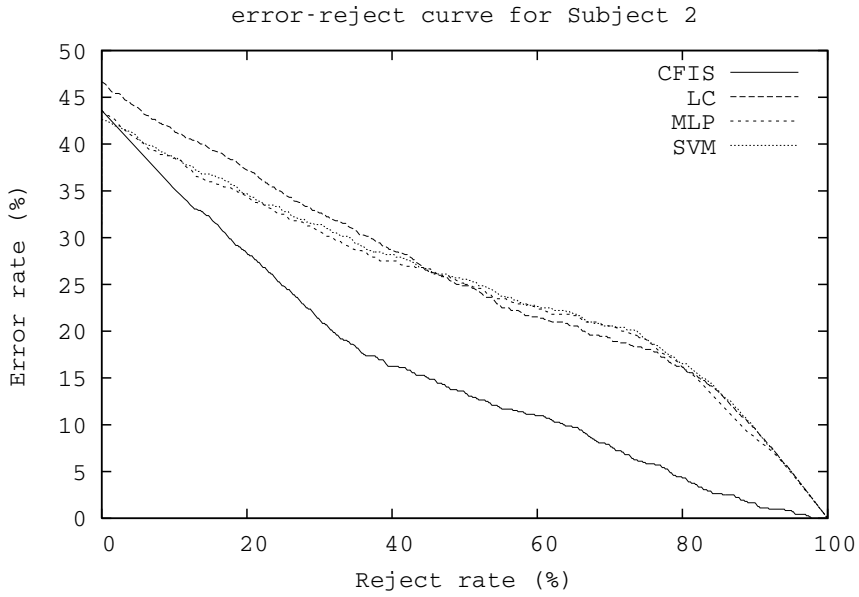


Figure 3.7: The error-reject curves for each classifier, on data of subject 2.

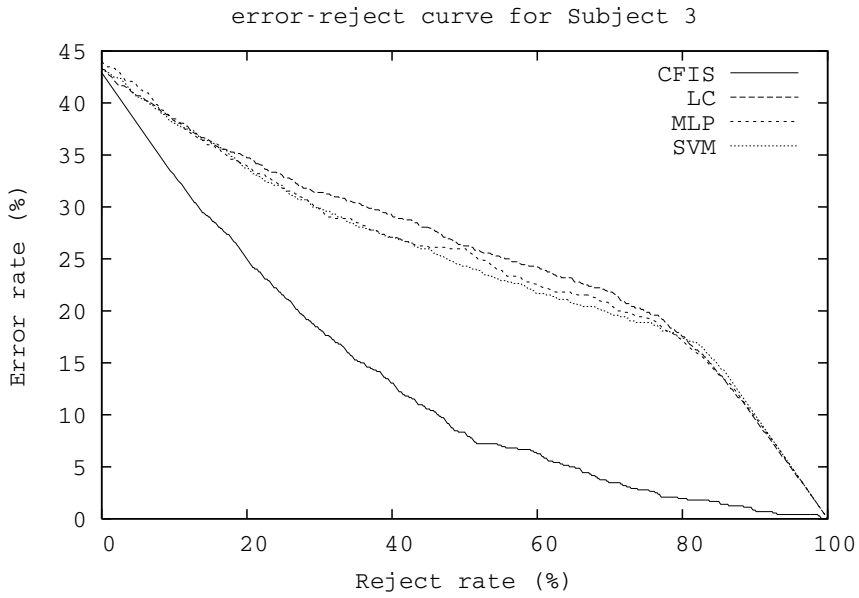


Figure 3.8: The error-reject curves for each classifier, on data of subject 3.

The error-reject curves for MLP, LC and SVM suggest that these classifiers must reject a lot of feature vectors before reaching a low error rate which means they cannot make a clear distinction between outliers and regular vectors. On the contrary, the area below the CFIS curves is much smaller than the one of any other classifier. As an example, CFIS can reach an

error rate of about 7% when rejecting 28% of the data of Subject 1. To reach the same error rate, MLP, Perceptron and SVM have to reject nearly 93% of the data. This shows that CFIS is able to identify and reject efficiently the ouliers, which makes its error rate drop dramatically.

These differences can be explained by the fact that CFIS is a generative classifier whereas MLP, LC and SVM are discriminative classifiers (see section 1.6.1). Indeed, discriminant classifiers can only identify to which of the targeted classes the feature vector most likely belongs to, and not if it actually belongs to one of these classes or not.

3.7.3 Conclusion

The results of the rejection tests suggest that the CFIS algorithm was able to identify the outliers more efficiently than the MLP, LC and SVM classifiers. In presence of outliers, CFIS could reach a small error rate when rejecting only few feature vectors. These tests were performed on artificial outliers, and as such, should be further confirmed on real data. However, the clear difference between the error-reject curves of CFIS and that of the other classifiers makes CFIS a very promising classifier for outlier rejection in motor imagery-based BCI.

3.8 Conclusion

In this chapter we have studied the use of a Fuzzy Inference System (FIS) for motor imagery classification in BCI. We first studied the performance, in terms of classification accuracy and mutual information, of the Chiu's FIS (CFIS) [Chi97a]. It reached similar results than the most popular classifiers used in BCI. It should be mentioned that the interest of FIS in terms of classification performance for BCI has been noticed simultaneously by our group [Lot06] and another group [HPM06]. Second, we stressed the interpretability of the CFIS, which could be used for brain knowledge discovery. Third, we studied the possibility of adding a priori knowledge to the CFIS under the form of Hand-Made Fuzzy Rules (HMFR). Suitable HMFR were shown to improve the performance of a trained CFIS in some cases. Finally, the CFIS capabilities of rejecting outliers were assessed on artificial data, showing its superiority over classifiers commonly used for BCI. Taken together, our results suggest that FIS classifiers are promising for BCI as they address several issues raised in the community, such as the need for interpretable classifiers to which a priori knowledge could be added [MAM$^+$06].

The fact that FIS are interpretable classifiers, providing that their input features are also interpretable, makes such a classifier a very promising companion for the FuRIA features. Indeed, combining interpretable features, such as FuRIA features, with an interpretable classifier, such as CFIS, may lead to an efficient and fully interpretable BCI system. Such a system would be able to recognize the different mental states while, at the same time, informing the BCI designer about the brain regions and brain rhythms (frequency bands) involved. This system could also inform the BCI designer about which activity in these regions and rhythms corresponds to which mental state. These points will be highlighted in the next chapter (Chapter 4).

Chapter 4

Towards a Fully Interpretable BCI System: Combining Inverse Solutions with Fuzzy Logic

4.1 Introduction

In Chapters 2 and 3, we have proposed and studied feature extraction based on inverse solutions and classification based on fuzzy inference systems. We have shown that both methods could provide insights on how the brain works, and as such we suggested that they could be used to design an interpretable BCI system. In this chapter we propose an algorithm, which is based on inverse solutions and fuzzy inference systems, to design fully interpretable BCI systems. The proposed algorithm goes beyond a combination between FuRIA and fuzzy inference systems. Indeed, this algorithm also makes it possible to express the knowledge automatically extracted by the BCI using simple words. As such, it should enable people not familiar with classifiers or fuzzy inference systems to understand what has been automatically learnt by the BCI. This chapter is organized as follows: Section 4.2 investigates the limitations of methods currently proposed to extract knowledge from BCI. Section 4.3 describes the algorithm we propose in order to design a fully interpretable BCI. Finally, Section 4.4 reports on an evaluation of our method on two kinds of EEG data.

4.2 Extracting knowledge from current BCI systems

Despite the promising possibilities offered by an interpretable BCI [MAM+06, KFP93], very little has been done in the literature to design such a BCI. A few works have explored the possibility to extract knowledge about the brain from what the classifiers have learnt. Typically, this has been achieved by studying the weights of a trained neural network [AS96], the nodes of a decision tree [KFP93] or the weights of a trained support vector machine [BJL+08]. This could give insights on what were the most relevant features, and indirectly, on what were the most relevant channels and/or frequency bands. Although this information appears as interesting, it requires the analysis of tens or hundreds (sometimes thousands) of features, and does

not provide a concise, straightforward and easy to understand explanation of what has been learnt by the BCI. Moreover, such methods can report on the most relevant features, but may not explain which values for these features correspond to which mental state.

Similarly, feature selection techniques have been applied to interpretable BCI features and analyzed in order to gain some knowledge from the data analysis method [BCM02, dRMFM$^+$02]. As in works previously mentioned, such methods can identify the most relevant features and, as such, the most relevant channels and/or frequency bands. However, these methods cannot explain the relationship between these feature values and the mental states performed. In other words, such methods may not be used to give insights about what is happening in these relevant channels or frequency bands.

4.3 An algorithm to design interpretable BCI

In this chapter, we propose an algorithm to design an interpretable and efficient BCI system. The proposed method can report on what are the relevant brain regions and frequency bands involved in the mental states used in the BCI, and can also report on which activity in these regions and frequency bands corresponds to which mental state. Moreover, our method can report on all this information in a synthetic way and *by using simple words*. In order to do so, our method first combines efficient and interpretable features with an efficient and interpretable classifier. Then it relies on the "computing with words" framework of Zadeh [Zad96b] in order to increase the interpretability of the system by using words instead of numbers. This section first proposes an overview of our method, then it details the 3 steps composing this algorithm, namely, extracting interpretable features, using an interpretable classifier and improving the interpretability.

4.3.1 Overview

Our method to design an interpretable BCI can be divided into three steps:

1. **Feature extraction:** To obtain an interpretable BCI, we first need to use interpretable features. As interpretable and efficient features, we used inverse solution-based features and more especially FuRIA features (see Chapter 2).

2. **Classification:** Similarly, to obtain a fully interpretable BCI, the classifier should also be interpretable and should not behave as a black box. We need a classifier which can report on which input feature values correspond to which output class, i.e., to which mental state. Consequently, we used as classifier a Fuzzy Inference System (FIS) and more precisely, the Chiu's FIS (CFIS) (see Chapter 3).

3. **Improving interpretability:** Interpreting means reasonning, and humans are more used to reason with words than with numbers. Consequently, the last step of our method consists in performing linguistic approximation, i.e., in presenting what has been learnt by the system using words rather than numbers.

These three steps are described in more details in the next sections and are schematized, along with an artificial example, in Figure 4.1.

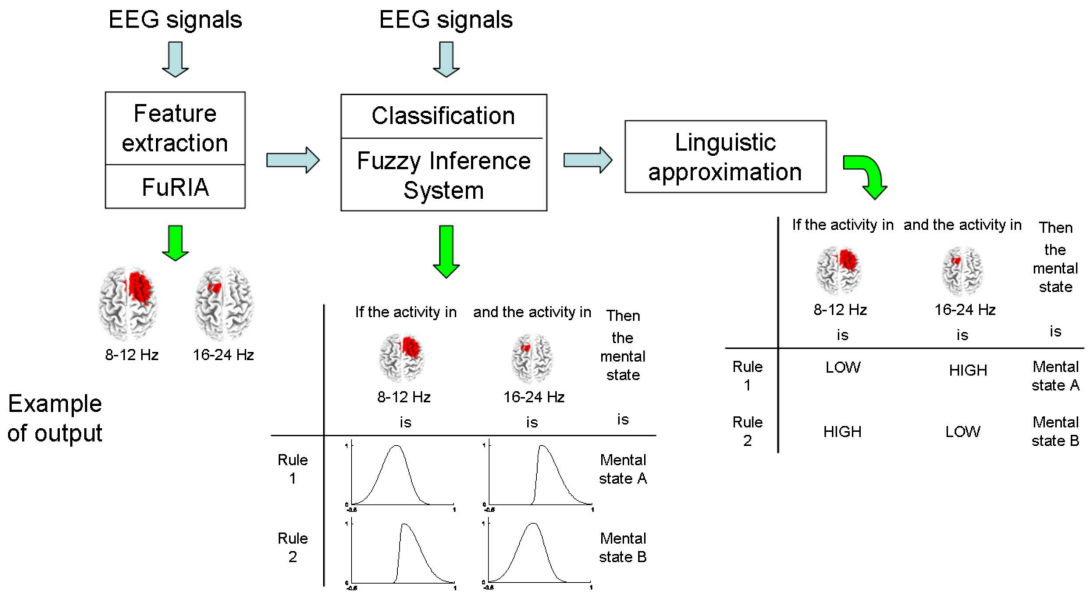


Figure 4.1: Schematic representation of the proposed algorithm towards the design of an interpretable BCI. An artificial example is also provided to ease the understanding. In the example tables provided, each row corresponds to an "if-then" rule, and each column to a feature. These rules describe the feature values for the mental state they infer.

4.3.2 Feature extraction: FuRIA features

As interpretable and efficient features, we selected features learnt and extracted by the FuRIA algorithm. As mentioned in Chapter 2, features extracted with FuRIA correspond to the activity in a few brain regions and their associated frequency bands. This information corresponds to a relevant physiological information which can already give insights on how the brain works. Moreover, FuRIA generally extracts a small number of relevant features, i.e., a small number of brain regions and frequency bands. This is also interesting as too many features would mean too much information to be analyzed by the human, and as such, the interpretation would become difficult. However, on their own, such features cannot report on which values of these features are related to which mental state. To obtain this information, it is necessary to input these features into an interpretable classifier.

4.3.3 Classification: the Chiu's Fuzzy Inference System

In order to classify the FuRIA features we selected the CFIS classifier. Let us recall that, as all FIS classifiers, the CFIS can automatically extract fuzzy "if-then" rules from data and can use these rules to classify new input data. With the CFIS the form of the j^{th} fuzzy rule is as follows (see Chapter 3 for details):

$$\text{If } X_1 \text{ is } A_{j1} \text{ and } \ldots \text{and } X_i \text{ is } A_{ji} \text{ and } \ldots \text{and } X_N \text{ is } A_{jN} \text{ Then Class is } C_j$$

In our BCI, the X_i are the FuRIA features i.e., the activity in a given brain region and a given frequency band, and C_j is a given mental state. This formalism based on rules makes it possible to interpret what has been learnt by the classifier. Indeed the extracted rules can report on which activity in each relevant brain region and frequency band corresponds to which mental state (see Chapter 3). Combining these FuRIA features and this CFIS classifier leads to the final BCI we used to classify the mental states. The resulting BCI could already be interpreted by any BCI researcher who is familiar with fuzzy membership functions. However, despite the interpretability already attained, it should be recognized that these fuzzy membership functions A_{ji} do not provide the most simple and straightforward way to interpret the rules. Indeed, these membership functions are mathematical functions and, as such, express knowledge with numbers. In order to ease the interpretability of the system, we performed a last step in our method: the linguistic approximation. This should enable us to display the automatically extracted knowledge in a way understandable by persons who do not know anything about fuzzy membership functions.

4.3.4 Improving interpretability: linguistic approximation

Zadeh, the creator of fuzzy sets and fuzzy logic, stated that "the main contribution of fuzzy logic is a methodology for computing with words. No other methodology serves this purpose" [Zad96b]. Interestingly enough, humans tend to reason with words, which words generally correspond to a fuzzy definition. Fuzzy logic can express the fuzziness of these definitions by using linguistic terms. A linguistic term is actually a fuzzy set that describes a word. In [Zad96b], Zadeh defined the process which consists in replacing a set of rules based on non-linguistic fuzzy sets (i.e., a set of rules based on numbers) by a set of rules based on linguistic terms and their associated fuzzy sets. This process is known as a "linguistic approximation", or, equivalently, as a "retranslation process". The fuzzy sets used and learnt by the CFIS do not correspond to linguistic terms. As such, performing a linguistic approximation of the learnt rules would lead to a CFIS expressed with words, hence leading to a BCI system which should be more easily interpretable.

In order to perform this linguistic approximation process, we followed Yager's framework [Yag04]. Following this framework, we should first define a vocabulary, i.e., a set of fuzzy sets L_k ($k \in [1..N_t]$). Each fuzzy set L_k represents and describes the word W_k, i.e., the L_k are linguistic terms. Then, we would like to express each "X_i is A_{ji}" by "X_i is W_k". In other words, we would like to replace each fuzzy set automatically learnt by an appropriate word, i.e., we would like to express numbers by words. To do so, we first need to select from the vocabulary the linguistic term L_k that best matches A_{ji}. Once this selection is done, we can replace "X_i is A_{ji}" by "X_i is W_k" as W_k is the word described by L_k. The way we performed these steps of vocabulary definition and linguistic terms selection is described in details in the following section.

4.3.4.1 Defining the vocabulary

The first step required to express the fuzzy rules using words is to define the vocabulary of these words (i.e., the linguistic terms) and their associated fuzzy sets. The CFIS classifier can use two kinds of membership functions for its fuzzy sets: simple Gaussian membership functions or two-sided Gaussian membership functions (see Chapter 3). We chose to use the same kind of membership function for defining the linguistic terms. Our vocabulary is composed of a set of these fuzzy sets, regularly spaced in the [-1:1] interval and with the same standard deviations σ for all membership functions. Naturally, different levels of granularity could be used, by selecting a different number N_t of linguistic terms. Here, we used only odd values for N_t. In order to define a vocabulary with N_t linguistic terms, based on simple Gaussian membership functions, we used the following equations to define the mean μ_k and the standard deviation σ of the k^{th} membership function:

$$\mu_k = -1 + \frac{2k}{(N_t - 1)} \tag{4.1}$$

$$\sigma = \frac{1}{(N_t - 1)\sqrt{2ln(2)}} \tag{4.2}$$

In order to define a vocabulary with N_t linguistic terms, based on two-sided Gaussian membership functions, we used the following equations to define the left mean μ_{Lk}, the right mean μ_{Rk} and the standard deviation σ (the left and right standard deviations are equal):

$$\mu_{Lk} = -1 + \frac{2k}{(N_t - 1)} - \frac{1}{2(N_t - 1)} \tag{4.3}$$

$$\mu_{Rk} = -1 + \frac{2k}{(N_t - 1)} + \frac{1}{2(N_t - 1)} \tag{4.4}$$

$$\sigma = \frac{1}{2(N_t - 1)\sqrt{2ln(2)}} \tag{4.5}$$

Once the fuzzy sets were defined, we associated the corresponding word to each of them. For instance, for $N_t = 3$, we used the words “Low”, “Medium” and “High”, whereas for $N_t = 5$, we used the words “Very low”, “Low”, “Medium”, “High” and “Very high”. Figure 4.2 displays an example of a vocabulary with $N_t = 5$ linguistic terms, based on two-sided Gaussian membership functions.

Once this vocabulary has been defined, the actual linguistic approximation process can be achieved. More particularly, this linguistic approximation consists in selecting a linguistic term in order to replace each fuzzy set used in the fuzzy rules.

4.3.4.2 Selecting the appropriate linguistic terms

Various criteria can be used to select the appropriate fuzzy set from the vocabulary with which a fuzzy set from the CFIS will be replaced. Among the different criteria proposed by Yager [Yag04], we chose to use a single one, namely the “closeness”, which reflects how close two

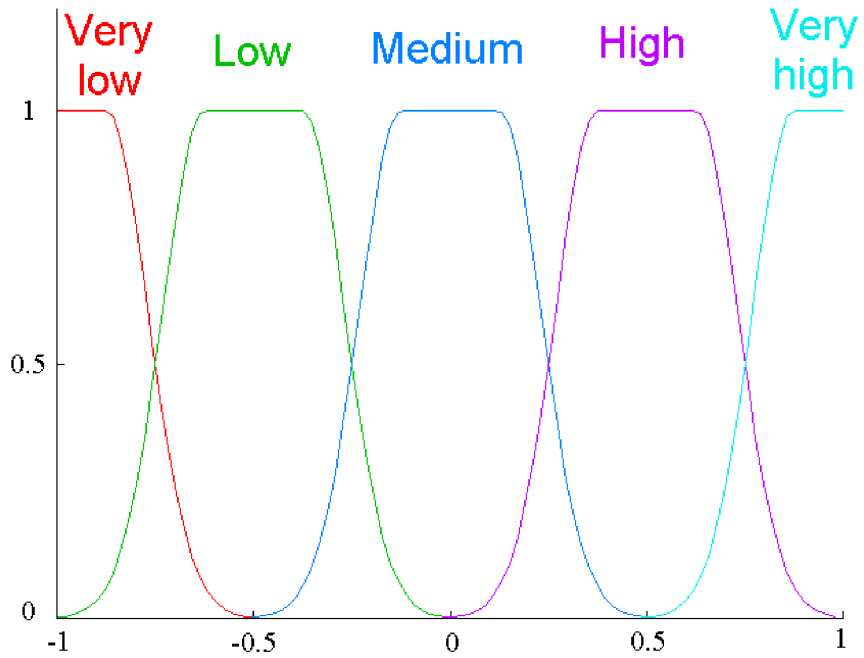


Figure 4.2: A vocabulary of linguistic terms, with 5 terms (Very low, Low, Medium, High, Very high). The fuzzy membership functions used are two-sided Gaussian functions.

fuzzy sets are from each other. In our system, we defined closeness with respect to the distance between two fuzzy sets A and B:

$$closeness(A,B) = \frac{1}{1+dist(A,B)} \tag{4.6}$$

As the vocabulary we used is based on Gaussian membership functions with the same standard deviation, we considered a simple definition for the distance between two fuzzy sets, which does not take into account the standard deviation of their membership function. When using simple Gaussian membership functions, this distance is:

$$dist(A,B) = |\mu_A - \mu_B| \tag{4.7}$$

with μ_A and μ_B being the mean of the two Gaussian membership functions representing the fuzzy sets A and B. When using two-sided Gaussian membership functions this distance is:

$$dist(A,B) = |\frac{(\mu_{LA}+\mu_{RA})}{2} - \frac{(\mu_{LB}+\mu_{RB})}{2}| \tag{4.8}$$

with μ_{LA}, μ_{RA}, μ_{LB} and μ_{RB} being the left and right means of the two-sided Gaussian membership functions representing the fuzzy sets A and B.

Once the closeness is defined, the remaining of the process consists in replacing "X_i is A_{ji}" by "X_i is L_k" where L_k is the fuzzy set from the vocabulary for which $closeness(A_{ji}, L_k)$ is the

highest. Finally, it becomes possible to express "X_i is A_{ji}" by "X_i is W_k" where W_k is the word described by L_k.

Before selecting these linguistic terms, a small normalization process is however necessary. Indeed, the fuzzy sets from the vocabulary are defined on the [-1:1] interval, but the fuzzy sets learnt by the CFIS can be defined on a different interval. Consequently, before computing the closeness between the fuzzy sets, the means (μ, μ_L or μ_R) of the fuzzy sets from all rules should be normalized in [-1:1], independently for each input feature. This means that the term "Low" may represent different values for the first and for the third feature, for instance. In other words, the labels used are relative to a given feature and not absolute.

This step of linguistic approximation is the last step of our method. After this process, what has been learnt automatically by the BCI system can be expressed by a set of "if-then" rules. These rules can report on which activity in relevant brain regions and frequency bands corresponds to which mental state, all of this using only words. As such, it could be expected that this tool would provide some valuable information on the brain dynamics while designing a BCI system. This point is assessed in the next section of this chapter, dedicated to the evaluation of the proposed method.

4.4 Evaluation

In order to evaluate our algorithm, in terms of both performance and interpretability we worked on two different kinds of EEG data. The first set of EEG data was the data set IV of the BCI competition 2003 [BMK+06], studied and described in Section 2.4.1.1. The second set of data gathered brain signals related to visual spatial attention and eye movement preparation [TLK04, TAA08]. The aim of this evaluation was first to assess if the rules automatically extracted with the method were consistent with the physiological literature on these signals. Additionally, we evaluated the performances of the designed BCI in terms of classification accuracy.

This section first presents the two EEG data sets used. Then, it presents the results obtained, i.e., the classification accuracies obtained and the linguistic rules extracted from the data.

4.4.1 EEG data used

4.4.1.1 BCI competition 2003, data set IV

EEG signals contained in this data set correspond to either left hand movement intentions or right hand movement intentions, for one subject [BMK+06]. They have previously been described in more details (see Section 2.4.1.1). The parameters of FuRIA we used for these data are the ones presented in Chapter 2, and as such, the resulting ROI and frequency bands obtained are the ones displayed in Figure 2.4.

4.4.1.2 EEG signals related to Visual Spatial Attention

EEG signals from this data set were recorded by Dr. Areti Tzelepi from the Institute of Communication and Computer Systems in Greece, and Dr. Ricardo Ron Angevin from Malaga University in Spain [TAA08]. The aim of the experiment was to record EEG signals corresponding to visual spatial attention towards the left or the right. Hence, there were 2 classes of signals, denoted as "left" and "right". The data used were recorded on one subject who participated to 3 sessions, each containing 25 trials from each class. During these sessions, the subject was looking at a screen which displayed a virtual environment representing a road. The timing and principle of a trial of these sessions is represented on Figure 4.3. More precisely, at $t = 0$ s (start of a trial), a fixation cross appeared as well as a virtual car rendered in 3D. The car was continuously moving along the road. At $t = 2$ s, a virtual wall appeared, either on the left or on the right side of the virtual environment. The subject was asked to keep fixing the fixation cross at that time. The cross disappeared at $t = 6$ s. At that time, the subject had to make an eye movement towards the wall.

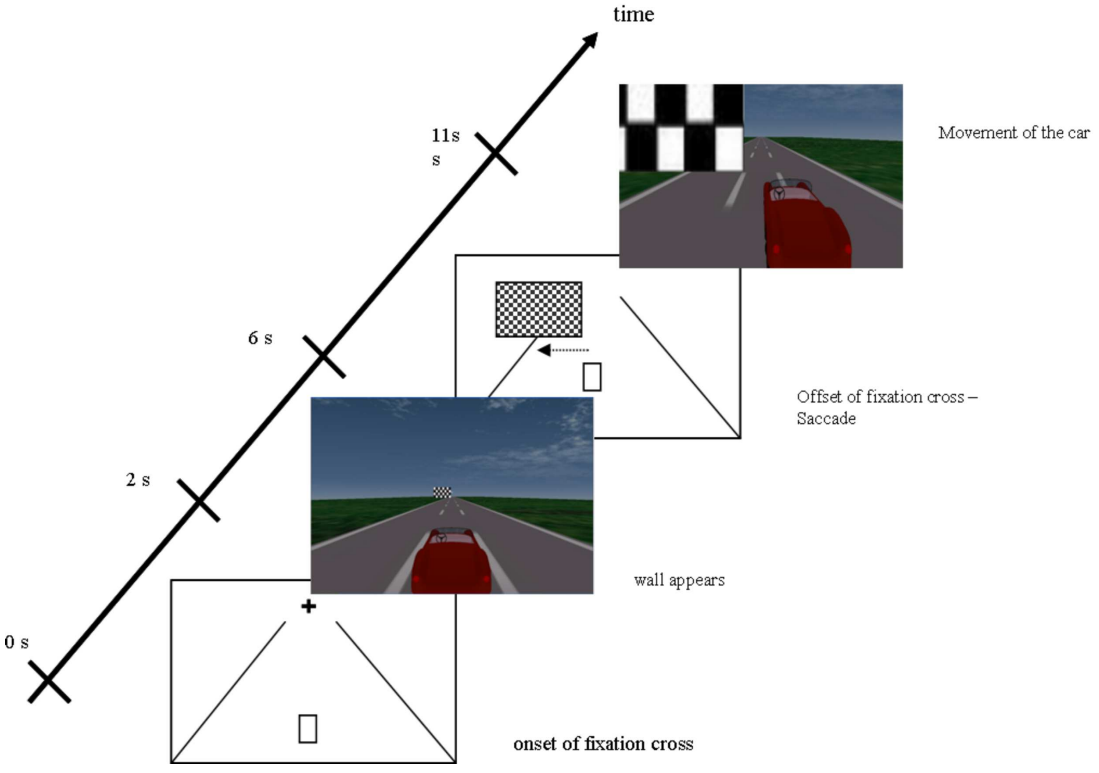


Figure 4.3: Schematic representation of the timing of a trial (picture from [TAA08]).

EEG signals were recorded using 64 electrodes placed according to the extended 10/20 international system. The initial sampling frequency was 512 Hz, but signals were downsampled to 128 Hz before analysis. Signals were also filtered in the 3-30 Hz frequency band before any analysis. More details about the experimental procedure for recording these EEG signals

can be found in [TAA08]. For our analyses, the first two sessions were used as the training set whereas the last session was used as the testing set.

For these data, we used FuRIA based on IIR filters (butterworth filter order 4) and a standard head model composed of 2394 voxels which models the head as three concentric spheres. In this BCI, we considered only spatial fuzzification, and not frequential fuzzification as our results presented in Chapter 2 suggested that spatial fuzzification was the most robust. We investigated for relevant frequency bands in the 8-30 Hz interval. As in previous experiments, we used sLORETA as the inverse solution [PM02], and used a regularization parameter of 100 for this inverse solution. Concerning the CFIS, we used two-sided Gaussian functions as the membership functions.

As with any synchronous BCI, it is necessary to identify the most appropriate time window in which to extract the features and classify them. To do so, we extracted Band Power (BP) features for several time windows of different sizes and positions, and estimated the efficiency of each time window by 10*10 fold cross validation using an LDA classifier. The frequency bands in which to extract the BP features were selected using a statistical analysis-based method similar to the one presented in Section 3.3.2.1. Using this method we analyzed the 4-7 Hz (Delta rhythm), 8-13 Hz (Mu rhythm) and 13-29 Hz (Beta rhythm) frequency bands. This method of time window identification was similar to what has been done by Tzelepi et al for similar data [TAA08]. Indeed, BP features and LDA classifiers are computationally very efficient, and as such, very fast to train and to use. It appeared that the optimal time window was a 0.5 s time window started approximatly at t=2.15 s (see Figure 4.3). This time window corresponds to a few hundreds ms after the wall appearance, which is consistent with the literature [TAA08].

In order to make the system more robust to small time shifts, we extracted several segments in this optimal time window. From each 0.5 s time window we extracted 5 overlapping segments, each segment being 40 samples long and starting 5 samples after the start of the previous one. A feature vector was extracted from each segment and then used for training or testing.

4.4.2 Results

For each data set, we trained the feature extractor and the classifier on the available training set and tested the resulting BCI on the testing set. In order to evaluate the interpretability of our system we investigated the rules that it automatically extracted from data, and confronted them with that knowledge from the literature. These rules, as well as the classification performances of the system are presented below.

4.4.2.1 BCI competition 2003, data set IV

The raw rules (without linguistic approximation) extracted by our system are displayed in Figure 4.4, whereas Figures 4.5 and 4.6 display the rules after linguistic approximation, for $N_t = 3$ and $N_t = 5$ respectively. Let us recall that in these figures, each row represents a fuzzy if-then rule and each column represents a feature (i.e., the activity in the ROI and frequency displayed on top). As such, the functions or words displayed in the tables describe the value of the activity

in a ROI and frequency band for the mental state infered by the corresponding rule.

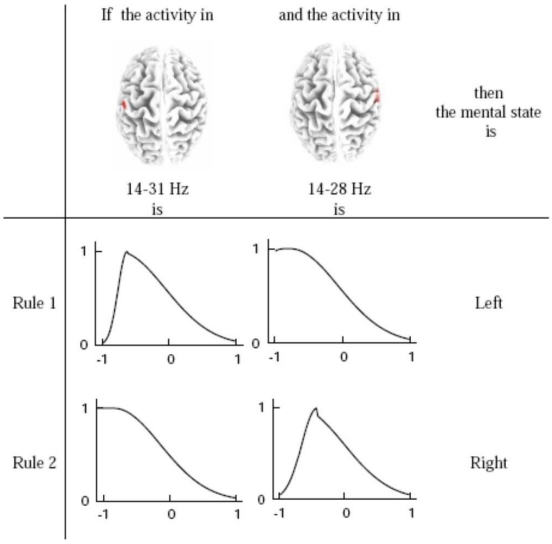


Figure 4.4: Rules extracted automatically by the BCI system on data set IV from the BCI competition 2003, without using any linguistic approximation.

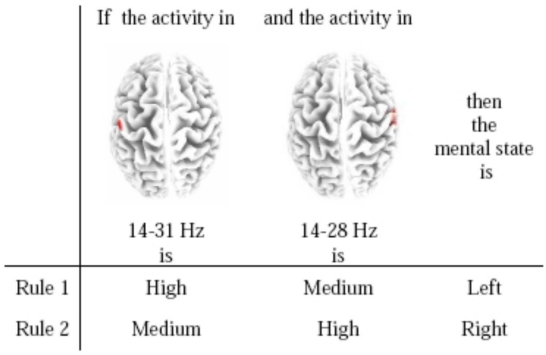


Figure 4.5: Linguistic rules extracted automatically by the BCI system on data set IV from the BCI competition 2003, using a coarse vocabulary (3 terms: Low, Medium and High).

First, it should be noticed that even if a BCI researcher could easily understand the raw rules (Figure 4.4), the linguistic approximations are more comfortable to read. Moreover, these linguistic approximations are more accessible than the raw rules to persons such as clinical employees who do not know what a fuzzy membership function is. Concerning the interpretation of the system, Rule 1 suggests that, during an intention of left hand movement (mental state “left”), the Beta band (here 14-28 Hz or 14-31 Hz, the standard Beta band being 13-30 Hz) activity is lower (label “Medium” or “Low”) in the right motor cortex area than in the left motor cortex area (label “High” or “Very High”). Rule 2 suggests a symmetric behavior, i.e.,

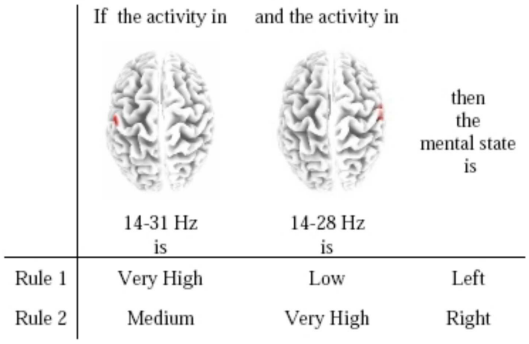


Figure 4.6: Linguistic rules extracted automatically by the BCI system on data set IV from the BCI competition 2003, using a medium vocabulary (5 terms: Very low, Low, Medium, High and Very high).

that, during an intention of right hand movement (mental state "right"), the Beta band activity is lower (label "Medium") in the left motor cortex than in the right motor cortex (label "High" or "Very High"). This behavior is consistent with the literature on movement intention. Indeed, hand movement intention is known to trigger an event related desynchronisation, i.e., a decrease of activity, in the motor cortex contralateral to the hand concerned, in the Beta band [WZL$^+$04, CLL06, PdS99]. The rules automatically learnt by our system actually reflect such a phenomemon.

Concerning the performance of the system, the resulting BCI (using the raw rules) reached an accuracy of 85 % on the test set, i.e., a slightly better score than the one of the winner of the competition on these data, who reached a score of 84 % [WZL$^+$04].

4.4.2.2 EEG signals related to Visual Spatial Attention

Figure 4.7 displays the raw rules extracted from the data, without linguistic approximations, whereas Figures 4.8 and 4.9 display the rules after linguistic approximation, for $N_t = 3$ and $N_t = 5$ respectively.

In terms of performances, this BCI system reached an accuracy of 86 % on the testing set. By comparison, we also used a classical BCI design based on band power features and an LDA classifier (as in [TAA08] on similar data). This design reached an accuracy of 74.8 %. This also suggests that our BCI design can be efficient.

Concerning the interpretability of the system, we presented these rules to Dr. Areti Tzelepi who is a neuroscience expert in visual spatial attention. According to her, there are numerous evidences in the literature that a visual spatial attention task triggers an increase of activity in the occipital, temporal and parietal areas, contralaterally to the side where the attention is drawn. In addition to this contralateral activity, results from the literature suggest that the temporal and parietal areas from the right hemisphere are also activated during spatial attention tasks, independantly from the side of the stimulus (see also [LWV00]). Dr. Areti Tzelepi could find this behavior in the rules we presented her. She could easily exploit these rules thanks to

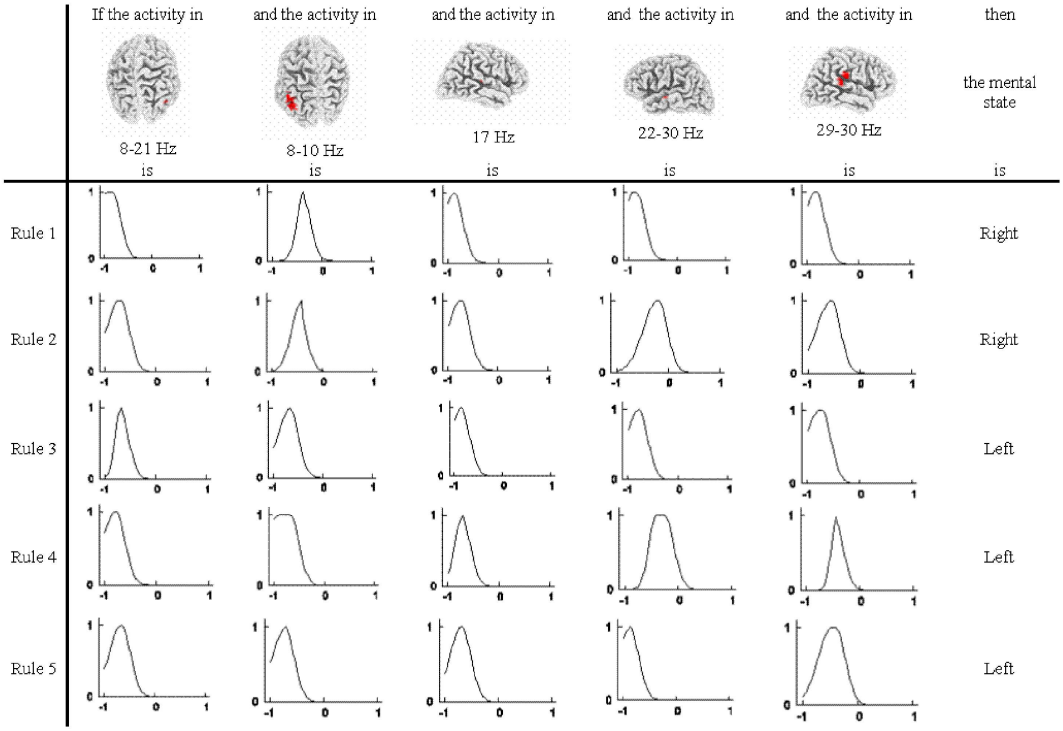
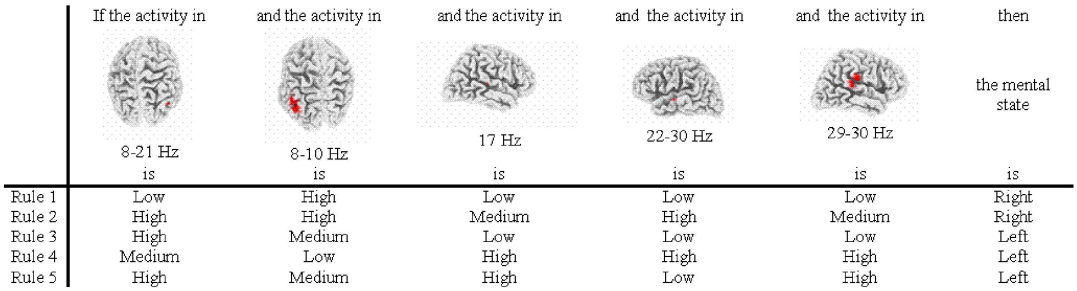


Figure 4.7: Rules extracted automatically by the BCI system on visual spatial attention data without using any linguistic approximation.



	8-21 Hz is	8-10 Hz is	17 Hz is	22-30 Hz is	29-30 Hz is	the mental state is
Rule 1	Low	High	Low	Low	Low	Right
Rule 2	High	High	Medium	High	Medium	Right
Rule 3	High	Medium	Low	Low	Low	Left
Rule 4	Medium	Low	High	High	High	Left
Rule 5	High	Medium	High	Low	High	Left

Figure 4.8: Linguistic rules extracted automatically by the BCI system on visual spatial attention data, using a coarse vocabulary (3 terms: Low, Medium and High).

the linguistic approximation, as she was not familiar with fuzzy inference systems and fuzzy membership functions. Indeed, the contralateral increase of activity in the occipital, temporal and parietal areas was well reflected by the first two features and the different rules, whereas the activation of temporal and parietal areas from the right hemisphere was reflected by the third and fifth features and the rules. However, Dr. Areti Tzelepi could not explain the meaning of the fourth feature and of the fourth rule. Consequently, we removed this feature and this

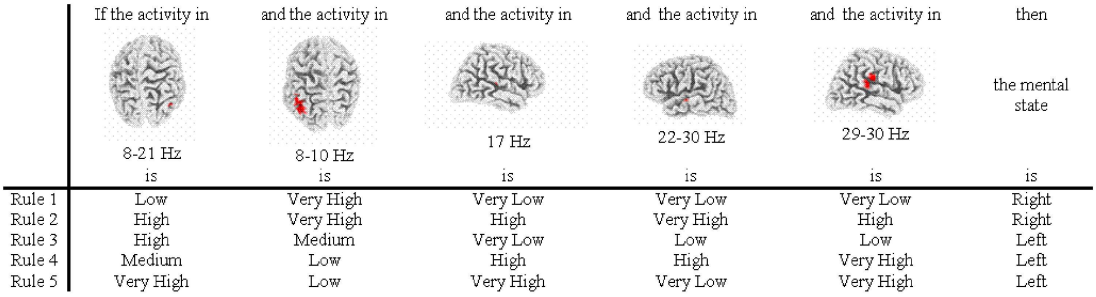


Figure 4.9: Linguistic rules extracted automatically by the BCI system on visual spatial attention data, using a medium vocabulary (5 terms: Very low, Low, Medium, High and Very high).

rule from the fuzzy inference system, and tested it again on the test set in order to assess the contribution of this rule and of this feature (see Figure 4.10, for the resulting set of rules). Interestingly enough, we then obtained an accuracy of 87.2 % whereas we obtained previously 86 % with all rules and features. This seems to confirm that this rule and this feature were not necessary. More precisely, only removing the fourth rule left the accuracy unchanged (86 %), whereas removing only the fourth feature already leaded to an accuracy of 87.2 %. This point stresses that it is interesting to be able to interpret the learnt BCI, in order to check what has been learnt, and, possibly, improve the system, as we have done on these data.

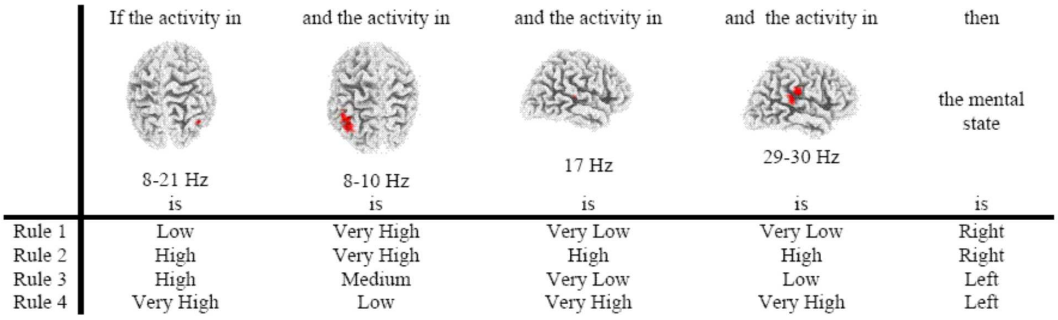


Figure 4.10: Linguistic rules used by the BCI system on visual spatial attention data, after the removal of the fourth rule and the fourth feature. The vocabulary used here is a medium vocabulary (5 terms: Very low, Low, Medium, High and Very high).

4.5 Conclusion

In this chapter, we have presented an algorithm towards the design of a fully interpretable BCI system. This algorithm relies on the combination of an inverse-solution based feature extraction (here, FuRIA), a fuzzy inference system as classifier, and a linguistic approximation. This

system can explain which activity in which brain regions and frequency bands corresponds to which mental state, thanks to if-then rules expressed using simple words rather than mathematical functions. This algorithm aims at providing a tool for easily verifying what has been learnt by the BCI and to compare it with the literature on the brain signals analyzed. The use of linguistic approximations (employing words rather than numbers) aims at proposing a way of displaying the knowledge automatically extracted by the BCI to persons who do not know what a fuzzy membership function is, such as a medical doctor or any other clinical employee.

The evaluation of our algorithm on two kinds of brain signals (hand movement intention and visual spatial attention signals) suggested that knowledge from the literature actually seemed to be reflected by the rules automatically learnt. Moreover, this evaluation suggested that being able to interpret the BCI may help in improving it. Finally, the evaluations also suggested that the proposed BCI was efficient in terms of classification accuracy.

Concerning the limitations of the proposed algorithm, we should admit that this method worked relatively well because FuRIA tends to extract few features. However, it may not always be the case, and with too many features, the whole system of rules might become complex to read and interpret. Further validations would also be required in order to validate the approach on more subjects and more kinds of brain signals.

It would also be interesting to study other ways of representing the linguistic labels. For instance, we could use color coding rather than words, and display a level of activity using a given color for the corresponding brain regions. It could also be interesting to gather, in a single figure of the brain, the level of activity of the different regions and frequencies for a given rule. This would make it possible to quickly understand the whole brain state corresponding to a given mental task. Finally, it could be valuable to find methods to select features independently for each fuzzy rule and/or to merge fuzzy rules, in order to obtain a more compact and simpler set of rules, i.e., a more easily interpretable system.

Chapter 5

Self-Paced BCI Design: a Pattern Rejection Approach

5.1 Introduction

In order to design a convenient BCI with a high information transfer rate, we have already stressed the need to conceive efficient Self-Paced BCI (SPBCI). Let us recall that a SPBCI is a BCI which enables its users to send commands at any time [MKH+06]. Designing a SPBCI requires to continously analyse EEG signals [MKH+06]. Indeed, this analysis should determine if the user is in an Intentional Control (IC) state, i.e., if he is producing one of the brain activity patterns used to control the BCI, or, conversely, if he is in a Non Control (NC) state, i.e., in any mental state except the targeted mental states used to control the BCI [MKH+06]. Finally, if the user is in an IC state, the system must also determine which kind of brain activity pattern is being produced by the user [MKH+06].

In this chapter, we consider the design of a SPBCI as a pattern rejection problem [MA06b], where NC states must be rejected by the BCI, whereas IC states must be accepted and properly classified.

So far, despite the need to design efficient SPBCI, relatively few algorithms have been explored to deal with the NC state. Moreover, to our best knowledge, no study has systematically compared several reject options using several classifiers in the field of BCI. This lack of studies prevents from identifying the desirable properties of reject options and classifiers for SPBCI design.

In this chapter, we first introduce two reject options that have not been used yet in the BCI field: the rejection class strategy, and thresholds on reliability functions based on the automatic multiple thresholds learning algorithm. Second, we assess and compare several reject options using several kinds of classifiers in order to study their behavior and identify the most appropriate ones for SPBCI design.

This chapter is organized as follows: Section 5.2 focuses on a brief state-of-the-art of self-paced BCI design. Section 5.3 describes the algorithms we used in this study, i.e., the classifiers and the pattern rejection techniques as well as the evaluation methodology. Section 5.4 presents the data we used for evaluation (i.e., EEG signals and features) and the results

obtained. Finally, Section 5.5 concludes this chapter.

5.2 Self-paced BCI design

Even though most present EEG-based BCI systems are synchronous, some BCI groups are investing research efforts in the design of EEG-based SPBCI. Pioneer works have focused on the design of 2-state SPBCI, that is, binary BCI which aim at distinguishing between the NC state and a single IC state [BLH93, MB00, YTI03]. As this problem is still far from being resolved, current research works are still addressing this topic [FWB08]. More recently, multi-state SPBCI have been proposed [SSL$^+$07, ZWG$^+$07, dRMFB07]. Such BCI aim at distinguishing the NC state from 2 or more IC states. As such, their design is generally more complex and more advanced than those of 2-state SPBCI. The two following sections briefly review these two kinds of design.

5.2.1 2-state self-paced BCI

Designing a 2-state SPBCI can be seen as the design of a binary BCI whose classes are NC state and IC state. A such, 2-state SPBCI have been designed using binary classifiers, such as LDA, SVM or kNN [FWB08, MB00, BMB06, LSFP07, YTI03] or a detection scheme in which a threshold on a feature value was used to determine the IC or the NC state [YTI03, LFMP$^+$07]. Such features could be a band power feature [LFMP$^+$07] or a matched filter feature [YTI03] for instance. Such SPBCI provide a kind of "brain switch", i.e., a single bit command, which can be issued in a self-paced manner. Even if such BCI are self-paced, providing a single command to the user may appear as relatively limited or not convenient. This is particularly true for applications such as virtual reality applications in which the number of necessary degrees of freedom is generally much higher. Consequently, designing multi-state SPBCI, which use several IC states, appears as essential.

5.2.2 Multi-state self-paced BCI

Relatively recently, 3-state (1 NC state + 2 IC states) [TGP04, BWB07, ZWG$^+$07, BKM$^+$07b] and 4-state (1 NC state + 3 IC states) [MM03, SLS$^+$08, dRMFB07] SPBCI have been proposed. As multi-state SPBCI deal with several IC states, their design appears as more complex than the one of 2-state SPBCI. In the BCI literature, two main strategies are employed to design multi-state SPBCI: the use of Thresholds on Reliability Functions (TRF reject option) or the use of Specialized Classifiers (SC reject option). Algorithms of the first category use one or two thresholds, generally manually defined, on reliability functions [MM03, TGP04, dRMFB07]. If the reliability function, which is generally the classifier output, is higher than the given threshold, then the IC state is chosen. Otherwise, rejection is performed and the NC state is chosen. In the second category, specialized classifiers, known as reject classifiers, are used to distinguish IC from NC states [SLS$^+$08, SSL$^+$07, BWB07, ZWG$^+$07, BKM$^+$07b]. Another classifier, known as the recognition classifier, is then used to distinguish between the targeted patterns. Generally, a different set of features is used for the reject classifier and for the recog-

nition classifier.

As our work focuses on pratical BCI applications such as virtual reality applications, this chapter focuses on multi-state SPBCI, which are more flexible and more convenient for the user. More precisely, the following sections present and evaluate several pattern rejection strategies for the design of a 3-state SPBCI based on motor imagery.

5.3 Method

In this Section, we present the different classifiers and reject options that we investigated, including two reject options that have not been used so far in BCI, to our best knowledge. We also present the evaluation criteria that we used in order to assess the designed SPBCI.

5.3.1 Classifiers

For this study we used four different classifiers, which exhibit different properties with regards to classification performances and rejection. Two of these classifiers are generative classifiers and two are discriminant classifiers (see Section 1.6.1). The generative classifiers describe the training data, which can be interesting to reject the NC state using reliability functions and to generalize using noisy training data. The discriminant classifiers have powerful classification performances. The classifiers we used are a support vector machine, a radial basis function network, a fuzzy inference system and a linear discriminant analysis. Their main properties are briefly described in the following.

5.3.1.1 Support Vector Machine

Support Vector Machines (SVM) are discriminant classifiers, very popular and efficient for BCI design [Bur98, LCL+07]. They use hyperplanes with maximal margins to discriminate features from 2 classes (see Section 1.6.2.2 for more details). In this study, we used a nonlinear SVM based on a Gaussian kernel.

5.3.1.2 Radial Basis Function Network

Radial Basis Function Networks (RBFN) are classical neural networks composed of three layers of neurons: an input layer, a single hidden layer and an output layer [Bis96]. The activation functions used in the neurons of the hidden layer are Radial Basis Functions (RBF) [MA06a, DHS01]. The activation of these RBF is computed using the Mahalanobis distance [DHS01, Bis96]. For the RBFN used in this study, the RBF were learnt using non supervised clustering. The output layer neurons are a linear combination of the activiation value of hidden neurons, learnt with the pseudo inverse method [Bis96]. Due to this architecture, RBFN embed generative knowledge through their hidden layer and discriminant knowledge through their output layer. Such an RBFN has been used successfully for handwritten character recognition [MA06a, MA06b].

5.3.1.3 Fuzzy Inference System

As seen in Chapter 3, Fuzzy Inference Systems (FIS) are a set of fuzzy "if-then" rules that can be learnt from data in order to discriminate classes. FIS are nonlinear and generative classifiers. The FIS used in this study is the one described and studied in Chapter 3: the Chiu's FIS [Chi97b].

5.3.1.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) classifiers are most probably the most used classifiers for BCI design [LCL+07]. LDA classifiers are simple linear and discriminant classifiers that use hyperplanes to separate classes [DHS01]. More details on this classifier can be found in Section 1.6.2.1.

All classifiers have been trained on the same training data set (see section 3.3). The hyperparameters of each classifier have been optimized for each data set separately, by splitting the training data into a training set and a validation set. For LDA and SVM, when multiclass classifiers were needed, several LDA or SVM were combined using the one-versus-the-rest scheme.

5.3.2 Reject options

In pattern rejection theory, two kinds of rejection tasks can be distinguished [MA06b]:

- **Ambiguity rejection:** In this kind of rejection, an input data is rejected if the classifier does not have enough confidence in its decision. This input data could indeed be assigned to two or more target classes. Thus, it may be better to reject this input data rather than misclassifying it.

- **Outlier rejection:** With outlier rejection, an input data is rejected if this data is too different from the classes learnt by the classifier. As such, this input data is more likely to be an outlier, i.e., a data from a class not learnt by the classifier. It is therefore better to reject this data, as it cannot be properly classified.

As the NC state can be any existing mental state except the targeted mental states used to control the BCI, the NC rejection task is clearly an outlier rejection problem. Thus, data from NC states are outliers and data from IC states are target class data.

In this study, we compared three reject options: specialized classifiers, thresholds on reliability functions and rejection class. They are described below.

5.3.2.1 Specialized classifier (SC)

A specialized two-class classifier, known as a reject classifier, is trained independently from the recognition classifier to reject - or not - the input data. If this reject classifier decides to reject the input data, the output class will be NC state. Otherwise, if the reject classifier accepts

the input data, the recognition classifier will classify this input data in order to try to ouput the correct IC state.

Separating the recognition and the rejection classifier allows the rejection classifier to take advantage of another family of classifiers or a different set of features.

5.3.2.2 Thresholds on reliability functions (TRF)

A reliability function is a function Φ in $\Re$ which aims at quantifying the reliability that a classifier has in its decisions [MA06b]. The reliability functions used depends on both the classifier and the rejection task targeted. The TRF reject option uses the knowledge of the recognition classifier through these reliability functions [MA06b]. TRF use the interpretation of reliability functions: the lower is the confidence (i.e., the reliability function value), the more the pattern must be rejected. Thus, the TRF reject option is defined with a set of N thresholds σ_i, each one associated with a reliability function Φ_i. Rejection is performed if all functions are lower than their respective thresholds, i.e., if:

$$\forall i = 1 \ldots N, \Phi_i \leq \sigma_i \tag{5.1}$$

The main problem is to set the threshold values which is increasingly difficult as the number of thresholds increases. Interestingly enough, most SPBCI based on TRF use a single threshold, manually defined.

In this chapter, we introduce in the BCI field the Automatic Multiple-Threshold Learning algorithm (AMTL) developed by Mouchere and Anquetil [MA06a, MA06b]. AMTL is a generic greedy algorithm based on empiric heuristics. It selects the threshold values using a training set D_{ex} of data to be accepted (i.e., examples) and a training set D_{out} of data to be rejected (i.e., outliers). This algorithm has one parameter Θ which is the desired True Acceptance Rate (TAR) on D_{ex}, i.e., the percentage of examples that have been actually accepted. Selecting the thresholds values with the AMTL algorithm is achieved as follows:

1. Compute the value of the reliability functions Φ_i for all examples and all outliers. Then, set the value of the thresholds σ_i so as to reject all examples and all outliers.
2. Repeat the next steps while the evaluated TAR on D_{ex} is lower than Θ.
3. Select the next threshold σ_i to be decreased, according to the function "choose" (see below).
4. Decrease the value of the selected threshold so as to accept one more example (and, as such, probably more outliers)

We used two variants of AMTL with different aims. These two variants are known as AMTL1 and AMTL2 [MA06a, MA06b], and differ on the way they select the threshold σ_i to be decreased, i.e., they differ in their function "choose". The function "choose" of AMTL1 finds the best trade-off between the rejection of the data from the target classes and the rejection of outliers. The function "choose" of AMTL2 finds the best description of target classes without using outliers. The AMTL algorithm has been successfully used for reject purposes

in handwritten character recognition applications [MA06a]. More details on the AMTL algorithm, and its variants AMTL1 and AMTL2 can be found in [MA06a, MA06b].

We can note that TRF include classical approaches which only use the score of the best class to make the reject decision. In the following, we denote as AMTL-MT, TRF using Multiple Thresholds, and as AMTL-ST, TRF using a Single Threshold on the best class score.

5.3.2.3 Rejection class (RC)

The Rejection Class reject option uses an additional class dedicated to the rejection problem [MA06a]. Thus, for an N_c class recognition problem, the RC reject option will use a classifier with $N_c + 1$ classes, the additional class representing all the outliers. As such, outliers are treated as the other target classes with this reject option. To our best knowledge, despite its simplicity, this method has not been considered yet for BCI purposes.

5.3.2.4 Implementation

The SC and RC reject options should take advantage of discriminant classifiers because they consider the rejection problem as a simple classification task. Conversely, the TRF architecture should take advantage of reliability functions representing generative knowledge for the rejection of the NC state [MA06a]. For the SC reject options, we used different features for the rejection and recognition classifiers. However, the classifier family was the same for both classifiers. For the TRF reject option, we used the classifier output scores as reliability functions. More precisely, we used the distances to the separating hyperplane for SVM and LDA, the values of the output neurons for RBFN and the degree of fullfilment of the fuzzy rules for FIS. With AMTL2, for RBFN, we used the activation of radial basis functions as reliability functions in order to have a better target class description.

5.3.3 Evaluation criteria

Assessing the performances of a SPBCI system requires using appropriate evaluation criteria [MKH$^+$06]. In order to evaluate the rejection capabilities of the SPBCI, the most popular measures are the false positive rate and the true positive rate, as well as the area under the receiver operating characteristic curve, which is a closely related measure. Finally, in order to evaluate the classification performances of the SPBCI, we used the accuracy. These measures are described below.

5.3.3.1 Recall, precision, false positive rate and true positive rate

When evaluating the rejection performances of a given system, it is interesting to build what is known as a confusion matrix [MKH$^+$06]. Such a matrix contains the number of data from a given class label that have been assigned to another given class label. As such, this matrix contains the number of True Positive (TP, acceptance of an IC state), of True Negative (TN, rejection on an NC state), of False Positive (FP, acceptance of an NC state) and of False Negative

(FN, rejection of an IC state). For a SPBCI problem, this confusion matrix will be as in Table 5.1

Table 5.1: A confusion matrix for a given SPBCI.

		Real labels	
		IC state	**NC state**
Estimated labels	**IC state**	TP	FP
	NC state	FN	TN

From this matrix, we can derive two interesting pairs of performance measures: Recall/Precision and False Positive Rate (FAR)/True Positive Rate (TAR). These measures are defined as follows:

$$TAR = Recall = \frac{TP}{TP+FN} \tag{5.2}$$

$$FAR = \frac{FP}{FP+TN} \tag{5.3}$$

$$Precision = \frac{TP}{TP+FP} \tag{5.4}$$

FAR and TAR represent the rejection performances of the evaluated system as they are independent from the proportion between IC and NC states. Precision is linked to the comfort of the final user, as it summarizes how often the BCI system will respond correctly. Precision depends on the proportion between IC and NC states. We used all these measures in our evaluations.

5.3.3.2 Receiver Operating Characteristic (ROC) Analysis

An interesting tool related to TAR and FAR is the ROC analysis [Faw06]. For a given rejection system and a given data set, we can obtain a couple (FAR, TAR) which can be displayed as a 2D point. Performing a ROC analysis consists in repeatidely changing the parameters of the rejection system (e.g., by changing the threshold values with TRF, or by changing the proportions between outliers and examples with RC and SC), in order to obtain a set of points (FAR, TAR). By arranging these points together, by increasing FAR, and by linking them, one can obtain a ROC curve (see Figure 5.1 for examples). Such a curve summarizes the rejection capabilities of the system.

The Area Under the Curve (AUC) is of particular interest. The higher the AUC for a given classifier, the better the rejection capabilities of this classifier. As an example, on Figure 5.1, the classifier A has a higher AUC than classifier B, and as such, it is better than classifier B. The AUC is a number between 0.0 and 1.0. An AUC of 0.5 corresponds to a random classifier which cannot distinguish outliers from examples (see Figure 5.1). In our evaluations, we also considered the AUC as an evaluation measure. However, we computed the AUC for FAR lower

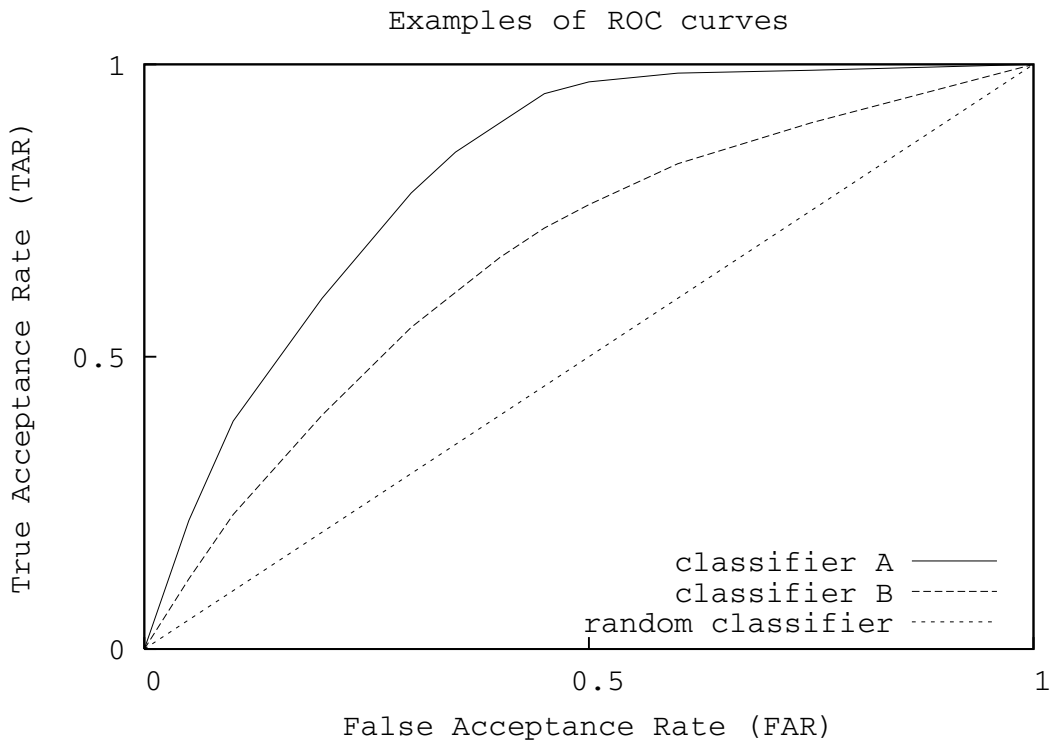


Figure 5.1: Example of ROC curves for 3 different classifiers.

than or equal to 0.2. We chose indeed to use the AUC for FAR ≤ 0.2 rather than the total AUC, because Mason *et al* highlighted that only the beginning of the ROC curves was relevant for BCI [MKH$^+$06]. Actually, a high FAR tends to cause high user frustration which makes the corresponding BCI not convenient to use. It should be noted that the AUC for FAR ≤ 0.2 would be 0.02 for a randomly performing classifier.

5.3.3.3 Accuracy

In addition to the rejection performances of the system, it is also interesting to evaluate the classification capabilities of the designed SPBCI. To do so we considered the accuracy of classification [MKH$^+$06] for a fixed FAR. The accuracy is defined as the percentage of accepted IC states that have been correctly classified.

5.4 Evaluation

This section deals with the evaluation of the previously mentioned classification and rejection algorithms on motor imagery EEG data sets. This section first describes the EEG data used and the features extracted from these data. Then, it presents the evaluation measures obtained on these data.

5.4.1 Motor imagery EEG data used

Evaluations were achieved on 4 EEG data sets of Motor Imagery (MI) acquired from 2 healthy subjects (males, 24 and 23 year old). Subjects had very few or no previous experience in BCI. During the experiments, subjects were asked to perform MI and more precisely imagination of left or right hand movements [PN01].

For each subject, data were collected over 2 days during which 3 to 5 sessions were recorded each day. A session was composed of 20 trials of each of the two classes (LEFT or RIGHT), arranged in a random order. The timing of the sessions was organized according to the Graz BCI protocol [PN01]. In this protocol, a trial lasted 8 seconds, during which the subject received instructions the first 3 seconds and had to perform the required MI task during the last 5 seconds (see Figure 5.2). Trials were seperated by periods of random length. We specifically asked subjects not to perform MI nor real movements outside the 5 second periods dedicated to MI. From second 4.125 to second 8, subjects were provided with a continuous feedback under the form of a bar with a changing length. This bar indicated to the subject what was the mental state recognized by the BCI.

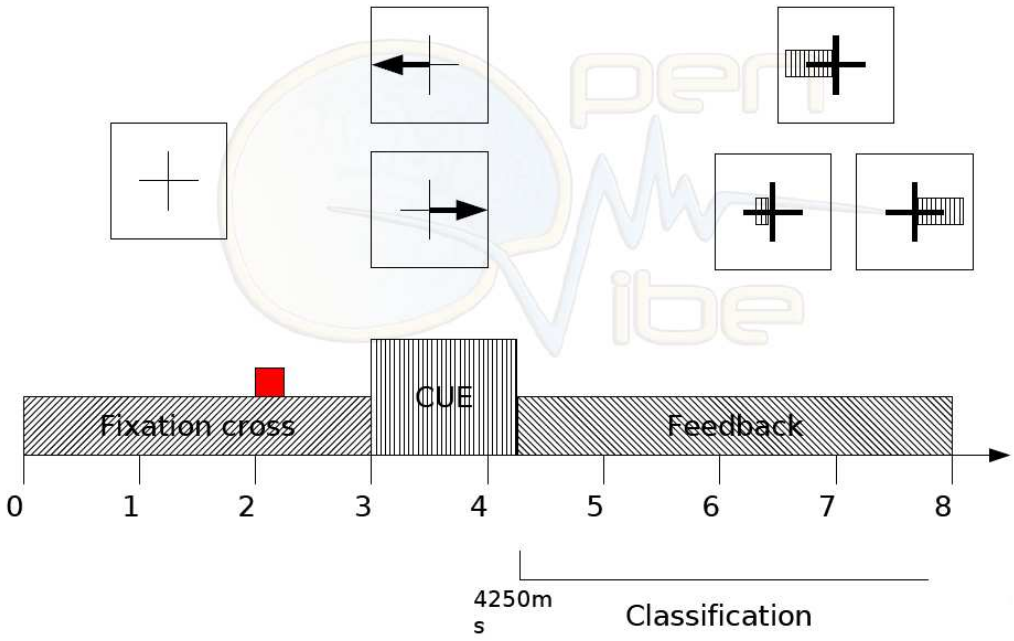


Figure 5.2: Timing of a trial. Trials are separated by rest periods of random length.

EEG signals were sampled at 512 Hz, and were recorded using electrodes FC3, FC4, C5, C3, C1, C2, C4, C6, CP3, CP4, with the reference electrode placed on the nose. The EEG data acquisition machine used was a Nexus 32b from the Mind Media company. These electrodes cover the motor cortex area, and correspond to standard electrode positions, placed according to the international 10-10 system [AES91] (see Figure 5.3). For each subject and each day, the first half of the sessions was used to build a training set whereas the remaining sessions were

used to build a test set. Hence, we used a total of 4 data sets, each one being composed of a training set and a test set. EEG signals from the training sets were visually inspected and periods of MI polluted by artifacts were removed. No artifact was removed from the test sets.

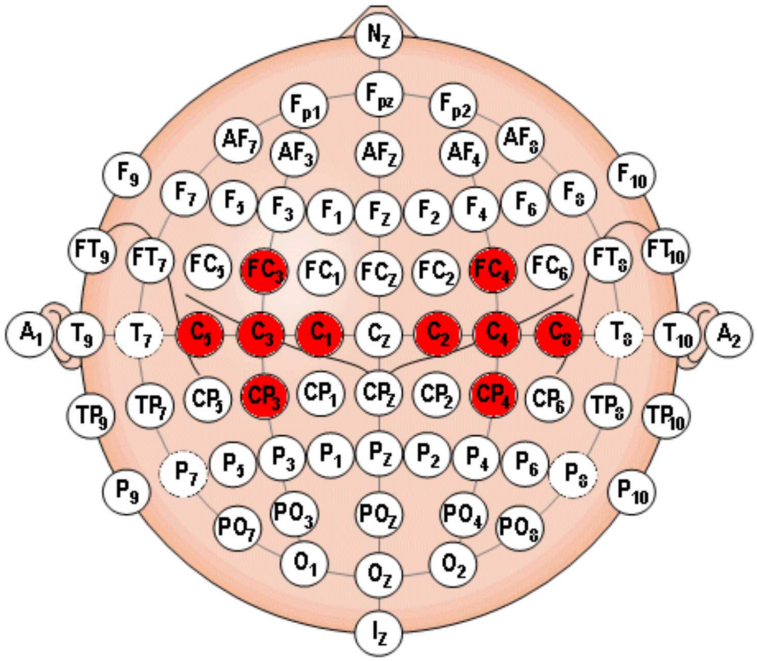


Figure 5.3: Placement of electrodes in the 10-10 international system. The electrodes we used in the experiment are displayed in red.

5.4.2 Data labelling

We labelled as belonging to the LEFT or RIGHT class the samples that were in the MI period of each trial, according to the imagined movement the subject was asked to perform. Samples from the first 0.5 s of each MI period were labelled as NC in order to take into account the user's reaction time. All other samples were also labelled as belonging to the NC state. Then, EEG signals were segmented into 1 s segments with 93.75 % (15/16) of overlap between consecutive segments. Each segment was labelled according to the most represented label among the samples composing it. Then, a feature vector was extracted from each segment and labelled with this segment label. As such, 16 feature vectors were extracted for each second.

5.4.3 Preprocessing

As the preprocessing step, we applied temporal and spatial filters to EEG signals. More specifically, EEG signals were band-pass filtered in 3-45 Hz, using a butterworth filter of order 4, in order to attenuate slow variations of EEG as well as 50 Hz power line interference. Then, from the 10 initial EEG channels, 2 new channels were designed by applying a discrete surface

Laplacian spatial filter over channel C3 and C4 [WBM+02, MMDW97] (see Section 1.4.1.2). More precisely, these 2 new channels $C3'$ and $C4'$ were obtained as follows (see [MMDW97] for details):

$$C3' = 4C3 - FC3 - C5 - C1 - CP3 \tag{5.5}$$

$$C4' = 4C4 - FC4 - C6 - C2 - CP4 \tag{5.6}$$

Features were extracted from these two virtual channels $C3'$ and $C4'$.

5.4.4 Feature extraction

In order to build the classifier inputs, we extracted logarithmic BP features from the two Laplacian channels [PN01] (see Section 1.5.2.1). As already mentioned, BP features are popular and efficient features for MI classification [PN01].

Two sets of BP features were generated: features for rejection and features for recognition. The features from the first set aimed at discriminating the IC state from NC states, whereas the features from the second set aimed at discriminating the two IC states i.e., left or right MI. The first set was obtained by extracting BP features in the frequency bands that best differentiated IC from NC, whereas the second set was obtained using frequency bands that best differentiated left MI from right MI. For each subject, these frequency bands were identified using a statistical analysis-based method similar to what has been presented in Section 3.3.2.1. The only differences are that 1) we used here a multi-comparison randomization paired t-test instead of a simple paired t-test, i.e., we used a more powerful test, and that 2) we did not perform any "shrinking" step as we realized this step did not always increase the performances and sometimes even decreased them. The statistical analysis compared the BP mean values for the two corresponding conditions (NC versus IC or left MI versus right MI) for different frequencies in the 4-35 Hz frequency band, with the aim of selecting the most discriminative frequency bands. We performed this analysis for the 4-7 Hz (Delta rhythm), 13-15 Hz, 15-18 Hz, 18-30 Hz (low, middle and high Beta rhythm) and 30-35 Hz (Gamma rhythm) frequency bands. Features for rejection were used as input of the reject classifiers whereas features for recognition were used for the recognition classifiers.

5.4.5 Results and discussion

This section presents the results obtained for all reject options and classifiers. Table 5.2 displays the Area Under the ROC Curve (AUC), for FAR lower than or equal to 0.2, obtained by all methods. Table 5.3 displays the accuracy, precision and TAR obtained by each classifier and reject option, averaged over the four data sets, for a fixed FAR of 10%. This FAR is similar to the FAR used in the work of Scherer *et al* [SLS+08]. For the complete details of accuracy, precision and TAR for each data set, please refer to Annex D.

Results showed that using a nonlinear classifier within the RC reject option led to the most efficient SPBCI. Independently from the reject option used, nonlinear classifiers, i.e., FIS, RBFN or SVM, provided the best rejection results. Using TRF, LDA provided the highest

Table 5.2: Rejection capabilities: area under the ROC curves for FAR ≤ 0.2, for all data sets and methods. The best result for each data set is displayed in bold characters.

reject option	Classifier	Subject 1		Subject 2	
		Day1	Day 2	Day 1	Day 2
SC	SVM	**0.105**	**0.077**	**0.057**	0.046
	FIS	0.102	0.075	0.052	0.039
	RBFN	0.103	0.074	0.055	0.044
	LDA	0.102	0.071	0.041	0.035
RC	SVM	0.102	**0.077**	0.056	**0.062**
	FIS	0.102	0.072	0.055	0.052
	RBFN	0.095	0.075	0.054	0.058
	LDA	0.095	0.072	0.053	0.048
TRF AMTL1 ST	SVM	0.025	0.040	0.028	0.033
	FIS	0.057	0.039	0.04	0.036
	RBFN	0.053	0.043	0.033	0.026
	LDA	0.02	0.036	0.047	0.036
TRF AMTL1 MT	SVM	0.025	0.041	0.028	0.032
	FIS	0.082	0.06	0.037	0.042
	RBFN	0.066	0.047	0.030	0.028
	LDA	0.038	0.039	0.038	0.037
TRF AMTL2 MT	SVM	0.025	0.040	0.028	0.032
	FIS	0.058	0.044	0.041	0.042
	RBFN	0.065	0.050	0.030	0.028
	LDA	0.021	0.027	0.039	0.035

accuracy, but this has to be moderated by the low TAR it provided. Actually, it is very likely that LDA was in fact performing ambiguity rejection [MA06b] and not outlier rejection, which could explain the results.

Concerning the reject options, the obtained AUC and TAR may appear as modest, but it should be noted that they are in line with results found in the literature. For instance, the 3-class SPBCI presented in the work of Scherer *et al* obtained an averaged FAR of 16.9 % and an average TAR of 28.4 % [SLS$^+$08].

The most efficient methods in terms of rejection capabilities are RC and SC. However, RC outperformed SC in terms of accuracy for a fixed FAR of 10 %. TRF had the lowest rejection capabilities, even if with a low resource cost the use of multiple thresholds improved the results as compared to a single threshold, especially for generative classifiers. Indeed, regarding the AUC in Table 5.2, it can be noticed that discriminant classifiers, i.e., SVM and LDA, obtained scores that are close to random classification scores with TRF. However, it is interesting to note that TRF provided the highest accuracy. This suggests that, implicitely, TRF also performed ambiguity rejection in addition to outlier rejection. It is also interesting to note that when

Table 5.3: Classification capabilities: average Accuracy (Acc), TAR and Precison (Prec), in percent, for a fixed FAR of 10%. The best results are displayed in bold characters.

		SVM	FIS	RBFN	LDA
SC	Acc	74.1±8	73.2±5.2	73,9±9	72.0±4.7
	TAR	38.2±15.2	34.3±16.6	37.0±15	33.1±17.7
	Prec	69.1±8.3	65.7±11.6	68.3±8.8	64.6±12.1
RC	Acc	83.4±7.7	79.4±7.3	80.2±8.3	81.1±7.3
	TAR	**40.0±12.2**	38.7±15.2	38.2±10.5	36.1±12.3
	Prec	**70.8±6.2**	69.5±7.8	70.0±5.8	68.4±7.4
AMTL1 ST	Acc	84.1±5.7	92.6±7.1	82.7±9.1	**94.5±5**
	TAR	16.3±3.6	22.8±4.9	20.1±6.4	17.2±8.8
	Prec	50.5±5.5	58.7±5	55.0±8.3	48.9±16.5
AMTL1 MT	Acc	84.1±5.8	77.6±8.1	83.5±8.1	93.2±4.8
	TAR	16.2±3.4	28.5±11.1	22.2±6.5	19.3±2.2
	Prec	50.4±5.3	62.8±8.8	57.6±7.4	55.0±3.2
AMTL2 MT	Acc	83.8±5.8	92.1±6.4	75.9±6.6	94.1±3.9
	TAR	16.2±3.6	24.1±4.7	22.5±9.8	13.6±6.3
	Prec	50.4±5.5	60.1±4.7	56.8±11.1	44.3±12.9

considering only TRF, FIS reached the best rejection performances. As such, this confirms results obtained in Section 3.7, where FIS showed very good outlier rejection capabilities when using TRF, as compared to other classifiers. However, it seems that, overall, TRF may not be the most appropriate method for SPBCI design.

The fact that non-linear classifiers perform the best on this problem can be stressed. Indeed, linear classifiers, and especially LDA, are widely used in the BCI community, and they are considered as the most efficient for a number of BCI problems [MAB03, BMK+06]. Currently, most BCI are synchronous, and linear classifiers actually seem to be appropriate for this problem [LCL+07]. However, when it comes to SPBCI design, things seem to be different, and non-linear classifiers seem to be the most efficient. Indeed, while linear methods could be appropriate for discriminating two or three classes, they are very unlikely to be able to discriminate the IC states from all the possible mental states that compose the NC state.

5.5 Conclusion

This chapter proposed to consider self-paced BCI design as a pattern rejection problem. As such, it aimed at introducing new rejection techniques in the BCI field as well as identifying the most appropriate ones for self-paced BCI design. More precisely, this chapter first introduced two pattern rejection strategies for self-paced BCI design: the Reject Class (RC) and the Thresholds on Reliability Functions (TRF) based on the Automatic Multiple Thresholds Learning (AMTL) algorithm. Then, it compared the Specialized Classifiers (SC), RC and TRF

reject options using a Gaussian Support Vector Machine (SVM), a Fuzzy Inference System (FIS), a Radial Basis Function Networks (RBFN) and a Linear Discriminant Analysis (LDA) classifiers, on left and right hand motor imagery data.

Our results first showed that using non linear classifiers such as Gaussian SVM, FIS and RBFN reached the best rejection performances. Our results also showed that using multiple thresholds with TRF led to better performances than when using a single threshold, as done in most current self-paced BCI based on TRF [dRMFB07]. Finally, they suggest that generative classifiers give the best performances when using TRF. Concerning the reject options, RC outperformed SC which outperformed TRF. To conclude, we could thus recommend using the RC reject option with nonlinear classifiers for efficient self-paced BCI design. Future works on this topic could consist in combining efficiently these different reject options. It would be also interesting to study confusion rejection techniques in order to reduce the number of erroneous classifications in BCI, and as such possibly increase the information transfer rate of the system.

Part 2:
Virtual reality applications based on BCI technology

Chapter 6

Brain-Computer Interaction with Entertaining Virtual Worlds: A Quantitative and Qualitative Study "out of the lab"

6.1 Introduction

Currently, a lot of Brain-Computer Interfaces studies are conducted inside laboratories, in highly controlled conditions, and with relatively few subjects trained over a number of sessions which may be large. A notable exception is the work of Guger *et al* which evaluated a BCI with 99 naive subjects during an exposition [GEH+03]. This work focused on the performances of subjects who had to control a synchronous, 2-class BCI based on 2 bipolar EEG channels and a trained classifier. Subjects were asked to imagine movements of their right-hand or their feet and were provided with a simple 2D visual feedback. Their results showed that 93% of the subjects were able to reach an accuracy equal or greater than 60%. Besides, most current studies are focused on the BCI performances and not on the subjects' preferences.

In the work presented in this chapter, we studied both the performances and the preferences of 21 naive subjects during an exhibition. These subjects used a self-paced BCI, based on a single EEG channel, which does not use machine learning of the mental state to be detected. The subjects could interact with an entertaining virtual reality application inspired from the "Star wars^{TM}" movie.

This Chapter is organized as follows: Section 6.2 describes the method employed for this study. It presents the BCI system used (Section 6.2.1), the entertaining VR application employed (Section 6.2.4) and the experiments conducted (Section 6.2.6). Then, Section 6.3 presents the results obtained, i.e., the subjects' performances (Section 6.3.1) and preferences (Section 6.3.2). A concluding section ends this chapter.

6.2 Method

During our experiments, subjects had to interact with a virtual reality application using a BCI. This section describes the BCI we used, the virtual reality application and the organization of the experiments.

6.2.1 The BCI system

We have designed a simple self-paced BCI system based on real or imagined foot movements. This BCI is based on a single EEG channel, located at position Cz and aims at detecting a Beta Event Related Synchronisation (ERS), appearing posterior to the real or imagined foot movement [Pfu99].

6.2.2 Preprocessing and feature extraction

In order to detect the post-movement Beta ERS, the EEG signal was first band-pass filtered in the 3-45 Hz band. Then, a single Band Power (BP) feature was extracted in the Beta band (16-24 Hz) for the last second of data. This feature was extracted every 100 ms and the last four consecutive features were averaged (with a moving average) in order to produce a smooth Control Signal (CS).

6.2.3 Classification

To detect the Beta ERS, and hence, the foot movement, we used a simple threshold Th. If the computed CS was higher than this threshold Th, a foot movement was detected (intentional control state) and a command was sent to the application. If the CS was lower than the threshold Th, the non-control state was detected and no command was sent to the application. This design enables the user to control the BCI in a self-paced way. The value of Th was simply determined according to the mean μ and standard deviation σ of a CS epoch obtained while the subject was relaxed, according to Equation 1.

$$Th = \mu + 3\sigma \tag{6.1}$$

This threshold determination procedure is similar to the one used in another virtual reality application based on BCI [LFMP+07]. It should be noted that Th is determined without using any example of real or imagined foot movement. A such, this BCI does not learn the mental state to be detected.

6.2.4 The Virtual Reality application: “Use the force!”

We have developed an entertaining Virtual Reality (VR) application, in order to provide the subjects with an engaging and motivating experiment. Our virtual environment corresponds to the inside of a “Star Wars^{TM}” mother ship, in which the subject could see a virtual spaceship (a Tie-Fighter) and a static character (Darth Vader) (see Figure 6.1). The purpose of the application was to lift the Tie-Fighter up by using the BCI. This task established an analogy

between the use of the BCI and the use of "the force" in the Star Wars^{TM} movie. As such, the application was named "Use the force!". More precisely, the Tie-Fighter was lifted-up when the VR application received the corresponding command from the BCI. The Tie-Fighter was lifted-up at a speed proportional to the value of the CS. When no command was received, the Tie-Fighter went down.

Figure 6.1: View of the virtual environment of the "Use the force!" application.

6.2.5 Implementation

The VR application was developed with the OpenMASK VR platform [MAC^{+}02, LCAA08] and the BCI was developed with the OpenViBE BCI platform [RGC^{+}07, Ope06b]. This BCI platform enables a fast and flexible design of BCI scenarii by assembling several modules. The VR application and the BCI system were easily connected using the VRPN protocol, thanks to dedicated modules of OpenViBE.

6.2.6 The experiment

Subjects participated in an experiment with a duration of approximately 45 minutes. This experiment was divided into seven successive steps: electrode montage, signal visualization, baseline, free interaction, real movement game, imagined movement game, questionnaire. These steps are described in the following sections.

6.2.6.1 Electrode montage

The first step of the experiment consisted in fixing the electrodes on the subject's head. For this experiment, only three electrodes were used: a ground electrode (located on the forehead),

a reference electrode (located on the nose) and the Cz electrode, which is located over the foot motor representation area of the brain (see Figure 1.2). Electrode Cz was fixed using an adhesive paste instead of a cap, for a faster setup. The EEG data acquisition machine used was a Nexus 32b from the Mind Media company. A view of the setup of the whole experiment is displayed on Figure 6.2.

Figure 6.2: Experimental setup.

6.2.6.2 Signal visualization

During the second step, subjects were shown their own EEG signal recorded at Cz (band-pass filtered in 3-45 Hz) while they were clenching their teeth or blinking. This aimed at showing them the need to be as relaxed as possible during the experiment and the need to avoid blinking. During the next steps, subjects were regularly reminded to stay as relaxed as possible.

6.2.6.3 Baseline

During this step, subjects were only asked to stay relaxed. Once they were relaxed, 20 seconds of EEG signal were recorded and converted into a CS which was used to compute Th using Equation 6.1.

6.2.6.4 Free interaction

During this step, subjects could interact freely with the VR application by using real foot movements. When the BCI detected a Beta ERS, the Tie-Fighter was lifted-up. Alternatively, the CS was shown to the subjects so that they could see the impacts of real foot movements on the

Beta power. This step aimed at making subjects familiar with the application and with the task. If a subject seemed unable to lift the spaceship, the baseline step was performed again, in order to obtain a new Threshold Th. Then, the next steps followed.

6.2.6.5 Real movement game

Subjects were invited to participate in a video-game-like experiment. During this game, subjects had to lift the Tie-Fighter up, by performing real foot movements during specific periods instructed by the application. These instructions were used to evaluate the system but were not used by the BCI for classifying the input data. Actually, the user could lift the Tie-Fighter up at any time and all the game long, independently from the instructions. In other words, we used a "paced test environment" to evaluate this self-paced BCI [MKH+06].

The game was composed of 10 trials. Each trial lasted 10 s, and was divided into 3 phases (see Figure 6.3): 1) A resting phase lasting 4 seconds during which no specific task was given to the subject. 2) A “move” phase, lasting 3 seconds, during which the subject was instructed to perform real foot movements. The instruction was given using a green text “move” appearing on the screen. 3) A “stop” phase lasting 3 seconds, during which the subject was instructed to stop performing the movement in order to lift the Tie-Fighter up. The instruction was given using a red text “stop moving” appearing on the screen. If the subject managed to lift the Tie-Fighter up during this third phase, his score was increased and displayed using a yellow gauge located on the left corner down the screen.

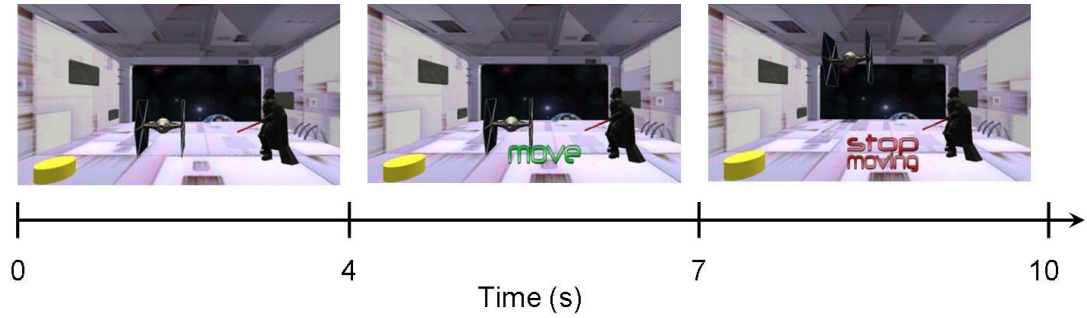


Figure 6.3: Temporal structure of a trial of the VR game.

6.2.6.6 Imagined movement game

This game was identical to the previous one except that subjects were instructed to perform imagined foot movements instead of real foot movements. We instructed subjects to perform kinaesthetic motor imagery rather than visual motor imagery [NSRP05]. Indeed, it has been shown that kinaesthetic motor imagery triggered higher ERS/ERD than visual motor imagery, over the sensori-motor areas. As such, it leads to better classification performances [NSRP05].

6.2.6.7 Questionnaire

After the experiment, subjects were asked to fill in a subjective questionnaire. This questionnaire gathered various questions related to their feelings during the BCI experiment. They were also encouraged to give free comments and remarks regarding the whole BCI system and the VR application. An excerpt of the questionnaire can be found in Annex E.

6.3 Results

The experiments took place during the Laval Virtual 2008 VR exhibition, on a booth. As such, the environment was a noisy environment with people moving and talking around. 21 naive subjects (mean age: 33.48 ± 9.14), 18 males and 3 females, participated voluntarily to the experiment. No selection was performed and all volunteers were accepted. All subjects gave their written informed consent before the experiment. Detailed information about subjects can be found in Annex F. This section describes the subjects' performances as well as the qualitative and quantitative data extracted from the questionnaires filled by the subjects.

6.3.1 Subjects' performances

We assessed the subjects' performances by computing the number of True Positives (TP) and False Positives (FP) they obtained during the games [MKH+06]. We counted a single TP when the CS value became higher than the threshold Th once or more times during the "stop moving" phase (see section 6.2.6). We counted a single FP when the CS value became higher than the threshold Th once or more times during the "move" phase, during which a Beta Event Related Desynchronisation (ERD) should be observed and not a Beta ERS. What happened during the resting phase was not taken into account in the performance analysis. From the FP and TP, we computed the Hit-False (HF) difference, which corresponds to the number of TP minus the number of FP [MKH+06]. Performances obtained by subjects are summarized in Figures 6.4, 6.5 and 6.6, under the form of absolute frequency diagrams for TP and HF difference. They show the number of subjects who obtained a given performance, for real and imagined movement games separately. The detailed performance results are gathered in Annex F.

These diagrams show that about half of the subjects (12 subjects out of 21) reached an HF difference ≥ 3 using real movements, and that about a quarter (5 subjects out of 21) reached an HF difference ≥ 3 using imagined movements. According to simulations performed as described in [MPP08], a system which reach an HF difference ≥ 3 with 10 trials per class, is better than a randomly performing system (one-tailed test) with a probability of type I error ≤ 0.054. This suggests that roughly half of the subjects had at least a small control over the Tie-Fighter using real movements and that a quarter had at least a small control using imagined movements. The mean HF difference was 3.14 ± 2.24 for real movements and 1.33 ± 2.03 for imagined movements while the mean TP was 4.95 ± 2.18 for real movements and 2.67 ± 2.08 for imagined movements. These results may appear as modest but one should consider the fact that subjects were naive and untrained and that a very simple BCI design was used. Actually, we used a single EEG channel, placed at a standard location (i.e., a non-optimized location) and we used a single feature, based on a standard frequency band (i.e., a

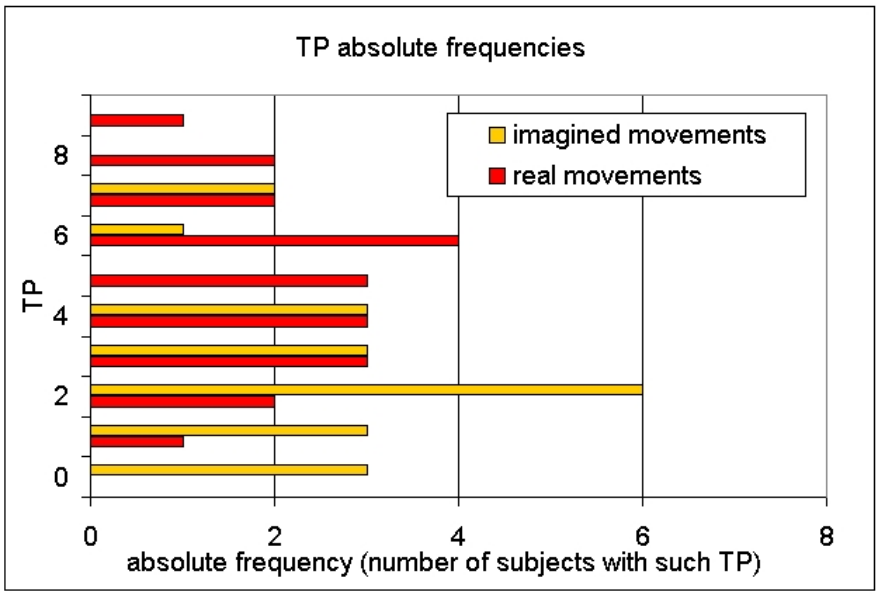


Figure 6.4: Absolute frequency diagrams for True Positives (TP), for real or imagined movements.

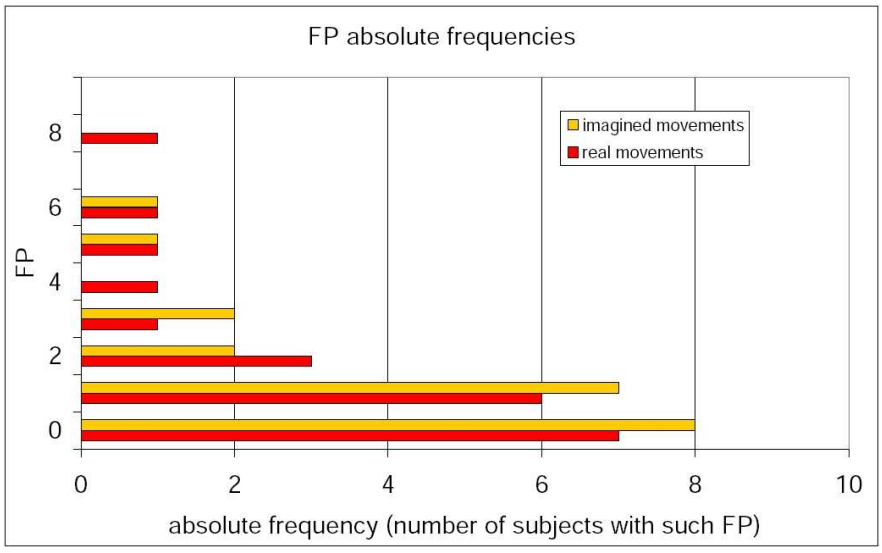


Figure 6.5: Absolute frequency diagrams for False Positives (FP), for real or imagined movements.

non-optimized frequency band), with a simple threshold. This enhances the need to use feature extraction and classification algorithms that can learn subject specific information (such as FuRIA or FIS) in order to design a more optimal and efficient BCI.

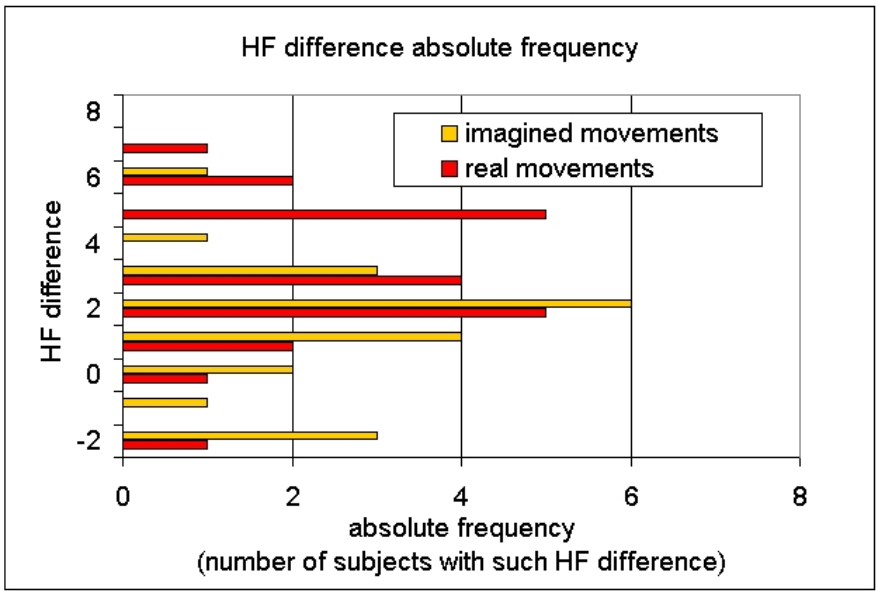


Figure 6.6: Absolute frequency diagrams for HF difference, for real or imagined movements.

6.3.2 Subjective questionnaires

The structured questionnaire filled by the subjects enabled us to extract two kinds of data (see Annex E for more details):

- **quantitative data** as subjects were asked to grade questions related to their feelings.
- **qualitative data** as subjects were also asked to answer several open questions.

The data gathered and their analysis are described in the following sections. The raw data can be found in Annex F.

6.3.2.1 Quantitative data

Subjects were asked to grade questions by giving a mark between 1 and 7. Table 6.1 displays the average marks given by the subjects for the two conditions (real movement game and imagined movement game) according to various criteria.

Our results first showed that the experiments did not seem *tiring* for the subjects. The experiments with imagined movements seemed however more *tiring* than that with real movements. However, this difference is not statistically significant (Wilcoxon test $W = -28$, $p > 0.1$). Despite the use of paste and gel to fix the electrodes, subjects found the experiment *comfortable* (global mean for question 2: 5.19 ± 1.23). According to oral discussions with subjects, it seemed that their curiosity and will to test a BCI was stronger than their apprehension to have gel in their hair.

Question	Answer for real movements	Answer for imagined movements
Q1- Did you get tired because of the experiment? (1: not tired at all, 7: very tired)	1.76 ± 1.04	2.05 ± 1.40
Q2- Did you find the experiment comfortable? (1: not comfortable at all, 7: very comfortable)	5.10 ± 1.26	5.29 ± 1.23
Q3- Did you feel that you could control the spaceship (that is, that you could lift it voluntarily?) (1: you didn't feel you could control it at all, 7: you controlled it perfectly)	3.95 ± 1.80	2.81 ± 1.86
Q4- Did you feel frustration or annoyance during the experiment? (1: neither frustration nor annoyance, 7: a lot of both frustration and annoyance)	2.33 ± 1.56	3.29 ± 1.65

Table 6.1: Average marks given by the subjects in the questionnaire, for the two conditions.

Concerning the *control*, it seems that subjects felt to have an average *control* over the spaceship using real movements whereas they felt to have a lower *control* using imagined movements. As expected, subjects had significantly more trouble *controlling* the spaceship using imagined movement than using real movements ($W = 79$, $p < 0.01$). Globally, subjects were able to assess properly their performances, as the marks they gave for question 3, related to their feeling of *control*, are significantly correlated with the HF differences obtained (Spearman correlation $r_s = 0.63$, $p < 0.00001$). Concerning only imagined movement games, the marks given by subjects to question 3 are significantly correlated with both the HF difference and the TP rate they obtained ($p < 0.05$). Interestingly enough, this correlation is slightly higher between the marks and the TP rate ($r_s = 0.56$, $p < 0.01$) than between the marks and the HF difference ($r_s = 0.51$, $p < 0.05$). This is not the case for real movement games for which there is no correlation between the marks and the TP obtained ($r_s = 0.31$, $p > 0.05$). This might suggest that for a difficult task such as lifting the spaceship using imagined movements, subjects paid more attention to the fact that the spaceship went up when it should have (TP) than when it should not have (FP).

Finally, questionnaire answers showed that subjects found real movement games not really *frustrating* or *annoying* whereas imagined movement games where more *frustrating* and *annoying*. The difference between the two conditions is significant ($W = -64$, $p < 0.05$). This *frustration* might be due to the increased difficulty to lift the spaceship with imagined movement. However and surprisingly, there is no correlation between the *frustration* felt by subjects during the imagined movement games and their performance, i.e., the HF difference they ob-

tained, ($r_s = -0.11$, $p > 0.05$) nor between the *frustration* felt and the subject impression of *control* ($r_s = 0.26$, $p > 0.05$). One explanation could be related to the absence or lack of visual feedback. Indeed, during imagined movement games, subjects had generally less feedback as the spaceship was lifted up less often or it was lifted up less high and stayed in the air a shorter time. This may suggest that less feedback leads to more frustration, whatever the performance. This seemed to be confirmed by oral discussions with subjects.

6.3.2.2 Qualitative data

Thanks to the use of open questions, the questionnaire enabled us to investigate which kinds of imagined movements the subjects performed, as well as to obtain their remarks and comments concerning the application itself.

Regarding the kinds of movement imagined by the subjects, it is interesting to note that a large variety of strategies where employed. For instance, subjects reported that they imagined themselves swimming, running, taping their feet, braking and accelerating, walking or using stairs. A subject even reported that he was "imagining putting his feet behind the head". Unfortunately, this strategy did not enable him to control the spaceship (HF difference = -2). 33 % of the subjects (7/21) reported they imagined the same foot movement as the one they did in the real movement game, whereas 48 % (10/21) reported they imagined a different movement. On average, subjects for whom the real and imagined movements were the same obtained better results (mean HF=3 ± 2.67) than the others (mean HF=0.8 ± 1.75). However, this difference is not statistically significant (Mann-Whitney test $U_{7,10} = 20$, $p = 0.16$), but it would be interesting to study this point further in the future, by using a dedicated experiment. The questionnaires also revealed that 33 % of the subjects (7/21) imagined themselves as performing a sports movement. Interestingly, 62 % of the subjects (13/21) reported they used a single strategy during the experiments whereas 33 % (7/21) reported they used several strategies. However, there is no difference between these two groups in terms of performances ($U_{13,7} = 41$, $p = 0.75$).

Concerning the free remarks of subjects, it is interesting to note that 3 subjects complained about the difficulty to concentrate considering the environment they were in, i.e., an exhibition. They would have prefered to be in a more isolated place. This point appears as a strong problem which is independent from algorithms and it would have to be resolved, in a way or another, in order to use BCI in public places. Most subjects reported that they found the application and the interface well designed, enjoyable and motivating. These remarks are in line with previous studies that showed that VR could increase the motivation of subjects for BCI [LSF+07]. Finally, another valuable comment made by 2 subjects concerned the frustration they felt due to the absence or lack of feedback when they did not succeed to lift the spaceship. They suggested that an additional or more complete feedback could be used in order to give them more information and, possibly, improve their learning.

6.4 Conclusion

This chapter reported on an evaluation of a self-paced BCI application conducted with 21 naive subjects. We studied both performances and preferences of subjects placed voluntarily in a

challenging situation: first-time session, no human learning, no machine learning of the mental state to be detected, “out of the lab”, use of a single EEG channel. Subjects interacted with an entertaining Virtual Reality application and were asked to control the take-off and height of a virtual spaceship by using real or imagined foot movements.

Results showed first that, without training, roughly half of the subjects were able to control the application by using real foot movements, and a quarter were able to control it by using imagined foot movements. These relatively modest performances enhance the need for algorithms that can learn and use subject specific information, such as FuRIA or FIS, in order to increase performances. It should also be recalled that the context was challenging for the subjects, as it was their first BCI use, they were in a noisy environment and they used a BCI based on a single channel and a single feature.

Taken together, the results of the subjective questionnaire stressed the importance of the mental strategies and the visual feedback. More precisely, results suggested that a lack or an absence of feedback during the detection of the non-control state could lead to an increased frustration for the subjects. Results also suggested that subjects could give more attention to true positives than to false positives during games based on imagined movements. Thus, when designing a self-paced BCI, we recommend to provide subjects with a continuous feedback, and to provide feedback (possibly a different one) even when the non-control state is detected. For instance, we could imagine a feedback indicating the subject how close he is from the intentional control state. This may be likely to reduce the subject frustration, to improve his motivation and possibly accelerate his learning. In addition, results of this evaluation showed that subjects enjoyed their BCI experience with the VR application, thus confirming that VR could indeed increase the motivation of BCI users [LSF+07].

This first study of brain-computer interaction with virtual worlds has enabled us to gather some relevant information for further designs. First, concerning the BCI itself, it is clear that we should use trainable algorithms in order to obtain better performances. Second, concerning the VR application, we should pay a particular attention to the feedback provided to the user. Indeed, this feedback should be continuous, provided at any time and as informative as possible. These guidelines have enabled us to design a second BCI application that enables a user to visit a virtual museum by thoughts. The related BCI, VR application and the associated experiment are described in the next and last chapter of this manuscript.

Chapter 7

Exploring a Virtual Museum by Thoughts with Assisted Navigation: First Steps

7.1 Introduction

In the field of BCI-based interaction with VR, the works achieved so far (see Section 1.7.2) have shown that brain-computer interfaces could be used as promising interaction devices for exploring virtual environments. However, current interaction techniques used for navigating Virtual Environments (VE) by thoughts remain relatively basic and do not compensate for the small number of commands available for the subject. Indeed, interaction techniques presented so far are mostly based on very low-level commands, with generally a direct association between Motor Imagery (MI) tasks and turning left or right, or moving forward in the VE. We believe that high-level commands should be more exploited in order to provide a more flexible, convenient and efficient system (in terms of speed of task execution). In such a system, most of the complex tasks would be carried out by the system whereas the user would only have to provide a few and very high-level commands to accomplish the desired tasks. Such a principle is currently being applied to brain-computer interactions with robots [dRMRMG04, VML$^+$07, RBG$^+$07]. We believe that appropriate interaction techniques in VR should be designed in order to use more efficiently the few commands provided by a BCI.

In this chapter, we present a BCI-based application of Virtual Reality (VR) which enables a user to visit a virtual museum by using thoughts only. In order to exploit efficiently the small number of commands provided by a BCI, we propose here a novel interaction technique for BCI-based VR applications. This interaction technique enables the user to send high-level commands, leaving the application in charge of most of the complex and tedious details of the interaction task. Indeed, our interaction technique proposes the user to explore the museum by selecting points of interest such as artworks or navigation points (i.e., cross roads, room entrance, etc). The user can select these points thanks to successive binary choices. Once a given point of interest is selected, the application will be in charge of performing the interaction task such as navigating from the current point to the selected point or observing a given artwork.

In addition, we present a self-paced BCI system which can provide its users with 3 different commands. This BCI is used as the interaction device of our VR application.

A first evaluation of our system is achieved in order to assess if a subject can navigate in the museum by using our interaction technique and in order to compare performances with the state-of-the-art techniques.

This chapter is organized as follows: Section 7.2 presents the interaction technique we proposed in order to explore a virtual museum. Then, Section 7.3 presents the design of the self-paced BCI used to interact with this VR application. Section 7.4 describes the first evaluation achieved and the obtained results. Finally, Section 7.5 discusses the results and concludes.

7.2 The VR application and the interaction technique

The aim of our VR application is to enable a user to visit a virtual museum by using thoughts only. This application should enable this user to navigate in the virtual museum and to look at the artworks displayed in this museum. This VR application should be controlled by a self-paced BCI system which can provide its user with 3 commands, respectively associated to left hand, right hand and foot Motor Imagery (MI). In order to provide the user with a flexible interface and several possible choices of tasks though using only 3 mental commands, we propose a new interaction technique which relies on a binary tree approach. This technique is described in the following.

7.2.1 Selection of interaction tasks

In our application, the tasks available to the user at a given instant are organized according to a binary tree structure. This means that the possible tasks are recursively divided into two subsets of tasks and are placed at the node and leaves of a binary tree. In order to select a specific task, the user should first select one of the two subsets of tasks displayed, by using either left hand motor imagery (to select the first subset) or right hand motor imagery (to select the second subset). This choice done, the selected subset of tasks is again divided into two subsets and displayed to the user who shoud make another choice. This means that the current node of the binary tree has been changed to the root node of the left or right subtree. This process is repeated until the selected subset contains only a single task (i.e., until a leaf of the binary tree is reached), which task is then carried out by the system.

BCI systems are not perfect mental state recognizers and they tend to make mistakes by recognizing a mental state instead of another one. Moreover, the user can also make human mistakes and select the wrong task or subset of tasks. The user may also change his mind, and may finally want to do another task. Thus, we also provided the user with an “undo” option. At any time, the user can perform foot motor imagery in order to cancel the last choice he took. As a consequence, the current choice will be changed to a choice which corresponds to the previous node in the binary tree.

Based on this task selection principle, two navigation modes are provided to the user: the free navigation mode and the assisted navigation mode. The user can switch from one mode to the other by using the task selection principle described above. In others words, at the top of the binary tree, the user can select the free navigation mode by using right hand MI, and

the assisted navigation mode by using left hand MI. When the user leaves a given mode by using the undo option (foot MI), the other mode is automatically selected in order to save time. Figure 7.1 displays the architecture of this binary tree.

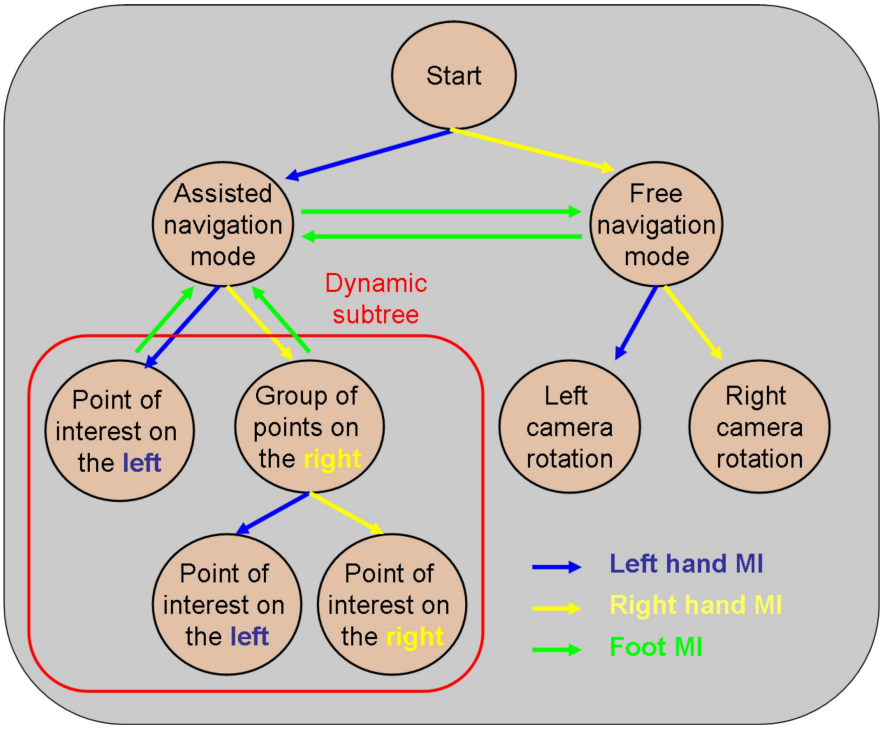


Figure 7.1: Architecture of the binary tree used for task selection. It should be noted that the architecture of the subtree surrounded by a red line is dynamic. Indeed, the number of nodes in this subtree and their arrangement depend on the user's view point (see Section 7.2.2 for details).

It is worth noting that, thanks to the binary tree selection principle, any number of new modes or tasks can be easily added to the interface. The section hereafter describes the two modes implemented in our current application, the core of our contribution being the assisted navigation mode which provides high-level commands to the user.

7.2.2 Assisted navigation mode

In the assisted navigation mode, the user can select points of interest by using the binary tree selection mechanism presented before. The points of interest that the user can select at a given instant depend on his position and field of view, as the user can only select visible points of interest. The points of interest located the farthest on the left of the user's field of view are placed on the left leaves of the binary tree whereas the points located the farthest on the right are placed on the right leaves. As such, the user can naturally use left hand MI to select the set of points on the left, and right hand MI to select the set of points on the right. The points of interest could be either artworks or navigation points.

Artworks represent any object exposed in the museum that is worthy of interest, such as a painting, a picture or a statue. To be identified by our application as artworks, these objects have to be listed in an XML file. This file should contain the name, position, orientation and 3D model of each of these objects. It should also contain the kind of artwork each object is. Indeed, different kinds of interaction can be proposed to the user according to the kind of artwork. For instance, concerning a painting, the user could need to focus and zoom on some parts of the painting in a 2D manner (a painting being generally 2D). On the other hand, concerning a statue, the user could need to turn around the statue to observe it from various points of view. As such, a different kind of interaction is needed for a statue or for a painting, for instance.

Navigation points are points that the user can navigate to. More precisely, in order to walk in the virtual museum, the user just needs to select the next navigation point from the available navigation points, using the binary tree selection technique. The application automatically drives the walk from the current point to the selected one. This relieves the user from all the cumbersome details of the trajectory to go from one point to the other. During this assisted walk, the user can perform foot MI at any time (i.e., undo) in order to stop the walking where he is. This could be useful if the user changes his mind or if, during the walk, the user spots an interesting artwork, not visible previously. Interestingly enough, these navigation points can be generated automatically by our application, by extracting topographical information from the geometry of the virtual museum (see Section 7.2.5).

7.2.3 Free navigation mode

This navigation mode is a simple mode which provides low-level commands to the user. It enables the user to rotate the camera towards the left or towards the right by using left hand or right hand MI respectively. As such, this navigation mode is equivalent to what has been proposed in [FLG+07, LSL+04, FTH08]. In our application, this mode enables the user to look around its current position, in order to localize his next destination or to look for a given artwork. Once this destination or artwork is localized, the user can employ the assisted navigation mode in order to reach quickly the corresponding point.

7.2.4 Graphical representation and visual feedback

Providing relevant feedback to any VR user is essential (e.g., see [SLMA06]). This is particularly important for BCI-based applications as the user can only rely on the feedback to know whether he correcly performed a given mental task [WBM+02]. Moreover, our results from Chapter 6 have suggested that, for self-paced BCI-based VR applications, providing a continuous and informative feedback at any time may reduce the user's frustation and improve his learning.

In our application, various colored icons are displayed dynamically on screen (see Figure 7.2). Among these icons, 3 are continuously displayed: these icons provide feedback on the mental states identified by the classifier. These 3 icons represent a left hand (in blue), a right hand (in yellow) and feet (in green) and are naturally associated to left hand MI, right hand MI and foot MI, respectively. When the VR application receives a given mental state from the BCI

classifier, the size of the corresponding icon increases. As long as the same mental state is being received, the size of the icon keeps increasing until the number of consecutive states required is reached. Indeed, to make the control of our VR application more robust, we require that the same mental state is received several times in a row in order to execute the corresponding command. In other words, we used a dwell time (see Section 7.3 for details). Dynamically changing the icon size depending on the classifier output enables us to provide feedback to the user even when the non-control state (any mental state except the targeted MI states) is finally detected, as our results from Chapter 6 suggest. When the icon reaches its maximum size, the corresponding command is executed. This command is also represented on screen under the form of an icon placed next to the corresponding mental state icon, and displayed with the same color. This aims at informing the user of what will happen if he performs a given mental command. As the available commands depend on the mode used (e.g., assisted or free navigation mode), the icons representing the commands are also dynamically changed.

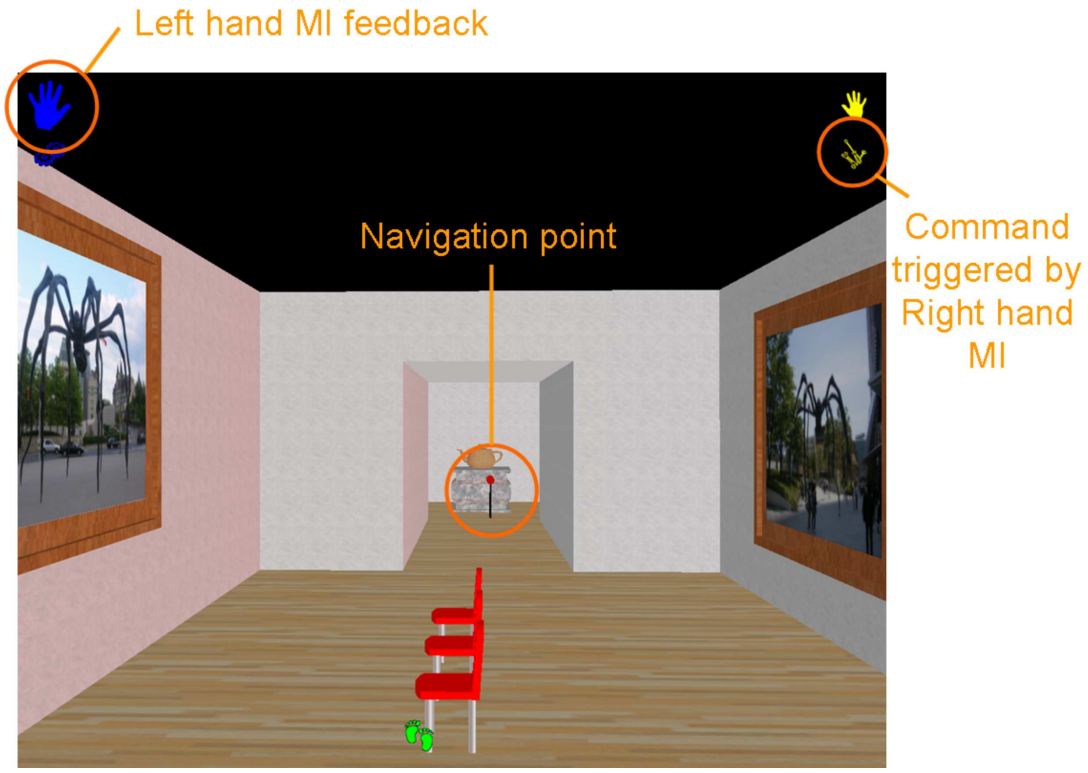


Figure 7.2: Graphical representation of the BCI-based virtual museum application.

The visible points of interest are displayed in the museum using colored pins (see Figure 7.2). When using the assisted navigation mode, the user can select these points to go automatically from point to point. In this mode, the point or the set of points that can be selected using left hand MI are displayed in blue (the left hand icon is also displayed in blue) whereas the point or the set of points that can be selected using right hand MI are displayed in yellow (the right hand icon is displayed in yellow). The other points, which cannot be selected anymore,

are displayed in red and black. Figure 7.3 displays an example of selection of navigation points, on which we can see these colored pins. When selecting these points of interest, no command icon is displayed on screen, as the points of interest are colored themselves according to the mental command needed to select them.

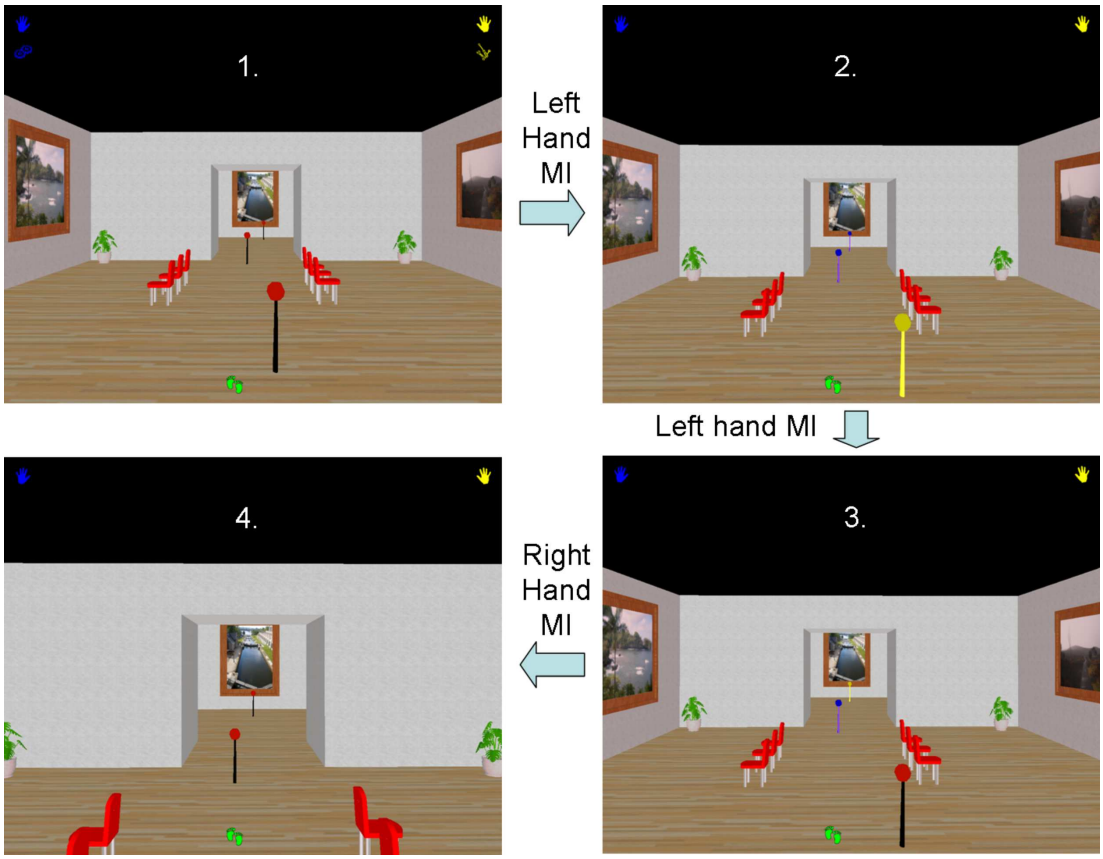


Figure 7.3: Example of use of the assisted navigation mode, with navigation points. Here, the user first selects the assisted navigation mode by using left hand MI from the starting node of the binary tree. Then, he selects the set of two navigation points located on the left by using again left hand MI. From these two points, he finally selects the one on the right by using right hand MI. The application automatically drives the walk to this point. Picture 4. displays the user's view during this walk.

7.2.5 Implementation

Various tools have been used in order to design this application. In this section, we first present the tool and the algorithm used to generate the navigation points and to compute automatically trajectories between these points, i.e., to perform path planning. Then we present the general software architecture of our VR application.

7.2.5.1 Generation of navigation points and path planning

In order to generate the navigation points, and to automatically compute the trajectories between two of these points, our application relies on the TopoPlan (Topological Planner) algorithm, developped by Dr. Fabrice Lamarche. TopoPlan is a computer software dedicated to the automatic processing of 3D virtual environments in order to enable entities to navigate automatically in these environments. Using the non-organized 3D geometry, TopoPlan first computes an exact 3D spatial subdivision of the corresponding environment. From this subdivision, and using various morphological information on the navigating entity (e.g., size, width, ...), Topoplan can extract a 3D topology of the environment. This topology defines the areas that can be navigated and their accessibility. From this 3D topology, different constraints are extracted, such as the boundaries of the obstacles or the borders of stair-steps. These constraints are used to compute a set of 2D topologic maps. Such 2D topologic maps are spatial subdivisions of the environment plane (in 2D). For 3D environments with multiple levels, several 2D maps are extracted so as to cover the whole environment. These topologic maps are computed using a slightly modified version of the constrained Delaunay triangulation [LD04]. Two kinds of constraints are used in this triangulation: obstacles or steps. The generated triangulation is a partition, in triangles, of the convex hull of the environment. This partition contains three kinds of segments: obstacle, step and free segments. As an example, Figure 7.4 displays the result of the subdivision of the building of the museum used in Section 7.4.1 (see Figure 7.9 for a view of the 3D model of this museum).

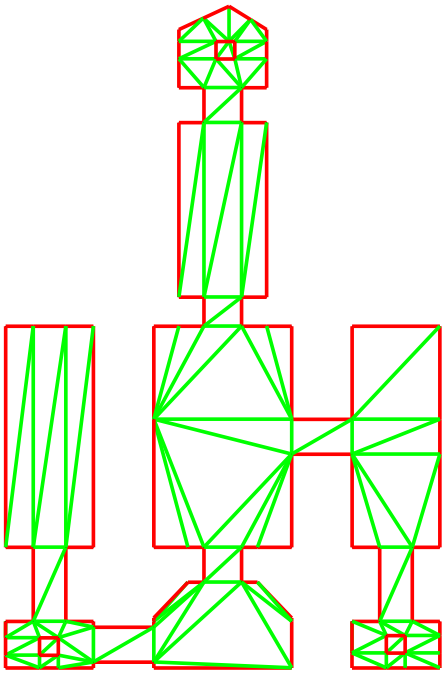


Figure 7.4: Example of the subdivision map of the building of a museum. The obstacles (here walls) are displayed in red lines whereas the results of the subdivision are displayed in green lines.

Each segment with the type “step” or “free” can be crossed in order to navigate between the triangles. In order to automatically compute a path within the environment, a roadmap is generated for each topological map. Such a roadmap is composed of a set of points representing good configurations for the navigating entity. We denote as a configuration a 2D point on which an entity can be located without colliding with the environment. As such, a configuration takes into account the size of the entity. In the roadmap, these configurations are connected by straight line paths along which the entity can navigate. Each configuration is generated on a segment with the type “step” or “free”. The configurations are generated so as to maximize the distance to the surrounding obstacles. Then, a path is generated between two configurations if and only if these two configurations are generated on two different segments which delineate the same triangle, and only if the path which connects them is free of collision. Figure 7.5 displays, as an example, the roadmap extracted from the museum building by using TopoPlan.

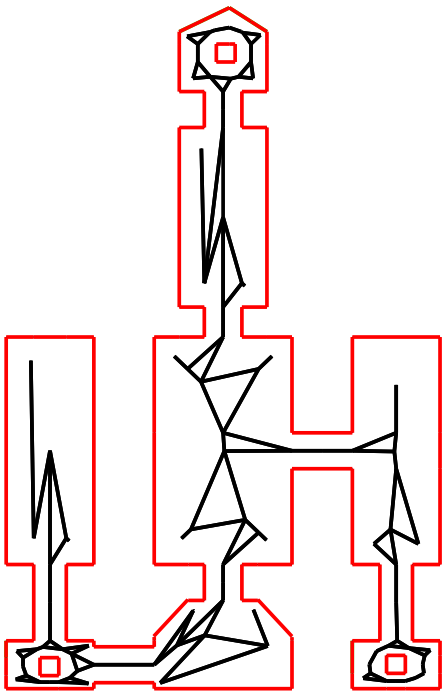


Figure 7.5: Example of a roadmap, in black lines, extracted using TopoPlan from a museum building.

By using these different data structures, it is possible to plan and to follow a path in order to navigate automatically from one configuration of the environment to the other.

Concerning the assisted navigation mode of our application, the challenge was to find a small number of significant navigation points that enable the user to navigate the VE by selecting these points. To this end, we propose to filter the roadmap in order to extract a subset of relevant configurations. To do so, all the configurations from the roadmap are first given a mark. This mark corresponds to the distance between the configuration and the nearest obstacle. In other words, the highest a configuration mark, the more this configuration maximizes

the coverage of the environment. The different configurations are then analysed by decreasing order of their associated mark. Let us define $S(p)$, a function which associates to each configuration p its successors in the graph representing the roadmap. A configuration $s \in S(p)$ is removed from the roadmap if and only if for all configurations $x \in S(s)$, the path (x, p) is valid. If a configuration s is removed, the paths (x, p) such that $p \in S(s)$ are added to the roadmap. This process is repeated until the algorithm converges. This algorithm enables us to filter the roadmap by keeping, in priority, the configurations which maximize the visibility on the VE. As such, it can greatly simplify the roadmap while ensuring that there is at least another visible configuration which can be selected to navigate the VE. The navigation points used in our virtual museum application correspond to the configurations that have been selected after the filtering.

However, even if these points appeared as functional, some of them did not have an optimal position. For instance, some navigation points were located in the corners of the rooms, which made it inconvenient to go from one room to the other or to observe the room. As such, after this automatic generation of points, we can perform a manual optimization of their positions and eventually remove/add some points. However, it is worth noting that Topoplan can still generate automatic trajectories from point to point, be the points generated automatically or by hand.

7.2.5.2 General software architecture

In order to use our BCI-based VR application, some offline operations are required beforehand. The main requirement is to have a 3D model of the virtual environment. For our evaluations, we used the Google SketchUp[1] modeling tool in order to create the 3D model of the virtual museum. In addition, an XML file, which contains the pathname of the artwork 3D models and the positions and orientations of these models in the museum, should be written. From the museum 3D model and the XML file, our application uses Topoplan to automatically generate the topology, the roadmap(s) and the points of interest of the virtual museum. Optionally, these points of interest can be manually optimized. Concerning the BCI system, we used the OpenViBE[2] platform [RGC+07]. OpenViBE is indeed used to generate the self-paced BCI using a set of training EEG signals recorded previously from the subject. The general software architecture of these offline operations is displayed in Figure 7.6.

Concerning the online use of our VR application, the corresponding software architecture is displayed in Figure 7.7. Within this architecture, OpenViBE is used to implement the BCI, here used as an interaction device for the VR application. Indeed, OpenViBE sends commands to the VR application via VRPN (Virtual Reality Peripheral Network) [THS+01]. The kernel of our VR application is the interaction engine. This software module processes the commands received from the BCI in order to perform the corresponding interaction tasks. It is also in charge of automatically generating the dynamic part of the binary tree, according to the user's

[1] http://sketchup.google.com/

[2] http://www.irisa.fr/bunraku/OpenViBE

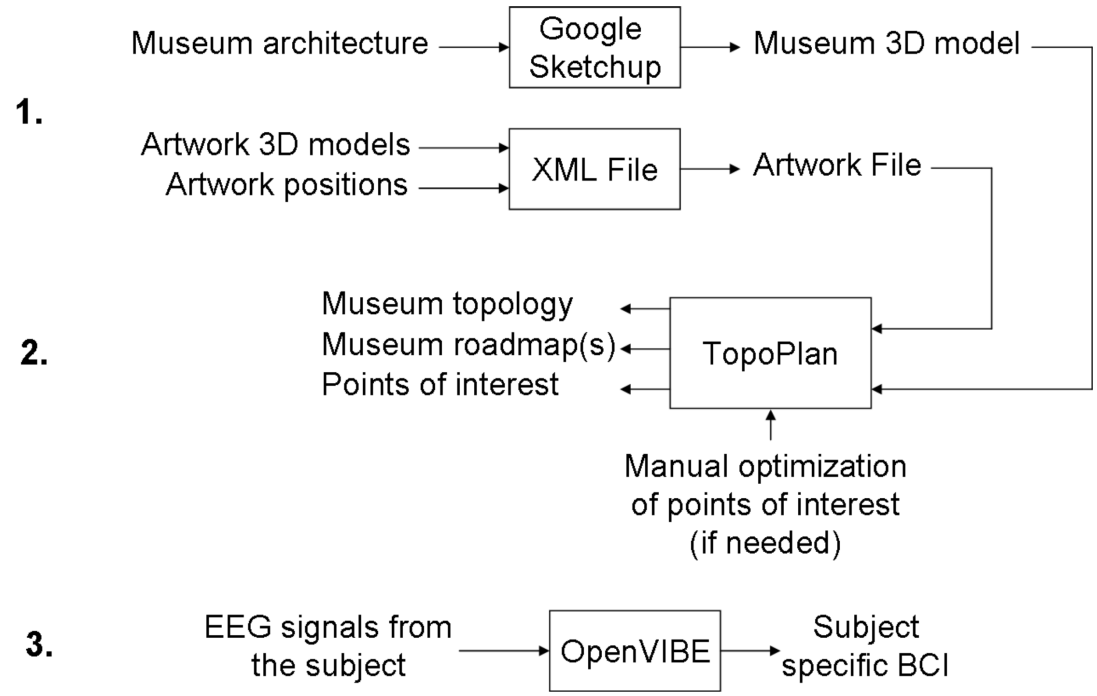


Figure 7.6: Software architecture for offline operations.

view point in the museum. The interaction engine uses the Ogre[3] 3D engine to render the museum and to display the visual feedback. Finally, the interaction engine uses Topoplan in order to perform the automatic navigation between two points of interest.

It is worth noting that any virtual museum could be used within our application. Indeed, generating the navigation points can be done completely automatically using only the geometry of the VE, thanks to Topoplan. Alternatively, the points can be generated by hand simply by providing their coordinates. Moreover, the generation of the trajectories from point to point is done completely automatically, still thanks to the analysis of the VE geometry by Topoplan. In this chapter, we focused on interaction tasks in a virtual museum. However, as our application has been designed in a generic way, it should enable users to interact with other virtual environments such as virtual apartments for instance.

7.3 The Self-Paced BCI

As mentioned earlier, our VR application is controlled by a self-paced BCI based on MI which can provide its user with 3 mental commands. As such, this BCI is a 4-state self-paced BCI. It can indeed recognize the 3 mental states associated to each command - the Intentional Control (IC) states - plus the Non-Control (NC) state, i.e., any mental state that does not correspond

[3] http://www.ogre3d.org/

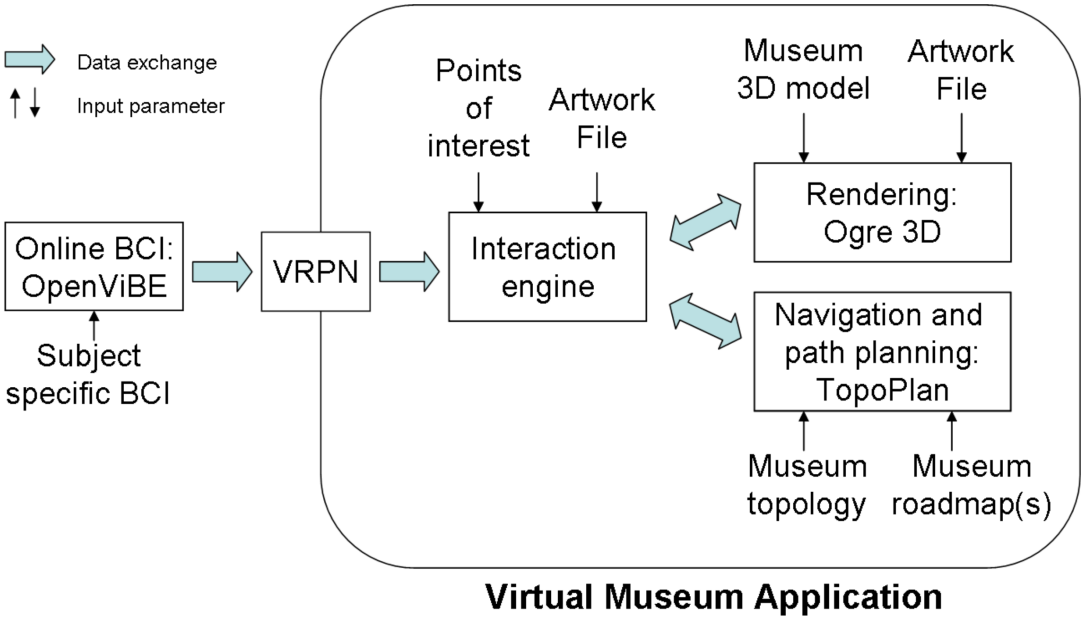


Figure 7.7: Software architecture for online operations.

to a control command. In this chapter, we describe the design of a 4-state self-paced BCI based on MI. The following sections describe the electrodes used in our BCI as well as the preprocessing, feature extraction and classification methods we used to design it.

7.3.1 Electrodes used

In order to record the MI brain signals, we used 13 EEG electrodes, located over the motor cortex areas. These electrodes were FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz and CP4 according to the 10-10 system [AES91] (see Figure 7.8).

7.3.2 Preprocessing

As the preprocessing step, we first band-pass filtered the raw EEG signals in the 4-45 Hz frequency band as this frequency band is known to contain most of the neurophysiological signals generated by MI [PN01]. Moreover, performing such a filtering can reduce the influence of various undesired effects such as slow variations of the EEG signal (which can be due, for instance, to electrode polarization) or power-line interference (50 Hz in France). To achieve this filtering, we used a Butterworth filter of order 4. In order to enhance the brain signals of interest, we also used a Surface Laplacian (SL) spatial filter [MMDW97] over C3, C4 and Cz, leading to three Laplacian channels C3', C4' and Cz' obtained as follows:

$$C3' = 4C3 - FC3 - C5 - C1 - CP3 \tag{7.1}$$

$$C4' = 4C4 - FC4 - C6 - C2 - CP4 \tag{7.2}$$

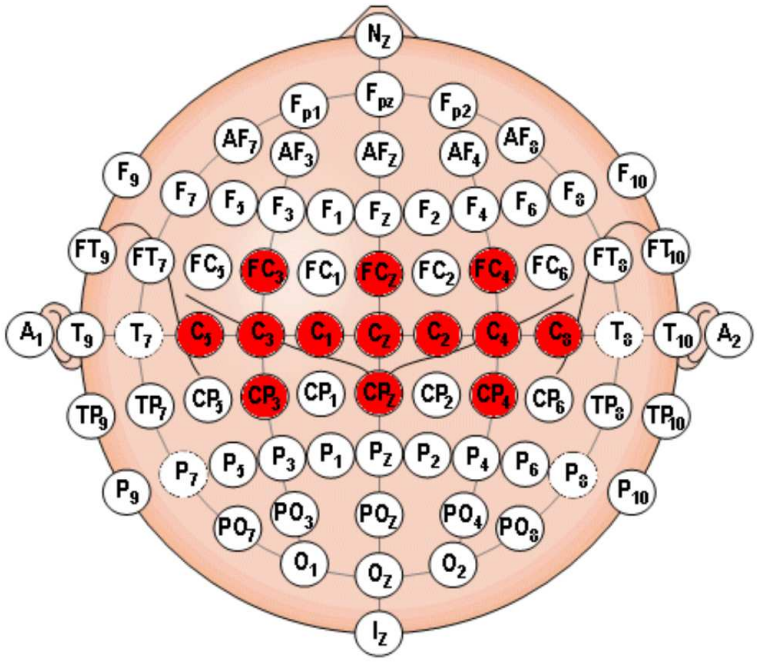


Figure 7.8: Placement of electrodes in the 10-10 international system. The 13 electrodes we used in our BCI are displayed in red.

$$Cz' = 4Cz - C1 - C2 - FCz - CPz \tag{7.3}$$

Indeed, channel C3', Cz' and C4' are respectively located over the right hand, foot and left hand motor cortices. Features were extracted from these three new channels.

7.3.3 Feature extraction

For feature extraction we used Band Power (BP) features. To obtain a more efficient BCI, we extracted several band power features for different frequency bands for the different Laplacian channels and selected the most relevant ones using the Sequential Forward Floating Search (SFFS) feature selection algorithm [PFK94]. This algorithm is indeed one of the most popular and efficient feature selection techniques [JZ97]. More precisely, we investigated BP features extracted in 2 Hz wide frequency bands between 4 and 34 Hz, with a 1 Hz step, and selected the 12 most efficient features using the SFFS algorithm. We indeed observed that using more than 12 features did not increase significantly the performance whereas it increased the computational burden. As our BCI is a self-paced BCI, it requires continuous classification of EEG signals. Consequently, we extracted a BP feature vector 16 times per second, over the last 1 s time window (i.e., using a sliding window scheme).

7.3.4 Classification

For classification we used the Fuzzy Inference System (FIS) presented in Chapter 3, i.e., the Chiu's FIS [Chi97b]. Indeed, as presented in Chapter 5, non-linear classifiers such as FIS have been shown to be the most efficient ones for self-paced BCI design. In order to design a self-paced BCI we relied on a pattern rejection approach (see Chapter 5). More precisely, we used the reject class technique, as this seemed to be the most efficient method according to our results from Chapter 5. With this technique, our FIS classifier will have to deal with 4 classes: one for each mental state used for control (IC states), and one class for all other mental states (NC state).

In order to make our self-paced BCI more robust, we also used a dwell time and a refractory period [TGP04]. When using a dwell time, a given control command is generated only if the classification identifies the same class N_D times in a row. Similarly, when using a refractory period, the N_R classifications which immediatly follow the identification of an IC state will be forced to the NC state. These two techniques lead to less False Positives (FP), i.e., to less identifications of an IC state instead of an NC state. In our system we used $N_D = N_R = 7$ (these values were defined experimentally).

7.4 Evaluation

We performed a first evaluation of our application in order to know whether the interaction technique we proposed was usable and efficient. To do so, we studied the performances of one subject who used our interaction technique to navigate from room to room in the virtual museum. As a comparison, the subject also had to perform the same task by using the current state-of-the-art navigation technique in VR when using a BCI: the technique of Scherer *et al* [SLS+08], i.e., turning left, turning right or moving forward by using left hand, right hand and foot MI, respectively (see [SLS+08] or Section 1.7.2 for more details about this work). This enabled us to compare our high-level approach to a low-level one. It should be noted that we used the same BCI design (the BCI presented in Section 7.3) for both interaction techniques. Consequently, this BCI design is different from the one used by Scherer *et al* [SLS+08], as we used a different classification algorithm, different electrodes, different signal processing techniques, etc. Moreover, in this chapter we aim at comparing the interaction techniques, but not the BCI designs. It should also be mentioned that we did not use a refractory period with the interaction technique of Scherer *et al*. Indeed, this enabled the user to maintain the MI task in order to maintain the movement in the museum (rotation or forward translation) and as such to move continuously rather than by periods. During such continuous forward translations, the user moved by about 25 cm for each corresponding classifier output, the classifier providing an output 16 times per second.

As a measure of performance, we evaluated the time needed by the subject to perform different navigation tasks. As this evaluation was dedicated to navigation tasks, artworks were not considered during this experiment, which means they could not be selected by the user. However, artworks were still displayed in order to provide an engaging VE. The next sections describe the virtual museum used, the population and apparatus, the task the subject had to do, the experimental procedure and finally the results obtained.

7.4.1 Virtual museum

As the virtual environment used in our evaluation, we developed a fictional virtual museum. This museum contains several pictures of landmarks from various countries as well as some statues (see Figure 7.2). This museum is composed of 8 different rooms each one containing either several pictures or a statue. The architecture of this museum is displayed on Figure 7.9.

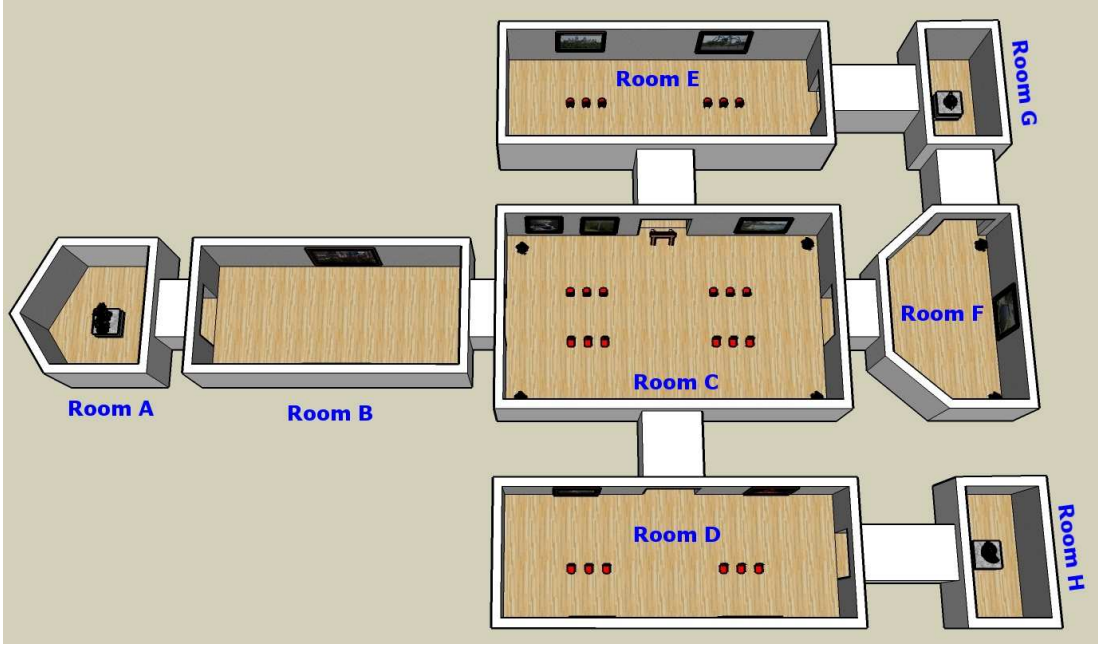


Figure 7.9: 3D model of the virtual museum used in our experiment.

7.4.2 Population and apparatus

For the experiment, EEG signals were recorded using a Nexus 32b EEG machine from the Mind Media company, at a sampling frequency of 512 Hz. One subject participated to this experiment (male, 25 years old), who had a previous MI-based BCI experience. The experimental setup is displayed on Figure 7.10.

7.4.3 Task

For this evaluation, the subject had to navigate from one room to another as fast as he could. The navigation tasks were categorized into three groups according to the distance between the starting room and the finishing room: short, medium or long. A short navigation task consisted in passing through 3 rooms, e.g., going from Room A to Room D (please refer to Figure 7.9 for the room names and positions) ; a medium navigation task consisted in passing through 4 rooms, e.g., going from Room A to Room H ; and finally a long navigation task consisted in passing through 5 rooms, e.g., going from Room A to Room E, i.e., by passing through Room G.

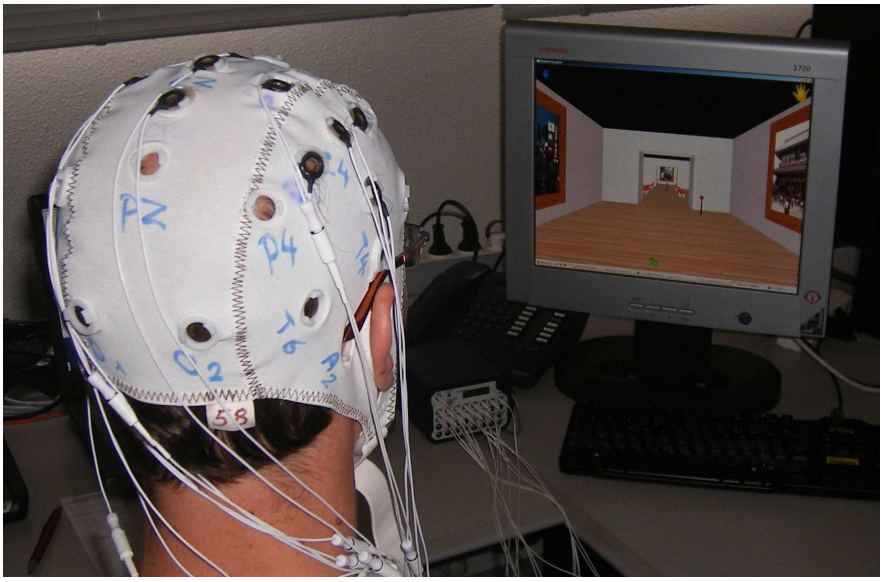


Figure 7.10: Setup of the experiment.

7.4.4 Procedure

Before participating to the virtual museum experiment, the subject participated in three sessions following the protocol of the 3 class Graz BCI (see [KFN+96]), in order to record training data for selecting the features and training the FIS classifier. These sessions were recorded on a different day than the virtual museum experiment. During a session, the subject had to perform 10 trials for each class (left hand, right hand and foot MI). As no classifier was available, the subject was not provided with any feedback. Once the three sessions were completed, the features were selected using SFFS and the FIS classifier was trained on these data (see Section 7.3).

The subject had to perform each kind of navigation task twice for each one of the two interaction techniques. The order of the tasks was arranged in blocks. Within a block, the subject used a single interaction technique and had to perform once each kind of navigation task (short, medium or long - the actual order being randomized within the block). The subject performed first a block using Scherer's method, then 2 blocks using our method and finally another block using Scherer's method. These blocks were achieved over three different days as the tasks were too tiring for the subject to be carried out in a single day. Each day, the duration of the experiment was approximately between 2 and 2.5 hours, including the time required for electrode montage. At the beginning of each day, the subject participated in three sessions of the protocol of the 3 class Graz BCI (see above). During these sessions, the user was provided with a feedback thanks to the classifier trained during the beginning of the previous day of experiment, or trained using the initial training data if it was the first day of experiment with feedback. It should be noted that the BCI used for these sessions was synchronous, which means that the NC state was not recognized by the BCI. However, the subject was instructed not to perform any MI nor real movement during the inter trial periods as the data in these

periods would be used as examples of the NC state for training the final self-paced BCI. Once the three sessions were completed, the FIS classifier was re-trained on these new data, in order to adapt to the user's current EEG signals and in order to design the self-paced BCI. Then the experiment with the virtual museum could begin.

7.4.5 Results

First, it should be mentioned that the user managed to reach all the rooms he was instructed by using any of the two interaction techniques. This suggests that the user could actually control the application and explore the virtual museum by using the proposed BCI system. As the BCI is self-paced and as the user sends commands at his free will, it is not possible to report on the online classification performances of the BCI. Indeed, we cannot compare the mental state recognized by the BCI with the mental state performed by the subject as we did not know what the user intented to do at a given instant. However, in order to evaluate whether our BCI system works better than a randomly performing BCI (i.e., a BCI which is unable to properly identify any mental state), we simulated a random BCI. More precisely, we sent randomly selected mental states to the application as if it was the mental state identified by the classifier. We performed these simulations twice for each interaction technique, the "objective" of the random classifier being to perform a short navigation task. For each task, even after 30 minutes of simulation (i.e., 1800 seconds), the random BCI was not able to reach the targeted room. By comparison, the average time required to go from one room to another (situated at a distance of 3 rooms or more) with our BCI, independently from the interaction technique or the navigation task, was 331.5 seconds, i.e., approximatly 5.5 minutes (see tables below). This suggests that our BCI indeed provided control to the user.

Table 7.1 displays the time needed by the user to accomplish the different navigation tasks (short, medium or long), according to the interaction technique.

Table 7.1: Time (in seconds) needed by the user to accomplish the different navigation tasks using the two interaction techniques.

interaction technique	navigation task	1^{st} block	2^{nd} block	mean	overall mean
Proposed technique	long	176	140	158	165.8 ± 35.5
	medium	174	235	204.5	
	short	130	140	135	
Scherer's technique [SLS+08]	long	574	830	702	497.2 ± 228.4
	medium	486	623	554.5	
	short	364	106	235	

These results show that, for our subject, navigating from one room to another by using the interaction technique we proposed is about three time faster, on average, than when using the interaction technique of Scherer *et al*. This difference is however smaller for short navigation tasks. A paired t-test comparing the time needed to navigate using our technique and the one of

Scherer *et al*, over all blocks and navigation tasks, revealed that our technique was significantly faster ($p < 0.05$).

It is also interesting to note that our technique has a small variance for the time needed to accomplish the task, contrary to the method of Scherer *et al*. This suggests that the time needed to go from one place to another with our technique may not vary too much with the distance to cover. Indeed, it takes approximatly the same time to select a point of interest which is close and a point of interest which is far, providing that they are both visible.

The time needed to navigate was of course related to the number of commands that the user had to send. Table 7.2 displays the number of commands needed by the user to accomplish the different navigation tasks, according to the interaction technique.

Table 7.2: Number of commands needed by the user to accomplish the different navigation tasks using the two interaction techniques.

interaction technique	navigation task	1^{st} block	2^{nd} block	mean	overall mean
Proposed technique	long	36	29	32.5	28.5 ± 8.2
	medium	33	33	33	
	short	13	27	20	
Scherer's technique [SLS+08]	long	missing data	142	142	97.3 ± 68.1
	medium	missing data	131	131	
	short	missing data	19	19	

This table shows that the user sent on average 97.3 commands to go from one room to another (situated at a distance of three rooms or more) using Scherer's interaction technique, and only 28.5 commands to do the same thing using our interaction technique. Thus, as we expected, navigating using points of interest enables the user to send only a few high-level commands while leaving most of the work to the application itself. In addition, the user reported that navigating using points of interest was less tiring than with Scherer's technique, as he could relax and rest during the automatic navigation periods from one point of interest to the other.

7.5 Conclusion and discussion

In this chapter we presented a VR application which enables a user to explore a virtual museum by thoughts only. In order to design this application, we proposed a new interaction technique based on points of interest. This technique enables the user to send only a few high-level commands to the application in order to navigate from one point of interest to the other. The actual navigation between two points of interest is carried out by the application itself, leaving the user free to rest or to observe the museum. Interestingly enough, the computation of these points and the navigation between these points can be achieved completely automatically using an algorithm called TopoPlan. In order to select these points of interest, the user can employ a selection mechanism which relies on a binary tree and depends on the user's view point. This

binary tree mechanism provides a unified framework for BCI-based interaction in VR. Indeed, it enables, for instance, to select navigation commands or object manipulation commands in the same way. In addition, we also proposed a design of a self-paced BCI which can issue three different commands, these commands being associated to left hand, right hand and foot motor imagery, respectively.

A first evaluation of our application, using one subject, has been conducted. The first results suggest that our method was efficient as it was faster to navigate using our interaction technique than using the state-of-the-art navigation technique in BCI-based interaction with VR. Naturally, these results should be moderated by the fact that the evaluations were performed with only one subject. It is indeed necessary to further validate our results by evaluating the application on more subjects. However, the results presented here seem promising. They also stress the importance of designing interaction techniques for BCI-based interaction with VR.

Future works could deal with evaluating this application with disabled people, in order to assess if the application can enable them to visit virtually various museums or even other buildings, which they might not be able to do for real. It should also be interesting to further improve the self-paced BCI. Indeed, even if this BCI seemed much better than a random classifier, the time needed to perform a navigation task by using it was also far from what we can do by using classical interaction devices such as keyboards or mice. To this end, we could study, in this online context, other feature extraction or classification techniques and using, for instance, the FuRIA algorithm presented in Chapter 2. It could also be valuable to explore new filtering techniques to select the relevant points of interest provided by TopoPlan. Finally, it could be interesting to explore and add various interaction commands to our binary tree mechanism. For instance, interaction techniques such as Navidget [HDKG08] could be adapted for BCI purposes in order to easily observe the museum virtual statues.

Conclusion

In this manuscript, we have studied EEG signal processing and classification techniques in order to design BCI systems and to use them in virtual reality applications, with three main objectives: 1) increasing the information transfer rate of current BCI, 2) designing interpretable BCI systems, and 3) developing BCI systems for VR applications.

In order to reach these objectives, we have first proposed contributions at the EEG signal processing and classification level. Concerning feature extraction, we have proposed an algorithm named FuRIA, which is based on inverse solutions and fuzzy sets (see Chapter 2). This algorithm can learn and extract a small number of interpretable and discriminant features, these features corresponding to the activity in Regions Of Interest (ROI) and frequency bands. We also proposed the concept of fuzzy ROI and fuzzy frequency bands which revealed to improve the classification performances of the features extracted by FuRIA. Our evaluations suggested that the learnt FuRIA features reflected knowledge consistent with the literature on the analyzed signals. They also revealed that FuRIA features could be used to design BCI with performances comparable to the ones of BCI competition winners.

Concerning classification, we have studied Fuzzy Inference Systems (FIS) for BCI design (see Chapter 3). More particularly, we studied FIS for classifying motor imagery signals. The results of our studies have first shown that FIS classifiers could obtain classification performances similar to the state-of-the-art classifiers used in BCI research, e.g., support vector machines. Our studies have also shown that FIS were interpretable classifiers, from which we can extract knowledge on the signals used for training. FIS also offer the possibility to use a priori knowledge for classification. Finally, our studies have shown that FIS can reject outliers efficiently.

We have combined the FuRIA feature extraction and the FIS classification techniques, along with linguistic approximation methods, in order to build a fully interpretable BCI system which can express what it has learnt from data using words only (see Chapter 4). Our evaluations suggested that the designed BCI actually reflected knowledge expected from the literature when used on movement intention or visual spatial attention EEG signals. The analysis of these last signals has been possible thanks to a collaboration with Dr. Areti Tzelepi from the Institute of Communication and Computer Systems in Greece, and Dr. Ricardo Ron Angevin from Malaga University in Spain. Our results suggest that the proposed algorithm could be used to check the BCI system learnt from data, to present the automatically extracted knowledge to persons without knowledge on classifiers or fuzzy sets and, possibly, to extract knowledge about the brain dynamics.

Finally, we proposed to design self-paced BCI based on a pattern rejection approach (see

Chapter 5). To this end, we first introduced and studied two pattern rejection techniques that have not been used before for BCI research: the reject class, and the thresholds on reliability functions based on the multiple threshold learning algorithm. This has been possible thanks to a collaboration with Dr. Harold Mouchère, from INSA Rennes, in France. We also compared various classifiers and rejection techniques. Our results showed that using a reject class with non-linear classifiers led to the most efficient self-paced BCI. Consequently, this study provided guidelines to increase the information transfer rate of current self-paced BCI.

In order to put these results in relation with our initial objectives, we could mention that objective 1) (increasing the information transfer rate of current BCI) has been addressed by the proposition of FuRIA and the FIS which both reached state-of-the-art results in terms of classification performances. Moreover, both methods can deal with multiple mental states, hence possibly improving the information transfer rate. Our study on pattern rejection techniques finally made it possible to design self-paced BCI with improved classification performances, by using a reject class and non-linear classifiers instead of currently used methods. Concerning objective 2) (designing interpretable BCI systems), it has also been addressed by FuRIA and the FIS, these two methods being also interpretable. Moreover, the results of the evaluation of the method based on FuRIA and FIS that we proposed suggested that the resulting BCI was indeed interpretable.

In a second part, we have proposed contributions for the design of BCI-based VR applications, which addressed our third and last objective, i.e., developping BCI systems for concrete VR applications.

For this purpose, we first studied the performances and preferences of 21 naive subjects who used a BCI to interact with an entertaining VR application in close to real-life conditions (see Chapter 6). Our results first highlighted the need to use subject-specific BCI thanks to machine learning techniques. They also highlighted the increase in motivation triggered by VR. Moreover, this study stressed that when using a self-paced BCI, subjects should be provided with a complete and continuous feedback at any time, even when the non-control state is detected by the BCI, in order to reduce the user's frustration and improve his learning.

Finally, we have developed a VR application which enables a user to visit a virtual museum by using only a BCI (see Chapter 7). To this end, we proposed a novel interaction technique which provides high-level commands to the user, leaving the low-level and tedious aspects of the interaction task to the application itself. The development of this VR application has been possible thanks to the considerable engineering work of Thomas Ernest and Yann Renard from INRIA Rennes. We also proposed a self-paced BCI design which can provide three different commands, based on motor imagery. A first evaluation of our application showed that a subject was able to explore a virtual museum by using our BCI. This first evaluation also suggested that the proposed interaction technique seems to enable the user to navigate from one room of the museum to the other faster than with the current state-of-the-art navigation techniques used in BCI-based VR application.

Taken together, these results suggest that BCI can actually be used as interaction devices for complex VR applications despite the few commands they can provide. Moreover, these results also highlight that BCI are promising tools also for the general public who can enjoy

entertaining VR applications based on a BCI.

Most of the algorithms used and studied in this PhD manuscript have been implemented in the BLiFF++ library (see Annex A), which will be soon available with an open source license. Some of these algorithms (e.g., LDA, FIS, band power features, ...) have also been implemented for real-time operation within the OpenViBE BCI platform [RGC+07], which will also be released soon with an open source license. Please refer to the website of the project for more details: `www.irisa.fr/bunraku/OpenViBE`.

Future work

The work presented in this manuscript left some questions unanswered and, as such, some future works.

Concerning the FuRIA algorithm, it would be interesting to study the influence of both the spatial resolution (number of electrodes used, number of voxels in the chosen head model) and the frequential resolution (number of frequencies investigated) on the performances, in order to possibly reduce the training times and/or improve the quality of the feature extraction. It would also be interesting to take into account the temporal information in FuRIA. To this end, an attractive possibility would be to replace the classical band-pass filters by wavelets, which have proven to be particularly adpated for neuroelectric signals and especially EEG [SBRS99].

Concerning our work on fuzzy inference systems, future work could deal with the exploration of different FIS such as NEFCLASS [Nau97], on different EEG data and with the comparison of FIS with other interpretable classifiers such as decision trees.

Concerning our study of rejection techniques, we focused so far on distance rejection (also known as outlier rejection). It could also be interesting to study confusion rejection, in order to increase the fiability of BCI systems by not outputing a command if this command is likely to be erroneous.

Finally, concerning the virtual museum application, future works are dedicated to more evaluations. Similarly, assessing the application with disabled people could lead to the design of entertaining and/or engaging VR applications adapted to their needs and preferences.

Perspectives

In addition to the future works mentioned above, the PhD thesis work also paved the way to further long-term research. Some of these aspects are described below.

Towards a unified approach using implicit surfaces

Implicit surfaces [Blo00, OM95, CG06] are powerful tools designed to manipulate and design complex geometric shapes. In this manuscript we have proposed to use inverse solutions and Fuzzy Inference Systems (FIS) for BCI design. It appears that these two methods could be represented using the same formalism, namely, the implicit surface formalism. Indeed, fuzzy

membersip functions used in FIS can be exactly seen as implicit surfaces whereas inverse solutions use brain Region Of Interest (ROI), which regions, being geometric shapes, can also be modelized using implicit surfaces. We believe that such a modeling could lead to interesting results and may enable us to use new and efficient algorithms from the geometric modeling community in order to design efficient BCI or to design new and insightful brain activity visualization techniques. We invite the interested reader to refer to annex B for some hints on how to perform this modeling.

Combining rather than selecting

Concerning the signal processing and classification part of BCI design, we believe that an interesting path to follow could be denoted as the "combining rather than selecting" path. Currently, numerous preprocessing, feature extraction and classification methods have been proposed and explored for BCI design. Even if some of them sometimes proved more efficient than others, no method has been identified as the best. Consequently, we believe that rather than trying to find a single best method, we should go towards combining together existing methods. Indeed, the different methods proposed so far exploit different aspects and properties of the EEG data, and/or rely on different processing and classification schemes. As such, these methods could be used together in a complementary way, and would probably lead to better results when used together, than when using "the best" method alone.

Recent results in the literature have highlighted that combining different kinds of features together could lead to better performances [DBCM04a]. Similarly, several papers have reported that combining several classifiers leads to better BCI performances than when using a single classifier [RG08, Sun07, HdS07]. Thus, it would be interesting to explore which kinds of preprocessing, feature extraction and classification methods are the most complementary when used together, in order to design more efficient and stable BCI. Naturally, studying how efficiently combining these methods should also prove valuable. In this manuscript we studied pattern rejection techniques for self-paced BCI designs, and we showed that the reject class reject option with non-linear classifiers was the most efficient. However, it remains an open question whether better results could be obtained by properly combining the various reject options studied.

Combinations of models could also be a solution towards the design of universal BCI, i.e., BCI that can be used by anybody without requiring the use of subject specific features, classifiers, etc. Indeed, recent results have suggested that combining classifiers [RGMA05b, RG08] or spatial filters [KTBM08] trained on EEG signals recorded at different days could be a solution to the problem of non-stationarities of EEG signals. By extending this idea, combining classifiers and/or features learnt on different subjects might lead to a universal BCI. It could also be imagined that categories of users with similar EEG signals properties could be identified, and that a BCI model could be learnt for each category. Then, by combining these models together, we might obtain a universal model. Possibly, if the resulting model is not efficient enough, it would be possible to incrementaly adapt this global model to a specific user, as it is sometimes done for handwritten character recognition [MAR07].

Still following the moto "combining rather than selecting", it should prove very interesting to combine different kinds of brain signals within the same BCI application. Indeed, cur-

rent BCI applications are based on a single kind of brain signal such as evoked potentials or spontaneous signals, but never use both at the same time. However, the different properties of these signals make them suitable for being used together. For instance, for VR applications, spontaneous signals (e.g., motor imagery) could be used for navigating the virtual world whereas evoked potentials could be used to select virtual objects. In this latter case, the virtual objects that can be selected would be responsible for sending the stimulus necessary to use evoked potentials. For instance, these objects would be randomly flashing (for using the P300) or flickering (for using SSVEP), and would be selected when the user draws his attention on them. Moreover this mode of operation may prove more natural than current applications, as navigation with evoked potentials is not really natural for instance. Naturally, combining different signals would also lead to interesting signal processing and classification challenges, as different detectors or classifiers would have to be used in parallel.

BCI-based VR applications for disabled subjects

In this PhD manuscript, we proposed entertaining VR applications (the "Use-the-force" application and the "virtual museum") controlled using a BCI, and evaluated them using healthy subjects. These BCI-based VR applications hold great promises for such users, for instance concerning video game applications for the general public. However, it should not be forgotten that BCI systems are also a promising communication channel for severely paralyzed persons. We believe that VR applications could also prove really useful for these disabled people.

First, as anyone, disabled people need entertainment, and unfortunately, only few video games are available for them. As such, BCI-based video games in VR could be proposed to these persons. Second, paralyzed persons may not be able to travel or do some sightseeing. As such, BCI-based VR application can be seen as a very promising tool for helping them in having access to numerous cultural experiences. For instance, using BCI technology, paralyzed persons could be able to visit, in VR, various cities, museums or parks. To this end, the interaction technique we proposed in Chapter 7 may prove useful. We indeed used it on a toy museum, but it could be used as well with a virtual representation of any real museum. Proposing VR applications to paralyzed persons would probably require to adapt the interaction techniques and the feedback to their situation and needs.

An advantage of VR, not specific to BCI, is that it allows to perform tests in safe and carefully controlled conditions. Concerning BCI and paralyzed persons, VR could be very useful to test any rehabilitation device controlled using a BCI, such as a wheelchair or a prosthesis. Recently, the Graz group has performed a simple wheelchair simulation in VR for a tetraplegic patient [LFMP$^+$07]. This was a preliminary step, as the wheelchair simulated was very simple and far from being like a real one. However, we believe that such a work is very promising and that such advantages of VR should be more exploited.

Finally comes the issue of using BCI in real-life applications for disabled people. Could we transfer the interaction techniques proposed in VR to the real world? It is indeed a difficult problem, as numerous stimuli or feedback information are displayed in the virtual environment and used for interaction. To translate them into real-life, the patient would need an additional screen to display these feedbacks. This may be inconvenient and, more importantly, this prevents the user from focusing his attention on the real world as he would need to look at this

screen. Consequently, in the long term, we will have to find solutions in order to still provide feedback and stimuli to the user without this user needing to constantly focus his attention on a computer screen. A possible solution could be to use non-visual stimuli, such as audio or haptic stimuli.

Naturally, a lot of work needs to be done in order to achieve the applications mentioned above. As BCI is still a young research field, there is no doubt that the next years will witness tremendous advances in the field and, at the same time, will open the way to new and exciting research challenges.

Appendix A

The BLiFF++ library: A BCI Library For Free in C++

A.1 Introduction

The work presented in this PhD manuscript has lead to a considerable amount of programming, mainly in C++. In order to produce reusable programs, most of the code produced has been gathered and organised as a C++ library. This library is known as BLiFF++, which stands for "A BCI Library For Free in C++". With the aim of having this library being useful for the BCI community, BLiFF++ will be soon provided for free under the terms of the open source license GPL. This annex briefly describes the library and some of its functionalities.

A.2 Library features

BLiFF++ aims at providing to the BCI community a set of tools, implemented under the form of C++ classes, in order to design, test and evaluate BCI systems, mainly for offline analysis. These classes may be gathered into two main categories: classes for designing BCI and classes for analyzing brain data.

A.2.1 Classes for BCI design

Such classes enable the BLiFF++ user to process brain signals in order to identify mental states. As such, BLiFF++ provides the necessary tools to manipulate brain signals and especially EEG signals. It also makes it possible to apply signal processing techniques to such signals, such as frequency filtering (FIR, IIR, FFT), inverse solutions or various feature extraction techniques. From these signals, BLiFF++ proposes to extract and manipulate features, which features could be used as input of a variety of classification algorithms provided by the library. Among these classification algorithms we can quote support vector machine (SVM), linear discriminant analysis (LDA), neural networks such as multilayer perceptron (MLP), fuzzy inference systems (FIS), mahalanobis distance-based classifiers, etc. It is worth noting that various tools are also provided in order to grant these classifiers with reject options or to combine several of these

classifiers together in order to build a meta-classifier. Finally, BLiFF++ also provided various evaluation metrics such as classification accuracy, error rate, confusion matrix, mutual information, ROC analysis, ... Figure A.1 displays some tools offered by BLiFF++ in order to design a BCI, as well as the architecture of the corresponding classes, using the Universal Modeling Language (UML).

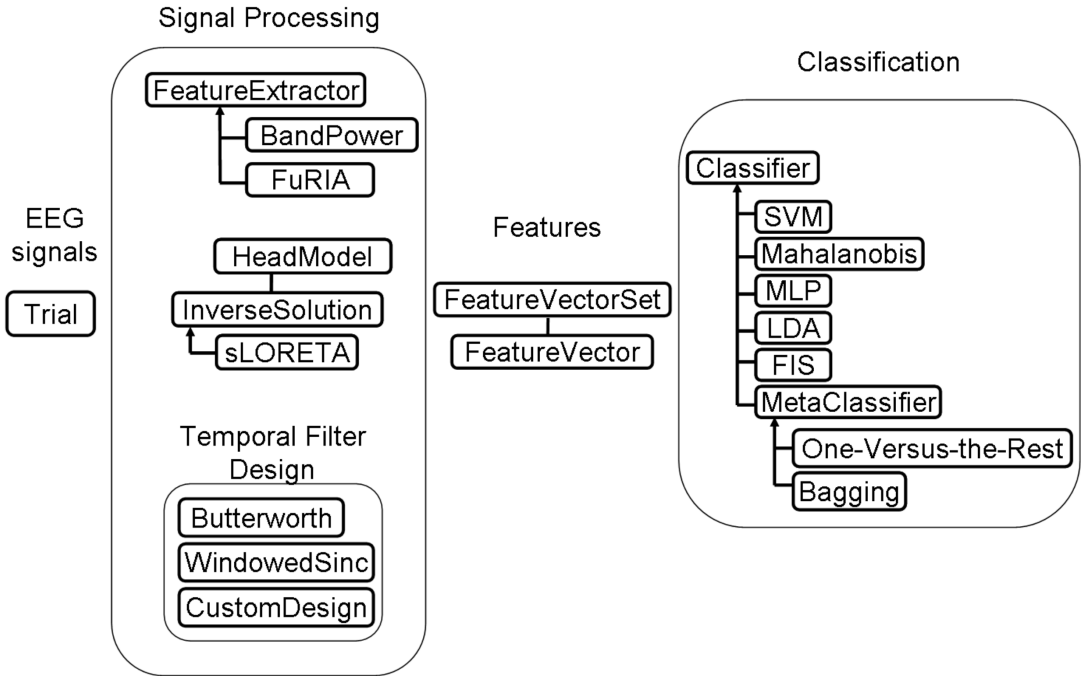


Figure A.1: Classes provided by the BLiFF++ library for designing BCI

A.2.2 Classes for data analysis

In addition to these tools for BCI design, BLiFF++ proposes tools for analyzing brain data. Such tools aim at investigating recorded data and at finding patterns or any other relevant information in them. For such a purpose, in addition to signal processing tools, BLiFF++ provides classes that perform data clustering (fuzzy C means, mean shift, ...), various statistical analysis methods (ANOVA, t tests, ...), feature selection techniques such as Sequential Forward Floating Search, ... Figure A.2 displays tools offered by BLiFF++ in order to analyze brain data, as well as the corresponding class architecture in UML.

This library has been used for most studies presented in this manuscript.

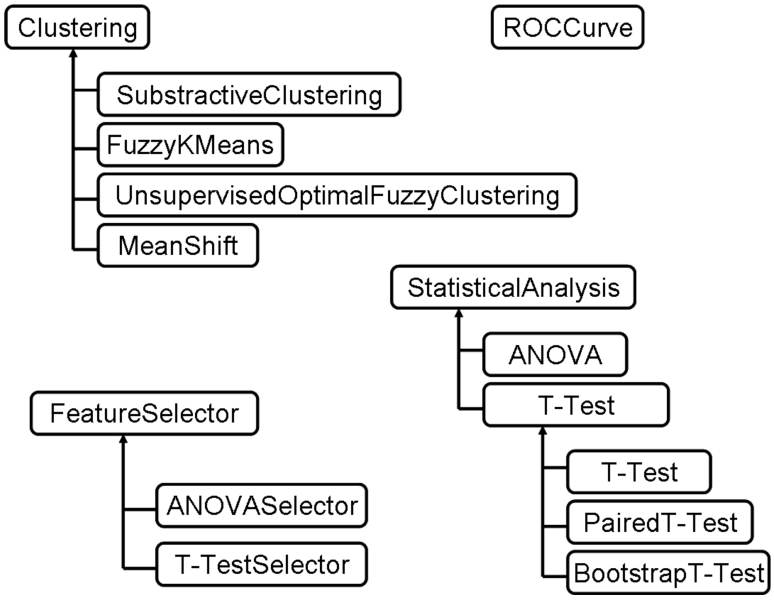


Figure A.2: Classes provided by the BLiFF++ library for EEG data analysis.

A.3 Test case: designing a motor imagery based BCI

As an example of how this library works and how simple it is, this section describes the design of a simple motor imagery based BCI. In order to design this BCI, the program below first band-pass filters the data in the 3-35 Hz frequency band, then applies a surface Laplacian spatial filter as preprocessing. Then, it extracts band power features from the signals, in the Mu and Beta bands. The resulting features are classified using a linear discriminant analysis (LDA) classifier.

```
//reading EEG signals
//we assume that the read EEG are recorded using electrodes:
//FC3, C5, C3, C1, CP3, Cp4, C2, C4, C6, FC4
//training data
vector<Trial> EEGsignalsTrain =
      EEGReader::readData(TRAIN_SIGNAL_FILE, TRAIN_STIMULATION_FILE, SAMPLING_FREQ);
//testing data
vector<Trial> EEGsignalsTest =
      EEGReader::readData(TEST_SIGNAL_FILE, TEST_STIMULATION_FILE, SAMPLING_FREQ);

//creating the coefficients of a butterworth band pass filter of order 4 in 3-35 Hz
ButterworthFilter butter(bandPass, 3, 35, SAMPLING_FREQ, 4);
vec B = butter.getBCoeff();
vec A = butter.getACoeff();

//creating the coefficients of a surface laplacian spatial filter
```

```
mat laplacian = "4 0 -1 0 -1 -1 0 0 -1 0 ; 0 4 0 -1 0 0 -1 -1 0 -1";

//preprocessing: applying the temporal and spatial filter to the EEG signals
for(unsigned int i=0; i < EEGsignalsTrain.size(); i++)
{
  EEGsignalsTrain[i].applyIIRFilter(A,B); //temporal filter
  EEGsignalsTrain[i].applySpatialFilter(laplacian); //spatial filter
}
for(unsigned int i=0; i < EEGsignalsTest.size(); i++)
{
  EEGsignalsTest[i].applyIIRFilter(A,B); //temporal filter
  EEGsignalsTest[i].applySpatialFilter(laplacian); //spatial filter
}

//defining and extracting band power features
//in the mu and beta bands
BandPower featureExtractor;
featureExtractor.addFreqBand(8,13); //Mu band
featureExtractor.addFreqBand(16,24);  //Beta band

//creation of the training set of FeatureVector for further classification
FeatVecSet trainFeatureSet = featureExtractor.createDataSet(EEGsignalsTrain);

//defining and training an LDA classifier
CLDAClassifier lda;
lda.train(trainFeatureSet);

//testing the resulting BCI
FeatVecSet testFeatureSet = featureExtractor.createDataSet(EEGsignalsTest);
lda.test(testFeatureSet);
cout << ''accuracy on test set: '' << lda.getAccuracy() << ''%'' << endl;
```

This relatively small program makes it possible to design a BCI for offline analysis.

A.4 BLiFF++ dependencies

BLiFF++ is based on several well known and very useful C/C++ libraries which are free and open-source. Indeed, it is based on IT++[1] for dealing with algebra (vector, matrix, matrix decomposition, ...) and for some signal processing tools. BLiFF++ also uses Torch 3[2], which notably provides an interesting implementation of Support Vector Machines. Finally, BLiFF++

[1] http://itpp.sourceforge.net/
[2] http://www.torch.ch/

also uses some functionalities of GSL[3] (GNU Scientific Library) and Boost[4].

A.5 Conclusion

BLiFF++ is a C++ library that enables a fast and easy design of BCI. To achieve this goal, BLiFF++ builds BCI systems by combining several different kinds of signal processing methods, feature extractors and classifiers. It also provides various tools for analyzing brain data, such as statistical analysis tools or clustering algorithms. This results in a flexible platform which can be easily extended and manipulated. BLiFF++ will be soon available for free under the terms of the GPL (GNU Public License) license. Our hope is that this library will prove useful for the BCI community.

[3] http://www.gnu.org/software/gsl/
[4] http://www.boost.org/

Appendix B

Towards a unified approach using implicit surfaces

B.1 Introduction

Implicit surfaces [Blo00, OM95, CG06] are powerful tools designed to manipulate and create complex geometric shapes. Formally, implicit surfaces are surfaces described implicitely by an equation of the following form:

$$F(X_1, X_2, \ldots, X_N) = Iso \tag{B.1}$$

where the X_i are the coordinates of a N-dimensional point, and Iso, a constant value. Different values for Iso would lead to different surfaces.

In this manuscript we have proposed to use inverse solutions and Fuzzy Inference Systems (FIS) for BCI design. It appears that these two methods could be represented using the same formalism, namely, the implicit surface formalism. Indeed, fuzzy membersip functions used in FIS can be exactly seen as implicit surfaces whereas inverse solutions are using brain Region Of Interest (ROI) which regions, being geometric shapes, can be implicitely modelized. In this annex we only present some hints on how performing this unified modeling and further research is needed to achieve this goal. However, we believe that such a modeling could lead to interesting results and may enable us to use new and efficient algorithms from the geometric modeling community in order to design efficient BCI or to design new and insightful brain activity visualization techniques.

B.2 Modeling FuRIA using implicit surfaces

First, let us propose some modeling of FuRIA using implicit surfaces, and let us start this modeling by the FuRIA training phase. The first step of FuRIA training consists in performing a statistical analysis for all voxels and frequencies, in order to identify which ones are the most discriminant. During this first step, each discriminant couple (voxel, frequency) can be modeled using a sphere-like implicit primitive such as a blob or a metaball [OM95]. For the

moment, as no fuzzification has been performed, each one of these primitives have the same weight i.e., the same influence radius. For a metaball, this leads to:

$$f(X) = \begin{cases} d_v(1-3\frac{r}{b}^2) & 0 \leq r \leq \frac{b}{3} \\ \frac{3d}{2}(1-\frac{r}{b})^2 & \frac{b}{3} \leq r \leq b \\ 0 & b \leq r \end{cases} \tag{B.2}$$

where r is the distance between a point X and the center of this metaball, b is the radius of this metaball and d is its weight.

The next step is the clustering step, which aims at gathering voxels and frequencies in clusters. Thus, for each cluster, we would obtain an implicit surface F (for a given level-set value) resulting from the fusion ("blending") of all implicit primitives f corresponding to each couple (voxel, frequency) belonging to this cluster:

$$F(X) = \bigcup f(X) \tag{B.3}$$

It should be noted that this implicit surface lies in a 4-dimensional space (3 dimensions for the spatial coordinates - the voxels - and 1 dimension for the frequency coordinate). We could then project this 4D implicit surface in the spatial domain (3D) in order to obtain an implicit surface delimiting a ROI. We could also project the same 4D implicit surface on the frequency domain (1D) in order to obtain an implicit surface delimiting the frequency band associated to this ROI. These two elements (ROI and frequency band) can be visualized using a rendering of the corresponding implicit surface.

The last step is the fuzzification step, which associates a fuzzy membership function to each ROI and frequency band. These membership functions weigh the contribution of the voxels and frequencies within a ROI or frequency band. As such, this amounts to changing the influence radius (or weight) d of the primitives that compose the implicit surface of ROI and frequency bands. This influence radious could be a function (possibly the identity) of the membership degree of the voxels and frequencies in their ROI and frequency bands respectively. It is still possible to visualize these elements (fuzzy ROI and fuzzy frequency band), by rendering the implicit surface for different values of the level-set. This enables us to focus on the core of a ROI or on all its voxels, for instance.

This modeling using implicit surface could be useful from the point of view of real-time visualization of brain activity and/or for neurofeedback. Indeed, such a modeling can be used to show the user the ROI involved by a given mental task. Moreover, the real-time variation of the activity in these ROI could be represented either by a color change of the implicit surface, or by a change of the level-set value (the higher the activity, the more the voxels displayed to the user), or even both at the same time.

B.3 Modeling FIS with implicit surfaces

Similarly its is possible to model the Chiu's FIS (CFIS) by using implicit surfaces, either during the training of the FIS or during its use for classification. For matter of clarity, let us recall that a CFIS uses fuzzy "if-then" rules of the following form:

$$\text{If } X_1 \text{ is } A_1 \text{ and } X_2 \text{ is } A_2 \text{ and } \ldots \text{and } X_n \text{ is } A_n \text{ then class is } C$$

where the X_i are features and A_i are fuzzy membership functions.

B.3.1 Training

Training a CFIS consists in performing a clustering step on the training feature vectors from each class, and in associating a fuzzy rule to each cluster (i.e., to each class prototype). Each one of these clusters could be represented by an implicit surface, by associating an implicit surface to each feature vector and in merging these implicit surfaces. These different implicit surfaces would then represent each class prototypes. However, these prototypes are not the one used by CFIS for classification. Indeed, for each cluster, a fuzzy membership function is generated for each dimension (i.e., the prototype is projected on each axis).

B.3.2 Classification

It should be noticed that a fuzzy membership function, such as a function A_i, can be seen as a 1D implicit function as it simply associates a value to any feature value. Thus, we can define the function f_R such that:

$$f_R(X) = \mu_R(X) = \prod_{k=1}^{N} A_k(X_k) \tag{B.4}$$

As such, f_R is still an implicit function. Let us consider the following equation:

$$f_R(X) = T \tag{B.5}$$

This equation represents the implicit surface that delineate one of the clusters belonging to class C, and more particularly the cluster corresponding to the fuzzy rule R. The T constant (which defines the isosurface), can here be seen as a rejection threshold [MA06b] (see also Chapter 5). For this threshold value, we can consider that a feature vector X which is outside of this isosurface does not belong to the corresponding cluster. We can also define the following function f_C:

$$f_C(X) = f_{R_1}(X) \cup f_{R_2}(X) \cup \ldots \cup f_{R_m}(X) \tag{B.6}$$

Then, let us consider the following equation:

$$f_C(X) = T \tag{B.7}$$

This equation defines the implicit surface that delineate the whole class C, by considering a rejection threshold T. This is due to the fact that in both the fuzzy set theory and the implicit function theory, union can be achieved using the max operator. Finally, a point X is assigned to the class for which it is "the most" in the associated implicit volume.

However, those definitions cannot be used to deal with multiple rejection thresholds, as did in [MA06b] and in Chapter 5. As such, we associate to each fuzzy rule R the following implicit function:

$$f_R(X) = \mu_R(X) - T_R = \prod_{k=1}^{N} A_k(X_k) - T_R \tag{B.8}$$

where T_R is the rejection threshold of each rule R. We then defined still define $f_C(X)$ as follows:

$$f_C(X) = f_{R_1}(X) \cup f_{R_2}(X) \cup \ldots \cup f_{R_m}(X) \tag{B.9}$$

As such, if $f_C(X) > 0$, then X belongs to class C (it should be noted that, thanks to the fuzzy formalism, X can belong to several classes at the same time), otherwise, it is rejected from this class. The final class C_X of X is then:

$$C_X = argmax_{C \in \{C_1,\ldots,C_{N_c}\}}(f_{C_1}(X), \ldots, f_{C_{N_c}}(X)) \tag{B.10}$$

if this maximum is greater than 0, otherwise X is rejeted.

B.4 Conclusion

This annex has presented some evidences that implicit surfaces have a similar mathematical formulation as brains regions of interest and fuzzy logic. As such we suggested that the FuRIA and FIS algorithms that we proposed might be modeled using a unified formalism, potentially leading to interesting findings. The first steps of this modeling presented here are only hints for starting using implicit surfaces for BCI, and a considerable further work is necessary to really achieve this. However, we believe that these formulations could lead to an interesting and unified theory which could enable us to apply algorithms from the geometric modeling field into BCI and potentially obtain interesting results.

Appendix C

Chapter 2 annex: Complete classification results for the evaluation of FuRIA

In the tables presented hereafter, the column named “NbFeat” displays the number of features extracted using the corresponding hyperparameters.

Table C.1: Data set IV, BCI competition 2003, test set: classification accuracy using FIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	28	80	82	80	84
	0.05	33	86	85	82	84
	0.1	31	84	79	84	80
	0.25	33	85	84	87	85
	0.5	34	83	80	84	80
1	0.01	16	83	82	83	84
	0.05	14	81	82	85	82
	0.1	12	77	80	82	80
	0.25	12	78	78	83	83
	0.5	9	82	76	80	75
1.25	0.01	8	81	83	83	85
	0.05	7	82	81	83	77
	0.1	6	77	81	85	81
	0.25	7	78	76	79	79
	0.5	4	73	74	83	74
1.5	0.01	5	81	81	81	78
	0.05	3	81	80	79	83
	0.1	2	74	74	86	82
	0.25	2	73	74	83	71
	0.5	2	63	75	71	78
1.75	0.01	2	79	80	83	72
	0.05	2	77	77	83	84
	0.1	2	73	77	85	85
	0.25	2	73	76	84	70
	0.5	2	64	73	75	80
2	0.01	2	79	78	82	79
	0.05	2	70	79	84	85
	0.1	2	77	78	86	83
	0.25	2	73	75	83	69
	0.5	2	68	71	75	77
		mean	77.17	78.37	82.1	79.63

Table C.2: Data set IV, BCI competition 2003, test set: classification accuracy using IIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	10	67	67	67	67
	0.05	19	76	77	77	77
	0.1	16	76	77	77	75
	0.25	19	78	78	78	77
	0.5	21	76	73	79	74
1	0.01	8	68	68	67	68
	0.05	13	75	75	74	75
	0.1	12	75	73	78	72
	0.25	14	76	75	78	78
	0.5	12	75	67	73	66
1.25	0.01	9	63	68	68	69
	0.05	10	74	73	74	75
	0.1	10	75	76	75	77
	0.25	11	78	77	80	80
	0.5	8	72	70	82	66
1.5	0.01	6	67	67	67	67
	0.05	8	73	76	76	77
	0.1	7	75	75	78	80
	0.25	7	77	75	79	67
	0.5	6	74	69	77	65
1.75	0.01	5	66	68	67	62
	0.05	7	69	71	78	66
	0.1	5	75	76	80	82
	0.25	5	78	64	77	80
	0.5	5	71	65	80	65
2	0.01	4	70	66	69	66
	0.05	5	71	68	76	69
	0.1	5	76	76	82	83
	0.25	4	72	66	79	78
	0.5	5	57	64	61	57
		mean	72.5	71.33	75.1	72

Table C.3: Data set IIIa, BCI competition 2005, test set, S1: classification accuracy using FIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	40	85.56	85.56	85	86.67
	0.05	31	87.78	83.89	88.89	86.11
	0.1	21	89.44	89.44	90	92.22
	0.25	33	79.44	83.33	83.89	83.33
	0.5	28	80.56	82.22	83.89	82.22
1	0.01	22	87.22	88.33	87.78	88.33
	0.05	18	87.22	88.33	90.56	87.78
	0.1	13	83.33	84.44	90	82.22
	0.25	17	87.22	88.89	87.78	85.56
	0.5	16	82.22	83.89	83.89	80.56
1.25	0.01	8	68.33	68.33	70	68.89
	0.05	10	83.89	82.22	86.11	71.67
	0.1	8	82.22	78.33	76.11	80
	0.25	4	56.11	67.78	77.22	72.78
	0.5	6	60.56	77.78	80.56	69.44
1.5	0.01	4	65.56	66.67	71.11	65
	0.05	4	72.78	73.33	66.67	66.67
	0.1	6	73.89	81.11	78.33	80.56
	0.25	6	56.67	74.44	78.89	69.44
	0.5	3	52.78	53.33	75	71.11
1.75	0.01	2	69.44	65	68.89	62.78
	0.05	4	76.11	73.33	66.11	62.22
	0.1	3	45.56	50	63.33	55.56
	0.25	3	44.44	47.78	61.67	65
	0.5	2	43.33	43.89	57.22	57.22
2	0.01	2	65.56	63.33	65.56	69.44
	0.05	2	56.67	59.44	58.89	53.33
	0.1	2	50	43.89	64.44	62.22
	0.25	2	46.11	52.78	57.22	62.78
	0.5	1	29.44	33.89	38.89	39.44
		mean	68.31	70.5	74.46	72.02

Table C.4: Data set IIIa, BCI competition 2005, test set, S1: classification accuracy using IIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	24	85	86.67	85.56	85
	0.05	26	85.56	85.56	84.44	86.11
	0.1	27	90	93.33	90.56	92.78
	0.25	23	86.67	90.56	87.22	93.33
	0.5	27	83.89	87.22	89.44	89.44
1	0.01	15	85	85	84.44	82.22
	0.05	11	80.56	87.22	85.56	85.56
	0.1	12	86.11	86.11	90	90
	0.25	12	84.44	86.11	83.89	81.67
	0.5	11	85	83.89	88.89	90
1.25	0.01	7	83.89	87.78	83.89	89.44
	0.05	8	85	83.89	85.56	86.11
	0.1	9	81.11	77.78	78.89	82.22
	0.25	5	68.33	80	75.56	62.78
	0.5	7	68.89	67.22	71.67	67.78
1.5	0.01	4	69.44	71.11	71.11	67.22
	0.05	4	78.89	82.22	81.67	86.11
	0.1	8	81.67	81.67	84.44	70
	0.25	4	57.22	65	68.89	70
	0.5	5	67.22	68.33	74.44	70
1.75	0.01	3	71.67	71.11	63.89	62.78
	0.05	3	70.56	73.89	65.56	66.11
	0.1	5	67.22	68.33	70.56	70.56
	0.25	2	38.33	43.89	66.11	56.11
	0.5	3	45	54.44	74.44	62.22
2	0.01	2	56.67	54.44	56.67	52.22
	0.05	4	67.22	65	57.78	60.56
	0.1	3	49.44	56.11	60.56	61.11
	0.25	1	30.56	42.22	52.22	50
	0.5	1	22.78	39.44	52.22	48.89
		mean	70.44	73.52	75.54	73.94

Table C.5: Data set IIIa, BCI competition 2005, test set, S2: classification accuracy using FIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	12	37.5	36.67	40.83	38.33
	0.05	16	55.83	57.5	52.5	56.67
	0.1	23	56.67	61.67	58.33	64.17
	0.25	28	56.67	60	61.67	63.33
	0.5	30	57.5	60	66.67	66.67
1	0.01	7	32.5	36.67	29.17	30.83
	0.05	13	57.5	58.33	55.83	55.83
	0.1	11	52.5	55	55.83	57.5
	0.25	10	59.17	55	60.83	55.83
	0.5	11	46.67	44.17	50	55.83
1.25	0.01	4	39.17	36.67	40	38.33
	0.05	8	56.67	55	55	54.17
	0.1	8	48.33	48.33	49.17	45.83
	0.25	8	50.83	53.33	59.17	53.33
	0.5	9	52.5	52.5	55.83	55
1.5	0.01	4	31.67	44.17	35.83	35.83
	0.05	6	52.5	51.67	49.17	51.67
	0.1	6	42.5	46.67	58.33	53.33
	0.25	5	50.83	47.5	53.33	55.83
	0.5	6	42.5	36.67	55	51.67
1.75	0.01	3	37.5	38.33	40.83	31.67
	0.05	6	55	55.83	52.5	54.17
	0.1	5	48.33	47.5	57.5	50
	0.25	3	47.5	42.5	41.67	40.83
	0.5	3	50.83	40.83	47.5	43.33
2	0.01	3	36.67	28.33	38.33	38.33
	0.05	5	35.83	37.5	43.33	40
	0.1	2	37.5	41.67	47.5	40.83
	0.25	2	43.33	40	42.5	36.67
	0.5	2	45	40.83	42.5	42.5
		mean	47.25	49.89	48.61	48.19

Table C.6: Data set IIIa, BCI competition 2005, test set, S2: classification accuracy using IIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.05	17	65	56.67	60.83	54.17
	0.1	29	60.83	63.33	60	65
	0.25	32	67.5	61.67	60	61.67
	0.5	40	62.5	62.5	65.83	58.33
1	0.05	12	63.33	40	63.33	57.5
	0.1	14	66.67	56.67	65	60
	0.25	17	54.17	51.67	54.17	53.33
	0.5	22	51.67	57.5	50.83	54.17
1.25	0.05	7	57.5	50	61.67	57.5
	0.1	13	57.5	54.17	57.5	53.33
	0.25	10	53.33	33.33	44.17	39.17
	0.5	13	53.33	45	52.5	55.83
1.5	0.05	4	52.5	43.33	48.33	47.5
	0.1	7	50.83	49.17	54.17	60
	0.25	7	45	44.17	52.5	39.17
	0.5	8	49.17	40.83	53.33	47.5
1.75	0.05	4	45.83	38.33	58.33	56.67
	0.1	2	44.17	43.33	53.33	42.5
	0.25	3	48.33	47.5	42.5	49.17
	0.5	4	47.5	40	53.33	56.67
2	0.05	3	43.33	37.5	56.67	58.33
	0.1	2	40	39.17	56.67	55.83
	0.25	1	37.5	25	30	31.67
	0.5	1	40.83	41.67	28.33	35.83
		mean	52.43	46.77	53.47	52.11

Table C.7: Data set IIIa, BCI competition 2005, test set, S3: classification accuracy using FIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	16	80	79.17	84.17	77.5
	0.05	26	80	75.83	84.17	77.5
	0.1	26	85	83.33	85.83	83.33
	0.25	29	84.17	80.83	80.83	83.33
	0.5	33	79.17	81.67	82.5	83.33
1	0.01	8	83.33	83.33	84.17	79.17
	0.05	11	77.5	81.67	75.83	79.17
	0.1	13	75.83	71.67	84.17	84.17
	0.25	16	76.67	81.67	77.5	76.67
	0.5	19	78.33	81.67	78.33	81.67
1.25	0.01	6	79.17	82.5	80	80
	0.05	7	75.83	80.83	78.33	81.67
	0.1	5	76.67	75	84.17	82.5
	0.25	10	70.83	79.17	70	77.5
	0.5	10	77.5	75	78.33	75.83
1.5	0.01	3	77.5	80	80	82.5
	0.05	6	79.17	78.33	81.67	81.67
	0.1	5	75	73.33	80.83	79.17
	0.25	7	68.33	80.83	74.17	77.5
	0.5	7	71.67	80	72.5	78.33
1.75	0.01	2	55	51.67	50	59.17
	0.05	5	64.17	70.83	60	68.33
	0.1	3	57.5	70.83	46.67	55
	0.25	6	55.83	72.5	55	58.33
	0.5	4	45	53.33	47.5	50.83
2	0.01	2	55	41.67	53.33	34.17
	0.05	4	61.67	74.17	42.5	78.33
	0.1	3	52.5	58.33	55.83	50.83
	0.25	3	42.5	55	50	39.17
	0.5	2	33.33	56.67	51.67	32.5
		mean	69.14	73.02	70.33	70.97

Table C.8: Data set IIIa, BCI competition 2005, test set, S3: classification accuracy using IIR filters (%)

H	α	NbFeat	Raw	Freq	Space	All
0.75	0.01	16	79.17	74.17	84.17	77.5
	0.05	28	75	76.67	74.17	74.17
	0.1	30	80	72.5	79.17	72.5
	0.25	32	79.17	78.33	80.83	80
	0.5	37	76.67	71.67	79.17	70
1	0.01	11	81.67	72.5	79.17	83.33
	0.05	16	72.5	75.83	77.5	74.17
	0.1	18	75	70.83	71.67	71.67
	0.25	13	59.17	63.33	68.33	72.5
	0.5	16	53.33	60	61.67	74.17
1.25	0.01	6	75.83	81.67	75.83	81.67
	0.05	8	65.83	76.67	73.33	77.5
	0.1	12	65.83	75.83	75	80.83
	0.25	10	52.5	63.33	70	70.83
	0.5	8	40.83	65	55.83	67.5
1.5	0.01	4	76.67	82.5	76.67	80
	0.05	8	65	75.83	65.83	76.67
	0.1	8	60	67.5	65	76.67
	0.25	7	57.5	60	55	60.83
	0.5	9	46.67	68.33	69.17	72.5
1.75	0.01	3	70	77.5	70	78.33
	0.05	6	51.67	65	37.5	54.17
	0.1	8	60.83	66.67	50.83	54.17
	0.25	5	43.33	47.5	30.83	50
	0.5	6	44.17	54.17	42.5	48.33
2	0.01	2	50	55.83	59.17	56.67
	0.05	5	47.5	63.33	30	63.33
	0.1	5	35.83	63.33	33.33	50.83
	0.25	3	36.67	36.67	26.67	47.5
	0.5	2	33.33	48.33	25.83	32.5
		mean	60.39	67.03	61.47	67.69

Appendix D

Chapter 5 annex: Detailed classification and rejection results for each data set

Table D.1: Data set of subject 1, Day 1: average Accuracy (Acc), TAR and Precison (Prec), in percent, for a fixed FAR of 10%.

		SVM	FIS	RBFN	LDA
SC	Acc	63.82	72.52	64.99	69.26
	TAR	58.34	54.35	56.54	56.15
	Prec	79.01	77.81	78.48	78.37
RC	Acc	73.36	72.14	70.95	71.84
	TAR	56.5	**59.51**	52.04	51.68
	Prec	78.47	**79.37**	77.05	76.93
AMTL1 ST	Acc	75.64	82.03	69.29	87.32
	TAR	12.5	29.96	28.02	5.07
	Prec	44.64	65.90	64.38	24.67
AMTL1 MT	Acc	75.64	70.71	72.78	87.31
	TAR	12.5	43.36	30.08	20.72
	Prec	44.64	73.67	65.99	57.21
AMTL2 MT	Acc	75.41	82.5	66.22	**89.11**
	TAR	12.54	30.8	34.33	5.6
	Prec	44.72	66.52	68.89	26.54

Table D.2: Data set of subject 1, Day 2: average Accuracy (Acc), TAR and Precison (Prec), in percent, for a fixed FAR of 10%.

		SVM	FIS	RBFN	LDA
SC	Acc	73.87	75.22	77.20	76.19
	TAR	40.89	40.96	39.9	37.81
	Prec	72.51	72.55	72.02	70.92
RC	Acc	81.97	77.71	77.56	79.66
	TAR	**41.82**	40.47	40.62	40.3
	Prec	**72.96**	72.31	72.38	72.22
AMTL1 ST	Acc	86.68	96.48	87.92	**98.95**
	TAR	21.16	20.6	22.14	18.69
	Prec	57.72	57.06	58.82	54.67
AMTL1 MT	Acc	87.59	71.43	88.51	**98.95**
	TAR	20.59	30.42	24.82	21.45
	Prec	57.05	66.25	61.56	58.05
AMTL2 MT	Acc	86.38	94.28	77.52	98.72
	TAR	21.06	23.5	26.73	11.76
	Prec	57.60	60.26	63.30	43.14

Table D.3: Data set of subject 2, Day 1: average Accuracy (Acc), TAR and Precison (Prec), in percent, for a fixed FAR of 10%.

		SVM	FIS	RBFN	LDA
SC	Acc	83.38	78.78	84.94	75.88
	TAR	29.24	24.28	29.21	20.79
	Prec	64.63	60.28	64.61	56.51
RC	Acc	91.36	89.5	90.83	89.01
	TAR	28.94	28.43	**29.54**	27.09
	Prec	64.40	63.99	**64.87**	62.87
AMTL1 ST	Acc	88.26	**97.05**	88.67	97.04
	TAR	15.08	21.51	16.93	26.01
	Prec	48.52	57.34	51.41	61.91
AMTL1 MT	Acc	88.33	87.65	90.79	92.86
	TAR	14.84	19.07	16.31	16.7
	Prec	48.12	54.37	50.48	51.07
AMTL2 MT	Acc	88.33	96.32	79.38	94.02
	TAR	14.84	22.21	15	18.81
	Prec	48.12	58.13	48.39	54.04

Table D.4: Data set of subject 2, Day 2: average Accuracy (Acc), TAR and Precison (Prec), in percent, for a fixed FAR of 10%.

		SVM	FIS	RBFN	LDA
SC	Acc	75.46	66.38	68.51	66.81
	TAR	24.13	17.45	22.16	17.67
	Prec	60.28	52.32	58.22	52.64
RC	Acc	86.73	78.43	81.36	83.95
	TAR	**32.78**	26.53	30.47	25.51
	Prec	**67.34**	62.53	65.71	61.60
AMTL1 ST	Acc	85.90	94.81	84.92	94.72
	TAR	16.64	19.05	13.29	18.92
	Prec	51.14	54.51	45.53	54.34
AMTL1 MT	Acc	84.63	80.65	82.06	93.51
	TAR	17.02	21.17	17.53	18.43
	Prec	51.70	57.11	52.44	53.68
AMTL2 MT	Acc	85.23	**95.25**	80.63	94.38
	TAR	16.52	19.83	13.87	18.4
	Prec	50.96	55.5	46.59	53.64

Appendix E

Chapter 6 annex: Excerpt of the questionnaire filled by subjects

Questionnaire

Age:
Gender:
Nationality:
Known perception problem (glasses, color blindness …):
Left/Right handed:

Please answer the following questions (please tick only one cell per question):

	1	2	3	4	5	6	7
Do you define yourself as attracted by technology? (1: no, not at all, 4: moderately, 7: yes, completely)							
Have you ever heard of brain-computer interfaces before? (1: never, 7: yes, I know the subject pretty well)							

Please answer the following questions by writing a mark between 1 and 7 in the appropriate cell:

	Real movements	Imagined movements
Did you get tired because of the experiment? (1: not tired at all, 7: very tired)		
Did you find the experiment comfortable? (1: not comfortable at all, 7: very comfortable)		
Did you feel that you could control the spaceship (that is that you could lift it voluntarily)? (1: you didn't feel you could control it at all, 7: you controlled it perfectly)		
Did you feel frustration or annoyance during the experiment? (1: neither frustration nor annoyance 7: a lot of both frustration and annoyance)		

Figure E.1: Excerpt of the first page of the questionnaire filled by subjects at the end of the "Use the force!" experiment (Chapitre 6).

Please answer the following questions freely:

What strategies did you employ in order to control the spaceship during the experiment based on imagined movements? What kind of foot movements did you imagine?

_____
_____
_____
_____
_____
_____.

What did you think of the interface and the software you have just used?

_____
_____
_____
_____
_____
_____.

Do you have any other comments or remarks concerning this experiment?

_____
_____
_____
_____
_____
_____
_____.

Figure E.2: Excerpt of the second page of the questionnaire filled by subjects at the end of the "Use the force!" experiment (Chapitre 6).

Appendix F

Chapter 6 annex: Detailed information extracted from the questionnaires

Table F.1: General information about the 21 subjects who participated in the experiment described in Chapter 6.

Subject	Age	Gender	Nationality	Preception problem	Hand
1	26	M	Czech	glasses	right
2	23	M	French		right
3	44	M	French		right
4	26	M	French		right
5	38	M	French		left
6	42	M	French		right
7	30	M	French	glasses	right
8	39	M	French	contact lens	right
9	27	M	French		left
10	29	M	French		right
11	29	M	French	glasses	right
12	27	F	French		right
13	39	F	French		right
14	31	M	French		right
15	43	M	French		right
16	34	M	French		right
17	24	M	French	glasses	not specified
18	60	M	French	not specified	not specified
19	27	M	French	glasses + lens	right
20	40	M	French	weak myopia	right
21	25	F	French	weak hypermetropia	both

Table F.2: Feelings of subjects as measured by the questionnaire. R stands for Real movement experiment and I stands for imagined movement experiment.

	Q1: Tireness		**Q2: Comfort**		**Q3: Control**		**Q4: Frustration**	
Subject	**R**	**I**	**R**	**I**	**R**	**I**	**R**	**I**
1	2	1	5	5	3	1	2	3
2	1	1	4	4	5	4	2	4
3	2	2	5	5	5	5	1	1
4	1	2	6	6	6	2	1	4
5	2	1	4	4	1	1	3	3
6	1	1	6	6	5	2	1	2
7	3	4	3	3	4	5	5	5
8	1	2	3	3	2	1	1	1
9	5	5	5	5	4	1	1	2
10	1	1	5	6	5	3	3	5
11	1	1	6	6	1	1	1	1
12	2	3	3	3	5	3	3	5
13	1	1	6	6	5	3	2	6
14	2	1	6	6	4	3	2	3
15	1	1	7	7	7	7	1	1
16	2	2	6	6	2	2	3	5
17	3	5	4	6	5	6	6	4
18	1	1	6	5	2	2	1	1
19	3	4	6	6	5	5	5	5
20	1	1	4	6	1	1	4	4
21	1	3	7	7	6	1	1	4

Table F.3: Performances of subjects (number of FP and TP, and HF difference) during the real and imagined movement experiments.

	Real			**Imagined**		
Subject	**TP**	**FP**	**HF diff**	**TP**	**FP**	**HF diff**
1	3	1	2	3	1	2
2	8	5	3	7	5	2
3	6	1	5	6	0	6
4	7	0	7	2	0	2
5	3	1	2	1	0	1
6	8	3	5	2	0	2
7	9	6	3	7	6	1
8	5	4	1	2	3	-1
9	5	0	5	2	1	1
10	5	0	5	2	1	1
11	2	2	0	3	1	2
12	4	2	2	1	3	-2
13	4	2	2	4	1	3
14	4	1	3	4	0	4
15	3	0	3	2	0	2
16	7	1	6	0	2	-2
17	6	1	5	4	1	3
18	6	8	-2	1	1	0
19	2	0	2	3	0	3
20	1	0	1	0	2	-2
21	6	0	6	0	0	0

Author's references

[CLL06] M. Congedo, F. Lotte, and A. Lécuyer. Classification of movement intention by spatially filtered electromagnetic inverse solutions. *Physics in Medicine and Biology*, 51(8):1971–1989, 2006.

[LCL+07] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4:R1–R13, 2007.

[LLA07a] F. Lotte, A. Lécuyer, and B. Arnaldi. FuRIA : un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur utilisant modèles inverses et modèles flous. In *Colloque GRETSI*, pages 665–668, 2007.

[LLA07b] F. Lotte, A. Lécuyer, and B. Arnaldi. FuRIA: A novel feature extraction algorithm for brain-computer interfaces using inverse models and fuzzy region of interest. In *Proc. of the 3rd IEEE-EMBS international Conference on Neural Engineering*, pages 175–178, 2007.

[LLA07c] F. Lotte, A. Lécuyer, and B. Arnaldi. Les interfaces cerveau-ordinateur : Utilisation en robotique et avancées récentes. In *Journées Nationales de la Recherche en Robotique*, 2007.

[LLA08] F. Lotte, A. Lécuyer, and B. Arnaldi. FuRIA : un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur utilisant modèles inverses et modèles flous. *Traitement du signal*, 2008.

[LLA09] F. Lotte, A. Lécuyer, and B. Arnaldi. FuRIA: An inverse solution based feature extraction algorithm using fuzzy set theory for brain-computer interfaces. *IEEE transactions on signal processing - accepted with mandatory minor revisions*, 2009.

[LLLA07] F. Lotte, A. Lécuyer, F. Lamarche, and B. Arnaldi. Studying the use of fuzzy inference systems for motor imagery classification. *IEEE Transactions on Neural System and Rehabilitation Engineering*, 15(2):322–324, 2007.

[LLR+06] F. Lotte, A. Lécuyer, Y. Renard, F. Lamarche, and B. Arnaldi. Classification de données cérébrales par système d'inférence flou pour l'utilisation d'interfaces

cerveau-ordinateur en réalité virtuelle. In *Actes des Premières Journées de l'Association Française de Réalité Virtuelle, AFRV06*, pages 55–62, 2006.

[LLR+08] A. Lécuyer, F. Lotte, R.B. Reilly, R. Leeb, M. Hirose, and M. Slater. Brain-computer interfaces, virtual reality and videogames. *IEEE Computer*, 41(10):66–72, 2008.

[LML08] F. Lotte, H. Mouchère, and A. Lécuyer. Pattern rejection strategies for the design of self-paced EEG-based brain-computer interfaces. In *International Conference on Pattern Recognition (ICPR)*, 2008.

[Lot06] F. Lotte. The use of fuzzy inference systems for classification in EEG-based brain-computer interfaces. In *3rd international Brain-Computer Interface Workshop and Training Course*, pages 12–13, 2006.

[LRL08] F. Lotte, Y. Renard, and A. Lécuyer. Self-paced brain-computer interaction with virtual worlds: a qualitative and quantitative study 'out-of-the-lab'. In *4th International Brain-Computer Interface Workshop and Training Course*, pages 373–378, 2008.

[RGC+07] Y. Renard, G. Gibert, M. Congedo, F. Lotte, E. Maby, B. Hennion, O. Bertrand, and A. Lécuyer. OpenViBE: An open-source software platform to easily design, test and use brain-computer interfaces. In *BCI Meets Robotics: Challenging Issues in Brain-Computer Interaction and Shared Control, MAIA workshop*, page 49, 2007.

[SLLC09] J.B. Sauvan, A. Lécuyer, F. Lotte, and G. Casiez. A performance model of selection techniques for p300-based brain-computer interfaces. In *ACM SIGCHI Conference on Human Factors in Computing Systems (ACM CHI) (note) - accepted*, 2009.

[ZLGL08] M. Zhong, F. Lotte, M. Girolami, and A. Lécuyer. Classifying EEG for brain computer interfaces using gaussian processes. *Pattern Recognition Letters*, 29:354–359, 2008.

References

[ABC+06] B. Arslan, A. Brouse, J. Castet, R. Lehembre, C. Simon, J.J. Filatriau, and Q. Noirhomme. A real time music synthesis environment driven with biological signals. In *Proceedings fo the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 14–19, 2006.

[ACM+05] C. Arrouët, M. Congedo, J. E. Marvie, F. Lamarche, A. Lecuyer, and B. Arnaldi. Open-ViBE : a 3D platform for real-time neuroscience. *Journal of Neurotherapy*, 9(1):3–25, 2005.

[AES91] American electroencephalographic society: Guidelines for standard electrode position nomenclature. *J Clin Neurophysiol.*, 8(2):200–202, 1991.

[AGG07] B. Allison, B. Graimann, and A. Gräser. Why use a BCI if you are healthy ? In *proceedings of BRAINPLAY 2007, playing with your brain*, 2007.

[AL96b] E. Anquetil and G. Lorette. On-line handwriting character recognition system based on hierarchical qualitative fuzzy modeling. In *Proceedings of the 5th International Workshop on Frontiers in Handwriting Recognition (IWFHR5)*, pages 47–52, 1996.

[AS96] C. W. Anderson and Z. Sijercic. Classication of EEG signals from four subjects during five mental tasks. In *Solving Engineering Problems with Neural Networks: Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'96)*, 1996.

[ASS98] C. W. Anderson, E. A. Stolz, and S. Shamsunder. Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks. *IEEE Transactions on Biomedical Engineering*, 45:277–286, 1998.

[AWW07] B.Z. Allison, E.W. Wolpaw, and J.R. Wolpaw. Brain-computer interface systems: progress and prospects. *Expert Review of Medical Devices*, 4(4):463–474, 2007.

[Bay03] J. D. Bayliss. The use of the P3 evoked potential component for control in a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):113–116, 2003.

[BB00] J. D. Bayliss and D. H. Ballard. A virtual reality testbed for brain-computer interface research. *IEEE Transactions on Rehabilitation Engineering*, 8(2), 2000.

[BB04] G. Blanchard and B. Blankertz. BCI competition 2003–data set IIa: spatial patterns of self-controlled brain rhythm modulations. *IEEE Transactions on Biomedical Engineering*, 51(6):1062–1066, 2004.

[BBS+01] F. Babiloni, L. Bianchi, F. Semeraro, J. del R Millan, J. Mourino, A. Cattini, S. Salinari, M.G. Marciani, and F. Cincotti. Mahalanobis distance-based classifiers are able to recognize EEG patterns by using few EEG electrodes,. In *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 651–654, 2001.

[BC00] K. P. Bennett and C. Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.

[BCM02] B. Blankertz, G. Curio, and K. R. Müller. Classifying single trial EEG: Towards brain computer interfacing. *Advances in Neural Information Processing Systems (NIPS 01)*, 14:157–164, 2002.

[BCM+07] F. Babiloni, F. Cincotti, M. Marciani, S. Salinari, L. Astolfi, A. Tocci, F. Aloise, F. De Vico Fallani, S. Bufalari, and D. Mattia. The estimation of cortical activity for brain-computer interface: Applications in a domotic context. *Computational Intelligence and Neuroscience*, 2007, 2007.

[BDK+06a] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, V. Kunzmann, F. Losch, and G. Curio. The Berlin brain-computer interface: EEG-based communication without subject training. *IEEE Trans. Neural Sys. Rehab. Eng.*, 14(2):147–152, 2006.

[BDK+06b] B. Blankertz, G. Dornhege, M. Krauledat, M. Schröder, J. Williamson, R. Murray-Smith, and K.-R. Müller. The Berlin brain-computer interface presents the novel mental typewriter hex-o-spell. In *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course*, pages 108–109, 2006.

[BDK+07] B. Blankertz, G. Dornhege, M. Krauledat, G. Curio, and K.-R. Müller. The non-invasive Berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage*, 37(2):539–550, 2007.

[Ber29] H. Berger. Ueber das elektroenkephalogramm des menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, 87:527–570, 1929.

[BFdM04] G. A. Barreto, R. A. Frota, and F. N. S. de Medeiros. On the classification of mental tasks: a performance comparison of neural and statistical approaches. In *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*, 2004.

[BFWB07] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural engineering*, 4(2):R35–57, 2007.

[BGM07a] M. Besserve, L. Garnero, and J. Martinerie. Cross-spectral discriminant analysis (CSDA) for the classification of brain computer interfaces. In *3rd International IEEE/EMBS Conference on Neural Engineering, 2007. CNE '07*, pages 375–378, 2007.

[BGM07b] M. Besserve, L. Garnero, and J. Martinerie. De l'estimation à la classification des activités corticales pour les interfaces cerveau-machine. In *Proc. GRETSI*, 2007.

[BGMP07] R. Boostani, B. Graimann, M.H. Moradi, and G. Pfurtscheller. A comparison approach toward finding the best feature and classifier in cue-based BCI. *Medical and Biological Engineering and Computing*, 45(4):403–412, 2007.

[Bir06] N. Birbaumer. Breaking the silence: Brain-computer interfaces (BCI) for communication and motor control. *Psychophysiology*, 43(6):517–532, 2006.

[Bis96] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[BJL+08] M. Besserve, K. Jerbi, F. Laurent, S. Baillet, J. Martinerie, and L. Garnero. Classification methods for ongoing EEG and MEG signals. *Biol. Res.*, 40(4):415–437, 2008.

[BKd+05] D. P. Burke, S. P. Kelly, P. deChazal, R. B. Reilly, and C. Finucane. A parametric feature extraction and classification strategy for brain-computer interfacing. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(1), 2005.

[BKD+07b] B. Blankertz, M. Krauledat, G. Dornhege, J. Williamson, R. Murray-Smith, and K.-R. Müller. A note on brain actuated spelling with the Berlin brain-computer interface. In *Universal Access in HCI, Part II, HCII 2007, volume 4555 of LNCS*, pages 759–768, 2007.

[BKG+00] N. Birbaumer, A. Kübler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, and H. Flor. The thought translation device (TTD) for completely paralyzed patients. *IEEE Transactions on Rehabilitation Engineering*, 8:190–193, 2000.

[BKJP05] D.A. Bowman, E. Kruijff, J.J. LaViola Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley/Pearson Education, 2005.

[BKM+07b] B. Blankertz, M. Krauledat, K.-R. Müller, G. Dornhege, V. Kunzmann, F. Losch, and G. Curio. *Towards brain-computer interfacing*, chapter The Berlin brain-computer interface: Machine learning-based detection of user

specific brain states. MIT Press, G. Dornhege, R. Millan Jdel, T. Hinterberger, D. J. McFarland & K. R. Müller edition, 2007.

[BKT+08] B. Blankertz, M. Kawanabe, R. Tomioka, F. Hohlefeld, V. Nikulin, and K.-R. Müller. Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. In *Advances in Neural Information Processing Systems 20, In . MIT Press, Cambridge, MA*, 2008.

[BLH93] G.E. Birch, P.D. Lawrence, and R.D Hare. Single-trial processing of event-related potentials using outlier information. *IEEE Transactions on Biomedical Engineering*, 40(1):59–73, 1993.

[Blo00] J. Bloomenthal. *Encyclopedia of Computer Science and Technology*, chapter Implicit Surfaces. Marcel Dekker Inc, ny edition, 2000.

[BM04] R. Boostani and M. H. Moradi. A new approach in the BCI research based on fractal dimension as feature and adaboost as classifier. *Journal of Neural Engineering*, 1(4):212–217, 2004.

[BMB06] J.F. Borisoff, S.G. Mason, and G.E. Birch. Brain interface research for asynchronous control applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):160–164, 2006.

[BMBB04] J. F. Borisoff, S. G. Mason, A. Bashashati, and G. E. Birch. Brain-computer interface design for asynchronous control applications: Improvements to the LF-ASD asynchronous brain switch. *IEEE Transactions on Biomedical Engineering*, 51(6):985– 992, 2004.

[BMC+04] B. Blankertz, K. R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, 51(6):1044–1051, 2004.

[BMG08] M. Besserve, J. Martinerie, and L. Garnero. Non-invasive classification of cortical activities for brain-computer interfaces: a variable selection approach. In *IEEE International Symposium on Biomedical Imaging (ISBI' 2008)*, pages 1063–1066, 2008.

[BMK+06] B. Blankertz, K. R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. D. R. Millan, M. Schroder, and N. Birbaumer. The BCI competition III: Validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):153–159, 2006.

[BML01] S. Baillet, J.C. Mosher, and R.M. Leahy. Electromagnetic brain mapping. *IEEE Signal Processing Magazine*, 18(6):14–30, 2001.

[Bos04] V. Bostanov. BCI competition 2003–data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *IEEE Transactions on Biomedical Engineering*, 51(6):1057–1061, 2004.

[BP92] J. C. Bezdec and S. K. Pal. *Fuzzy Models For Pattern Recognition*. IEEE PRESS, 1992.

[BP05] D. Balakrishnan and S. Puthusserypady. Multilayer perceptrons for the classification of brain computer interface data. In *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference*, 2005.

[Bre98] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[Bro08] N. Brodu. Multifractal feature vectors for brain-computer interfaces. In *International Joint Conference on Neural Networks, IJCNN*, 2008.

[BSCR07] C.J. Bell, P. Shenoy, R. Chalodhorn, and R. Rao. An image-based brain-computer interface using the P3 response. In *Proceedings of the 3rd IEEE EMBS International Conference on Neural Engineering*, pages 318–321, 2007.

[BSCR08] C.J. Bell, P. Shenoy, R. Chalodhorn, and R.P.N. Rao. Control of a humanoid robot by a noninvasive brain-computer interface in humans. *J. Neural Eng.*, 5:214–220, 2008.

[BTV96] A. B. Barreto, A. M. Taberner, and L. M. Vicente. Classification of spatio-temporal EEG readiness potentials towards the development of a brain-computer interface. In *Southeastcon '96. 'Bringing Together Education, Science and Technology', Proceedings of the IEEE*, 1996.

[BU03] O. F. Bay and A. B. Usakli. Survey of fuzzy logic applications in brain-related researches. *Journal of Medical Systems*, 2003.

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2:121–167, 1998.

[BWB07] A. Bashashati, R.K. Ward, and G.E. Birch. Towards development of a 3-state self-paced brain-computer interface. *Computational Intelligence and Neuroscience*, 2007.

[CAB+06] F. Cincotti, F. Aloise, F. Babiloni, M.G. Marciani, D. Morelli, S. Paolucci, G. Oriolo, A. Cherubini, S. Bruscino, F. Sciarra, F. Mangiola, A. Melpignano, F. Davide, and D. Mattia. Brain-operated assistive devices: the ASPICE project. In *BioRob 2006. The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 817– 822, 2006.

[CAR+07] A. Chatterjee, V. Aggarwal, A. Ramos, S. Acharya, and N.V. Thakor. A brain-computer interface with vibrotactile biofeedback for haptic information. *Journal of NeuroEngineering and Rehabilitation*, 4(40), 2007.

[CB04] S. Chiappa and S. Bengio. HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. In *European Symposium on Artificial Neural Networks ESANN*, 2004.

[CBM02] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. *IDIAP-RR 02-46*, 2002.

[CG06] M.-P. Cani and E. Galin. *Informatique graphique et géométrique, animation: chapitre Surfaces Implicites*, chapter 5. Hermès, editors: D. Bechmann and B. Péroche edition, 2006.

[CGGX02] M. Cheng, X. Gao, S. Gao, and D. Xu. Design and implementation of a brain-computer interface with high transfer rates. *IEEE Transactions on Biomedical Engineering*, 49(10):1181–1186, 2002.

[CGN+92] G.A. Carpenter, S. Grossberg, N.Markuzon, J.H.Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3:698–713, 1992.

[Chi97a] S. L. Chiu. An efficient method for extracting fuzzy classification rules from high dimensional data. *Journal of Advanced Computational Intelligence*, 1:31–36, 1997.

[Chi97b] S. L. Chiu. Extracting fuzzy rules from data for function approximation and pattern classification. *Chapter 9 in Fuzzy Information Engineering: A Guided Tour of Applications, ed. D. Dubois, H. Prade, and R. Yager, John Wiley and Sons*, 1997.

[CJ00] E.J.X. Costa and E.F. Cabral Jr. EEG-based discrimination between imagination of left and right hand movements using adaptive gaussian representation. *Medical Engineering & Physics*, 22(5):345–348, 2000.

[CKA+07] F. Cincotti, L. Kauhanen, F. Aloise, T. Palomäki, N. Caporusso, P. Jylänki, D. Mattia, F. Babiloni, G. Vanacker, M. Nuttin, M.G. Marciani, and J. del R. Millán. Vibrotactile feedback for brain-computer interface operation. *Computational Intelligence and Neuroscience*, 2007, 2007.

[CLL06] M. Congedo, F. Lotte, and A. Lécuyer. Classification of movement intention by spatially filtered electromagnetic inverse solutions. *Physics in Medicine and Biology*, 51(8):1971–1989, 2006.

[CM02] D. Comaniciu and P. Meer. Mean shift - a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[COM+07] A. Cherubini, G. Oriolo, F. Macri, A. Cherubini, G. Oriolo, and F. Macri. Development of a multimode navigation system for an assistive robotics project. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2336–2342, 2007.

[Con06] M. Congedo. Subspace projection filters for real-time brain electromagnetic imaging. *IEEE Transactions on Biomedical Engineering*, 53(8):1624–1634, 2006.

[CPM05] D. Coyle, G. Prasad, and T. M. McGinnity. A time-frequency approach to feature extraction for a brain-computer interface with a comparative analysis of performance measures. *EURASIP J. Appl. Signal Process.*, 2005(1):3141–3151, 2005.

[CR07] F. Cabestaing and A. Rakotomamonjy. Introduction aux interfaces cerveau-machine (BCI). In *21ème Colloque sur le Traitement du Signal et des Images, GRETSI'07*, pages 617–620, 2007.

[CS03] E. A. Curran and M. J. B. Stokes. Learning to control brain activity: a review of the production and control of EEG components for driving brain-computer interface (BCI) systems. *Brain and Cognition*, pages 326–336, 2003.

[CST+03] F. Cincotti, A. Scipione, A. Tiniperi, D. Mattia, M.G. Marciani, J. del R. Millán, S. Salinari, L. Bianchi, and F. Babiloni. Comparison of different feature classifiers for brain computer interfaces. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, 2003.

[CWM07] S.M. Coyle, T.E. Ward, and C.M. Markham. Brain-computer interface using a simplified functional near-infrared spectroscopy system. *Journal of neural engineering*, 4:219–226, 2007.

[CYL+00] F. H. Y. Chan, Y. S. Yang, F. K. Lam, Y. T. Zhang, and P. A. Parker. Fuzzy EMG classification for prosthesis control. *IEEE transactions on rehabilitation engineering*, 8(3):305–311, 2000.

[DBCM04a] G. Dornhege, B. Blankertz, G. Curio, and K.R. Müller. Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multi-class paradigms. *IEEE Transactions on Biomedical Engineering*, 51(6):993–1002, 2004.

[DBCM04b] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. Increase information transfer rates in BCI by CSP extension to multi-class. In *Advances in Neural Information Processing Systems*, pages 733–740, 2004.

[DBK+06] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K.-R. Müller. Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Trans. Biomed. Eng.*, 53(11):2274–2281, 2006.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Recognition, second edition*. WILEY-INTERSCIENCE, 2001.

[dM03] J. W. de Moor. Building a brain interface. *www-vf.bio.uu.nl/LAB/NE/scripties/Building_a_Brain_Interface.pdf*, 2003.

[DM04] A. Delorme and S. Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods*, 134:9–21, 2004.

[dRMFB07] J. del R. Millán, P.W. Ferrez, and A. Buttfield. *Towards brain-computer interfacing*, chapter The IDIAP brain-computer interface: An Asynchronous Multiclass Approach. MIT Press, G. Dornhege, R. Millan Jdel, T. Hinterberger, D. J. McFarland & K. R. Müller edition, 2007.

[dRMFM$^+$02] J. del R. Millán, M. Franzé, J. Mouriño, F. Cincotti, and F. Babiloni. Relevant EEG features for the classification of spontaneous motor-related tasks. *Biological Cybernetics*, 86(2):89–95, 2002.

[dRMMC$^+$00] J. del R. Millán, J. Mouriño, F. Cincotti, F. Babiloni, M. Varsta, and J. Heikkonen. Local neural classifier for EEG-based recognition of mental tasks. In *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2000.

[dRMRMG03] J. del R. Millán, F. Renkens, J. Mouriño, and W. Gerstner. Non-invasive brain-actuated control of a mobile robot. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.

[dRMRMG04] J. del R. Millán, F. Renkens, J. Mouriño, and W. Gerstner. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6):1026–1033, 2004.

[DSW00] E. Donchin, K.M. Spencer, and R. Wijesinghe. The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2):174–179, 2000.

[EE04] A. Erfanian and A. Erfani. ICA-based classification scheme for EEG-based brain-computer interface: the role of mental practice and concentration skills. In *Proc. IEEE Eng Med Biol Soc.*, pages 235–238, 2004.

[EGN03] H.H. Ehrsson, S. Geyer, and E. Naito. Imagery of voluntary movement of fingers, toes, and tongue activates corresponding body-part-specific motor representations. *J Neurophysiol.*, 90(5):3304–3316, 2003.

[ETI02] E.Yom-Tov and G.F. Inbar. Feature selection for the classification of movements from single movement-related potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(3):170–177, 2002.

[EVG03] T. Ebrahimi, J.-M. Vesin, and G. Garcia. Brain-computer interface in multimedia communication. *IEEE Signal Processing Magazine*, 20(1):14–24, 2003.

[Faw06] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[FBWB04] M. Fatourechi, A. Bashashati, R. Ward, and G. Birch. A hybrid genetic algorithm approach for improving the performance of the LF-ASD brain computer interface. In *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, volume 5, pages 345–348, 2004.

[FBWB07] M. Fatourechi, A. Bashashati, R. Ward, and G. Birch. EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, 118(3):480–494, 2007.

[FD88] L.A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.

[FF03] T. Felzer and B. Freisieben. Analyzing EEG signals using the probability estimating guarded neural classifier. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(4):361– 371, 2003.

[FLD+07] D. Friedman, R. Leeb, L. Dikovsky, M. Reiner, G. Pfurtrscheller, and M. Slater. Controlling a virtual body by thought in a highly-immersive virtual environment. In *GRAPP 2007*, pages 83–90, 2007.

[FLG+07] D. Friedman, R. Leeb, C. Guger, A. Steed, G. Pfurtscheller, and M. Slater. Navigating virtual reality by thought: What is it like? *Presence*, 16(1):100–110, 2007.

[FP84] B. Friedlander and B. Porat. The modified Yule-Walker method of ARMA spectral estimation. *IEEE Transactions on Aerospace Electronic Systems*, 20(2):158–173, 1984.

[Fri97] J. H. K. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[FSS+04] D. Friedman, M. Slater, A. Steed, R. Leeb, G. Pfurtscheller, and C. Guger. Using a brain-computer-interface in a highly immersive virtual reality. In *IEEE VR Workshop*, 2004.

[FTH08] J. Fujisawa, H. Touyama, and M. Hirose. EEG-based navigation of immersing virtual environment using common spatial patterns. In *IEEE Virtual Reality Conference (VR '08)*, pages 251–252, 2008.

[Fuk90] K. Fukunaga. *Statistical Pattern Recognition, second edition*. ACADEMIC PRESS, INC, 1990.

[FWB08] M. Fatourechi, R.K. Ward, and G.E. Birch. A self-paced brain-computer interface system with a low false positive rate. *J. Neural Eng.*, 5:9–23, 2008.

[FZO+04] G. M. Friehs, V. A. Zerris, C. L. Ojakangas, M. R. Fellows, and J. P. Donoghue. Brain-machine and brain-computer interfaces. *Stroke*, 35:2702–2705, 2004.

[GC04] E. Gysels and P. Celka. Phase synchronization for the recognition of mental tasks in a brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(4):406–415, 2004.

[GEH+03] C. Guger, G. Edlinger, W. Harkam, I. Niedermayer, and G. Pfurtscheller. How many people are able to operate an EEG-based brain-computer interface (BCI)? *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):145–147, 2003.

[GEV03a] G. N. Garcia, T. Ebrahimi, and J.-M. Vesin. Support vector EEG classification in the fourier and time-frequency correlation domains. In *Conference Proceedings of the First International IEEE EMBS Conference on Neural Engineering*, 2003.

[GEV03b] Gary N. Garcia, T. Ebrahimi, and J.-M. Vesin. Joint time-frequency-space classification of EEG in a brain-computer interface application. *EURASIP J. Appl. Signal Process.*, 2003(1):713–729, 2003.

[GGC+07] G. Garipelli, F. Galán, R. Chavarriaga, P.W. Ferrez, E. Lew, and J. del R. Millán. The use of brain-computer interfacing for ambient intelligence. In *In the book, Constructing Ambient Intelligence: AmI-07 Workshops Proceedings, LNCS, Springer Verlag, 2008.*, 2007.

[GGP+05] R. Grave de Peralta Menendez, S. Gonzalez Andino, L. Perez, P.W. Ferrez, and J. del R. Millán. Non-invasive estimation of local field potentials for neuroprosthesis control. *Cognitive Processing, Special Issue on Motor Planning in Humans and Neuroprosthesis Control*, 6:59–64, 2005.

[GHHP99] C. Guger, W. Harkam, C. Hertnaes, and G. Pfurtscheller. Prosthetic control by an EEG-based brain-computer interface (BCI). In *Proc. AAATE 5th European Conference for the Advancement of Assistive Technology*, 1999.

[GPAR+07a] C. Gouy-Pailler, S. Achard, B. Rivet, C. Jutten, E. Maby, A. Souloumiac, and M. Congedo. Théorie des graphes et dynamiques des connexions cérébrales pour la conception d'interfaces cerveau-machine asynchrones. In *Proc. GRETSI*, pages 673–676, 2007.

[GPAR+07b] C. Gouy-Pailler, S. Achard, B. Rivet, C. Jutten, E. Maby, A. Souloumiac, and M. Congedo. Topographical dynamics of brain connections for the design

of asynchronous brain-computer interfaces. In *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society (IEEE EMBC)*, pages 2520–2523, 2007.

[GPAT03] D. Garrett, D. A. Peterson, C. W. Anderson, and M. H. Thaut. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural System and Rehabilitation Engineering*, 11:141–144, 2003.

[Gui01] S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems*, 9(3):426–443, 2001.

[GXCG03] X. Gao, D. Xu, M. Cheng, and S. Gao. A BCI-based environmental controller for the motion-disabled. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):137–140, 2003.

[HBWF96] A. P. Holmes, R. C. Blair, J. D. Watson, and I. Ford. Nonparametric analysis of statistic images from functional mapping experiments. *Journal of Cerebral Blood Flow and Metabolism*, 16:7–22, 1996.

[HDKG08] M. Hachet, F. Dècle, S. Knödel, and P. Guitton. Navidget for easy 3D camera positioning from 2D inputs. In *Proceedings of IEEE 3DUI - Symposium on 3D User Interfaces*, 2008.

[HdS07] P.S. Hammon and V.R. de Sa. Preprocessing and meta-classification for brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, 54(3):518–525, 2007.

[HGV+05] U. Hoffmann, G. Garcia, J.-M. Vesin, K. Diserens, and T. Ebrahimi. A boosting approach to P300 detection with application to brain-computer interfaces. In *Conference Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering*, 2005.

[HHB+03] T. Hoya, G. Hori, H. Bakardjian, T. Nishimura, T. Suzuki, Y. Miyawaki, A. Funase, and J. Cao. Classification of single trial EEG signals by a combined principal + independent component analysis and probabilistic neural network approach. In *Proceedings ICA2003*, pages 197–202, 2003.

[HLS+01] E. Huupponen, M. Lehtokangas, J. Saarinen, A. Varri, A. Saastamoinen, S. L. Himanen, and J. Hasan. EEG alpha activity detection by fuzzy reasoning. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, pages 411–416, 2001.

[HNP+04] T. Hinterberger, N. Neumann, M. Pham, A. Kübler, A. Grether, N. Hofmayer, B. Wilhelm, H. Flor, and N. Birbaumer. A multimodal brain-based feedback and communication system. *Journal Experimental Brain Research*, 154(4):521–526, 2004.

[HO00] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[HP00] E. Haselsteiner and G. Pfurtscheller. Using time-dependant neural networks for EEG classification. *IEEE transactions on rehabilitation engineering*, 8:457–463, 2000.

[HP04a] N.-J. Huan and R. Palaniappan. Classification of mental tasks using fixed and adaptive autoregressive models of EEG signals. In *IEMBS '04. 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1–5, 2004.

[HP04b] N.-J. Huan and R. Palaniappan. Neural network classification of autoregressive features from electroencephalogram signals for brain-computer interface design. *J. Neural Eng.*, 1:142–150, 2004.

[HPM05] P. Herman, G. Prasad, and T. M. McGinnity. Investigation of the type-2 fuzzy logic approach to classification in an EEG-based brain-computer interface. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005*, pages 5354– 5357, 2005.

[HPM06] P. Herman, G. Prasad, and T. M. McGinnity. A fuzzy logic classifier design for enhancing BCI performance. In *Proc. of the 3rd international Brain-Computer Interface workshop*, pages 42–43, 2006.

[HSF+06] L.R. Hochberg, M.D. Serruya, G.M. Friehs, J.A. Mukand, M.Saleh, A.H. Caplan, A. Branner, D. Chen, R.D. Penn, and J.P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, 2006.

[HST90] A. Hiraiwa, K. Shimohara, and Y. Tokunaga. EEG topography recognition by neural networks. *IEEE Engineering in Medicine and Biology Magazine*, 9(3):39–42, 1990.

[HVE07] U. Hoffmann, J. Vesin, and T. Ebrahimi. Recent advances in brain-computer interfaces. In *IEEE International Workshop on Multimedia Signal Processing*, 2007.

[HVED08] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens. An efficient P300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, 167:115–125, 2008.

[IKP06] K. Inoue, K. Kumamaru, and G. Pfurtscheller. Robot operation based on pattern recognition of EEG signals. In *Proc. of the 3rd international Brain-Computer Interface workshop*, pages 116–117, 2006.

[Jas58] H. Jasper. Ten-twenty electrode system of the international federation. *Electroenceph. Clin. Neurophysiol*, 10:371–375, 1958.

[JC82] A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. *Handbook of Statistics*, 2:835–855, 1982.

[JDM00] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[JH91] C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

[JZ97] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[KA90] Z. A. Keirn and J. I. Aunon. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering*, 37(12):1209–1214, 1990.

[KASC08] A. Kachenoura, L. Albera, L. Senhadji, and P. Comon. ICA: A potential tool for BCI systems. *IEEE Signal Processing Magazine*, 25(1):57–68, 2008.

[KB05] B. Kleber and N. Birbaumer. Direct brain communication: neuroelectric and metabolic approaches at Tübingen. *Cognitive Processing*, 6(1):65–74, 2005.

[KBCM07] R. Krepki, B. Blankertz, G. Curio, and K. R. Müller. The berlin brain-computer interface (BBCI): towards a new communication channel for online control in gaming applications. *Journal of Multimedia Tools and Applications*, 33(1):73–90, 2007.

[KCVP07] J. Kronegg, G. Chanel, S. Voloshynovskiy, and T. Pun. EEG-based synchronized brain-computer interfaces: A model for optimizing the number of mental tasks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(1):50–58, 2007.

[KFN+96] J. Kalcher, D. Flotzinger, C. Neuper, S. Golly, and G. Pfurtscheller. Graz brain-computer interface II: towards communication between humans and computers based on online classification of three different EEG patterns. *Medical and Biological Engineering and Computing*, 34:383–388, 1996.

[KFP93] M. Kubat, D. Flotzinger, and G. Pfurtscheller. Discovering patterns in EEG-signals: Comparative study of a few methods. In *European Conference on Machine Learning*, pages 366–371, 1993.

[KHFH08] A. Kübler, S. Halder, A. Furdea, and A. Hösle. Brain painting - BCI meets art. In *4th International Brain-Computer Interface Workshop and Training Course*, pages 361–366, 2008.

[KKK+01] A. Kübler, B. Kotchoubey, J. Kaiser, J. R. Wolpaw, and N. Birbaumer. Brain-computer communication: unlocking the locked in. *Psychology Bulletin*, 127(3):358–375, 2001.

[KLH05] B. Kamousi, Z. Liu, and B. He. Classification of motor imagery tasks for brain-computer interface applications by means of two equivalent dipoles analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13:166–171, 2005.

[KMG+04] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter. BCI competition 2003–data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076, 2004.

[KMHD06] A. Kübler, V.K. Mushahwar, L.R. Hochberg, and J.P. Donoghue. BCI meeting 2005-workshop on clinical issues and applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):131–134, 2006.

[KNM+05] A. Kübler, F. Nijboer, J. Mellinger, T.M. Vaughan, H. Pawelzik, G. Schalk, D.J. McFarland, N. Birbaumer, and J.R. Wolpaw. Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface. *Neurology*, 34:1775–1777, 2005.

[Koh90] T. Kohonen. The self-organizing map. In *Proceedings of the IEEE*, volume 78, pages 1464–1480, 1990.

[KP00] A. Kostov and M. Polak. Parallel man-machine training in development of EEG-based cursor control. *IEEE Transactions on Rehabilitation Engineering*, 8(2):203–205, 2000.

[KPJ+06] L. Kauhanen, T. Palomäki, P. Jylänki, F. Aloise, M. Nuttin, and J. del R. Millan. Haptic feedback compared with visual feedback for BCI. In *Proceedings of the 3rd International BCI Workshop and Training Course 2006, Graz Austria*, pages 66–67, 2006.

[KPL+07] L. Kauhanen, P.Jylänki, J. Lehtonen, P. Rantanen, H. Alaranta, and M. Sams. EEG-based brain-computer interface for tetraplegics. *Computational Intelligence and Neuroscience*, 2007, 2007.

[KSC+06] D.J. Krusienski, E.W. Sellers, F. Cabestaing, S. Bayoudh, D.J. McFarland, T.M. Vaughan, and J.R. Wolpaw. A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering*, 3:299–305, 2006.

[KTBM08] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller. Towards zero training for brain-computer interfacing. *PLoS ONE*, 3(8):e2967, 2008.

[LBCM05] S. Lemm, B. Blankertz, G. Curio, and K.-R. Müller. Spatio-spectral filters for improving classification of single trial EEG. *IEEE Trans. Biomed. Eng.*, 52(9):1541–1548, 2005.

[LC03] H. Lee and S. Choi. PCA+HMM+SVM for EEG pattern classification. In *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications*, 2003.

[LCAA08] X. Larrodé, B. Chanclou, L. Aguerreche, and B. Arnaldi. OpenMASK: an open-source platform for virtual reality. In *IEEE VR workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, 2008.

[LCL+07] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4:R1–R13, 2007.

[LD04] F. Lamarche and S. Donikian. Crowds of virtual humans : a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum (Eurographics'04)*, 23(3):509–518, 2004.

[LFMP+07] R. Leeb, D. Friedman, G. R. Müller-Putz, R. Scherer, M. Slater, and G. Pfurtscheller. Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic. *Computational Intelligence and Neuroscience, special issue: "Brain-Computer Interfaces: Towards Practical Implementations and Potential Applications"*, 2007, 2007.

[LFS+06] R. Leeb, D. Friedman, R. Scherer, M. Slater, and G. Pfurtscheller. EEG-based "walking" of a tetraplegic in virtual reality. In *Maia Brain Computer Interfaces Workshop 2006 - Challenging Brain Computer Interfaces: Neural Engineering Meets Clinical Needs in Neurorehabilitation*, page 43, 2006.

[LFSP07] R. Leeb, D. Friedman, M. Slater, and G. Pfurtscheller. A tetraplegic patient controls a wheelchair in virtual reality. In *proceedings of BRAINPLAY 2007, playing with your brain*, 2007.

[LJB+07] J.P. Lachaux, K. Jerbi, O. Bertrand, L. Minotti, D. Hoffmann, B. Schoendorff, and P. Kahane. A blueprint for real-time functional mapping via human intracranial recordings. *PLoS ONE*, 2(10):e1094, 2007.

[LKF+05a] E. Lalor, S. P. Kelly, C. Finucane, R. Burke, R. Smith, R. Reilly, and G. McDarby. Steady-state VEP-based brain-computer interface control in an immersive 3-D gaming environment. *EURASIP journal on applied signal processing*, 2005.

[LLA07b] F. Lotte, A. Lécuyer, and B. Arnaldi. FuRIA: A novel feature extraction algorithm for brain-computer interfaces using inverse models and fuzzy region of interest. In *Proc. of the 3rd IEEE-EMBS international Conference on Neural Engineering*, pages 175–178, 2007.

[LLA07c] F. Lotte, A. Lécuyer, and B. Arnaldi. Les interfaces cerveau-ordinateur : Utilisation en robotique et avancées récentes. In *Journées Nationales de la Recherche en Robotique*, 2007.

[LLK+07] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller. Brain-computer communication: Motivation, aim and impact of exploring a virtual apartment. *IEEE, Transactions on Neural Systems & Rehabilitation Engineering*, 2007.

[LLLA07] F. Lotte, A. Lécuyer, F. Lamarche, and B. Arnaldi. Studying the use of fuzzy inference systems for motor imagery classification. *IEEE Transactions on Neural System and Rehabilitation Engineering*, 15(2):322–324, 2007.

[LLR+06] F. Lotte, A. Lécuyer, Y. Renard, F. Lamarche, and B. Arnaldi. Classification de données cérébrales par système d'inférence flou pour l'utilisation d'interfaces cerveau-ordinateur en réalité virtuelle. In *Actes des Premières Journées de l'Association Française de Réalité Virtuelle, AFRV06*, pages 55–62, 2006.

[LLR+08] A. Lécuyer, F. Lotte, R.B. Reilly, R. Leeb, M. Hirose, and M. Slater. Brain-computer interfaces, virtual reality and videogames: Current applications and future trends. *IEEE Computer*, 41(10):66–72, 2008.

[LMS+06] E.C. Leuthardt, K.J. Miller, G. Schalk, R.P.N. Rao, and J.G. Ojemann. Electrocorticography-based brain computer interface-the Seattle experience. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):194– 198, 2006.

[LN06] M.A. Lebedev and M.A.L. Nicolelis. Brain-machine interfaces: past, present and future. *Trends in Neurosciences*, 29(9):536–546, 2006.

[LNM06] R. Lehembre, Q. Noirhomme, and B. Macq. Inverse problem applied to BCI's : Keeping track of the EEG's brain dynamics using kalman filtering. In *Proc. of the 3rd Internationnal Brain Computer Interface Workshop*, pages 32–33, 2006.

[Lot06] F. Lotte. The use of fuzzy inference systems for classification in EEG-based brain-computer interfaces. In *Proc. of the 3rd international Brain-Computer Interface workshop*, pages 12–13, 2006.

[LSC04] S. Lemm, C. Schafer, and G. Curio. BCI competition 2003–data set III: probabilistic modeling of sensorimotor mu rhythms for classification of imaginary hand movements. *IEEE Transactions on Biomedical Engineering*, 51(6):1077–1080, 2004.

[LSC06] S. Lemm, C. Schäfer, and G. Curio. Aggregating classification accuracy across time: Application to single trial EEG. In *NIPS 2006*, pages 825–832, 2006.

[LSF+07] R. Leeb, R. Scherer, D. Friedman, F. Lee, C. Keinrath, H. Bischof, M. Slater, and G. Pfurtscheller. *Towards brain-computer interfacing*, chapter Combining BCI and Virtual Reality: Scouting Virtual Worlds. MIT Press, G. Dornhege, R. Millan Jdel, T. Hinterberger, D. J. Mcfarland & K. R. Müller edition, 2007.

[LSFP07] R. Leeb, V. Settgast, D. Fellner, and G. Pfurtscheller. Self-paced exploration of the austrian national library through thought. *International Journal of Bioelectromagnetism*, 9(4):237–244, 2007.

[LSL+04] R. Leeb, R. Scherer, F. Lee, H. Bischof, and G. Pfurtscheller. Navigation in virtual environments through motor imagery. In *9th Computer Vision Winter Workshop, CVWW'04*, pages 99–108, 2004.

[LWV00] S.J. Luck, G.F. Woodmana, and E.K. Vogel. Event-related potential studies of attention. *Trends in Cognitive Sciences*, 4(11):432–440, 2000.

[MA06a] H. Mouchère and E. Anquetil. Generalization capacity of handwritten outlier symbols rejection with neural network. In *Proceedings of the 10th International Workshop on Frontier in Handwriting Recognition (IWFHR'06)*, pages 187–192, 2006.

[MA06b] H. Mouchère and E. Anquetil. A unified strategy to deal with different natures of reject. In *Proceedings of the International Conference on Pattern Recognition (ICPR'06)*, pages 792–795, 2006.

[MAB03] K.-R. Muller, C. W. Anderson, and G.E. Birch. Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):165– 169, 2003.

[MAC+02] D. Margery, B. Arnaldi, A. Chauffaut, S. Donikian, and T. Duval. OpenMASK: Multi-threaded or modular animation and simulation kernel or kit: a general introduction. In *In: Richir, S., Richard, P., Taravel, B. (eds.) VRIC 2002, Laval, France*, 2002.

[MAM+06] D. J. McFarland, C. W. Anderson, K.-R. Muller, A. Schlogl, and D. J. Krusienski. BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):135 – 138, 2006.

[MAR07] H. Mouchère, E. Anquetil, and N. Ragot. Writer style adaptation in online handwriting recognizers by a fuzzy mechanism approach : The adapt method. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 21(1):99–116, 2007.

[MB00] S. G. Mason and G. E. Birch. A brain-controlled switch for asynchronous control applications. *IEEE Transactions on Biomedical Engineering*, 47(10):1297–1307, 2000.

[MB03] S.G. Mason and G.E. Birch. A general framework for brain-computer interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(1):70–85, 2003.

[MB04] E. Miranda and A. Brouse. Toward direct brain-computer musical interfaces. In *NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression*, pages 216–219, Singapore, Singapore, 2004. National University of Singapore.

[MB06] K.-R. Muller and B. Blankertz. Toward noninvasive brain-computer interfaces. *IEEE Signal Processing Magazine*, 23(5):125–128, 2006.

[MBBB04] S.G. Mason, R. Bohringer, J.F. Borisoff, and G.E. Birch. Real-time control of a video game with a direct brain-computer interface. *Journal of Clinical Neurophysiology*, 21(6):404–408, 2004.

[MBC07] P. Martinez, H. Bakardjian, and A. Cichocki. Fully online multicommand brain-computer interface with visual neurofeedback using SSVEP paradigm. *Comput. Intell. Neuroscience*, 1, 2007.

[MBF+07] S.G. Mason, A. Bashashati, M. Fatourechi, K.F. Navarro, and G.E. Birch. A comprehensive survey of brain interface technology designs. *Annals of Biomedical Engineering*, 35(2):137–169, 2007.

[MCM+95] G.R. McMillan, G.L. Calhoun, M.S. Middendorf, J.H Schuner, D.F. Ingle, and V.T. Nashman. Direct brain interface utilizing self-regulation of steady-state visual evoked response. In *Proceedings of RESNA*, pages 693–695, 1995.

[MEJS00] S. Makeig, S. Enghoff, Tzyy-Ping Jung, and T.J. Sejnowski. A natural basis for efficient brain-actuated control. *IEEE Transactions on Rehabilitation Engineering*, 8(2):208–211, 2000.

[Men95] J. M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377, 1995.

[MGPF99] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, 110(5):787–798, 1999.

[Mil08] J. del R. Millán. Brain-controlled robots. *IEEE Intelligent Systems*, 2008.

[MKD+04] K. R. Müller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz. Machine learning techniques for brain-computer interfaces. *Biomedical Technologies*, 49:11–22, 2004.

[MKH+06] S. Mason, J. Kronegg, J. Huggins, M. Fatourechi, and A. Schloegl. Evaluating the performance of self-paced BCI technology. Technical report, Neil Squire Society, 2006.

[MM99] D. S. Moore and G. P. McCabe. *Introduction to the Practice of Statistics 3rd ed.* W. H. Freeman, 1999.

[MM03] J. del R. Millán and J. Mouriño. Asynchronous BCI and local neural classifiers: An overview of the Adaptive Brain Interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, Special Issue on Brain-Computer Interface Technology*, 2003.

[MMCJ00] M. Middendorf, G. McMillan, G. Calhoun, and K. S. Jones. Brain-computer interfaces based on the steady-state visual-evoked response. *IEEE Transactions on Rehabilitation Engineering*, 8(2):211–214, 2000.

[MMDW97] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw. Spatial filter selection for EEG-based communication. *Electroencephalographic Clinical Neurophysiology*, 103(3):386–394, 1997.

[MML+04] C.M. Michel, M.M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave de Peralta. EEG source imaging. *Clin Neurophysiol.*, 115(10):2195–2222, 2004.

[Moo03] M. M. Moore. Real-world applications for brain-computer interface technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):162–165, 2003.

[MPP08] G.R. Muller-Putz and G. Pfurtscheller. Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on Biomedical Engineering*, 55(1):361–364, 2008.

[MPSNP06] G. Müller-Putz, R. Scherer, C. Neuper, and G. Pfurtscheller. Steady-state somatosensory evoked potentials: suitable brain signals for brain-computer interfaces. *IEEE transactions on neural systems and rehabilitation engineering*, 14(1):30–37, 2006.

[MPSP07] G. Müller-Putz, R. Scherer, and G. Pfurtscheller. Game-like training to learn single switch operated neuroprosthetic control. In *proceedings of BRAIN-PLAY 2007, playing with your brain*, 2007.

[MSB+07] J. Mellinger, G. Schalk, C. Braun, H. Preissl, W. Rosenstiel, N. Birbaumer, and A. Kübler. An MEG-based brain-computer interface (BCI). *Neuroimage*, 36(3):581–593, 2007.

[MW05] D. J. McFarland and J. R. Wolpaw. Sensorimotor rhythm-based brain-computer interface (BCI): feature selection by regression improves performance. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):372–379, 2005.

[Nau97] D. Nauck. Neuro-fuzzy systems: Review and prospects. In *In Proceedings of Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pages 1044–1053, 1997.

[NBL+06] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller. Seperability of four-class motor imagery data using independent components analysis. *Journal of Neural Engineering*, 3:208–216, 2006.

[NCdN06] K.D. Nielsen, A.F. Cabrera, and O.F. do Nascimento. EEG based BCI-towards a better control. brain-computer interface research at aalborg university. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):202–204, 2006.

[NdS05] E. Niedermeyer and F. Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, ISBN 0781751268, 5th edition, 2005.

[NHC+97] W. T. Nelson, L. J. Hettinger, J. A. Cunningham, M. M. Roe, M. W. Haas, and L. B. Dennis. Navigating through virtual flight environments using brain-body-actuated control. In *IEEE Virtual Reality Annual International Symposium*, pages 30–37, 1997.

[Nic01] M.A.L. Nicolelis. Actions from thoughts. *Nature*, 409:403–407, 2001.

[Nie05] E. Niedermeyer. *Electroencephalography: basic principles, clinical applications, and related fields*, chapter The normal EEG of the waking adult. Lippincott Williams & Wilkins, ISBN 0781751268, E. Niedermeyer and F. Lopes da Silva eds edition, 2005.

[NJ02] A. Ng and M. Jordan. On generative vs. discriminative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of Advances in Neural Information Processing*, 2002.

[NKM08] Q. Noirhomme, R.I. Kitney, and B. Macq. Single trial EEG source reconstruction for brain-computer interface. *IEEE Transactions on Biomedical Engineering*, 55(5):1592–1601, 2008.

[NSRP05] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller. Imagery of motor actions: differential effects of kinesthetic and visual-motor mode of imagery in single-trial EEG. *Brain Res Cogn Brain Res.*, 25(3):668–677, 2005.

[OGNP01] B. Obermeier, C. Guger, C. Neuper, and G. Pfurtscheller. Hidden markov models for online classification of single trial EEG. *Pattern recognition letters*, pages 1299–1309, 2001.

[OM95] A. Opalach and S. C. Maddock. An overview of implicit surfaces. In *Introduction to Modelling and Animation Using Implicit Surfaces*, pages 1.1–1.13, 1995.

[ONGP00] B. Obermaier, C. Neuper, C. Guger, and G. Pfurtscheller. Information transfer rate in a five-classes brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(3):283–288, 2000.

[Ope06b] Open-ViBE project, 2006. http://www.irisa.fr/bunraku/OpenViBE/.

[Pal05] R. Palaniappan. Brain computer interface design using band powers extracted during mental tasks. In *Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering*, 2005.

[PBP87] F. Perrin, O. Bertrand, and J. Pernier. Scalp current density mapping: Value and estimation from potential data. *IEEE Transactions on Biomedical Engineering*, BME-34(4):283–288, 1987.

[PBSdS06] G. Pfurtscheller, C. Brunner, A. Schlogl, and F.H. Lopes da Silva. Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1):153–159, 2006.

[PCG+08] L.C. Parra, C. Christoforou, A.D. Gerson, M. Dyrholm, A. Luo, M. Wagner, M.G. Philiastides, and P. Sajda. Spatiotemporal linear decoding of brain state. *IEEE Signal Processing Magazine*, 25(1):107–115, 2008.

[PdS99] G. Pfurtscheller and F. H. Lopes da Silva. Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, 110(11):1842–1857, 1999.

[PFB+07] F. Popescu, S. Fazli, Y. Badower, B. Blankertz, and K.-R. Müller. Single trial classification of motor imagination using 6 dry EEG electrodes. *PLoS ONE*, 2(7):e637, 2007.

[PFK93] G. Pfurtscheller, D. Flotzinger, and J. Kalcher. Brain-computer interface-a new communication device for handicapped persons. *journal of microcomputer application*, 16:293–299, 1993.

[PFK94] P. Pudil, F. J. Ferri, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Pattern Recognition, Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing*, volume 2, pages 279–283, 1994.

[Pfu99] G. Pfurtscheller. EEG event-related desynchronization (ERD) and event-related synchronization (ERS). *Electroencephalography: Basic Principles, Clinical Applications and Related Fields, 4th ed.*, pages 958,967, 1999.

[PGN06] G. Pfurtscheller, B. Graimann, and C. Neuper. *Wiley Encyclopedia of Biomedical Engineering*, chapter EEG-Based Brain-Computer Interface System. John Wiley & Sons, Inc., 2006.

[PGT+06] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, and F. Beverina. P300-based brain computer interface: reliability and performance in healthy and paralysed participants. *Clin Neurophysiol.*, 117(3):531–537, 2006.

[PK92] G. Pfurtscheller and W. Klimesch. Event-related synchronization and desynchronization of alpha and beta waves in a cognitive task. *E. Basar & T. H. Bullock Eds., Induced rhythms in the brain*, pages 117–128, 1992.

[PKK+05] D.A. Peterson, J.N. Knight, M.J. Kirby, C.W. Anderson, and M.H. Thaut. Feature selection and blind source separation in an EEG-based brain-computer interface. *EURASIP J. Appl. Signal Process.*, 2005(1):3128–3140, 2005.

[PLK+06] G. Pfurtscheller, R. Leeb, C. Keinrath, D. Friedman, C. Neuper, C. Guger, and M. Slater. Walking from thought. *Brain Research*, 1071(1):145–152, 2006.

[PM] R.D. Pascual-Marqui. LORETA/sLORETA website.

[PM02] R.D. Pascual-Marqui. Standardized low resolution brain electromagnetic tomography (sLORETA): technical details. *Methods and Findings in Experimental and Clinical Pharmacology*, 24D:5–12, 2002.

[PM07] R.D. Pascual-Marqui. Discrete, 3D distributed, linear imaging methods of electric neuronal activity. part 1: exact, zero error localization. Technical report, arXiv:0710.3341v2, 2007.

[PMP+03] G. Pfurtscheller, G.R. Muller, J. Pfurtscheller, H.J. Gerner, and R. Rupp. 'thought' - control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. *Neuroscience Letters*, 351(1):33–36, 2003.

[PMPPR05] G. Pfurtscheller, G.R. Müller-Putz, J. Pfurtscheller, and R. Rupp. EEG-based asynchronous BCI controls functional electrical stimulation in a tetraplegic patient. *EURASIP J. Appl. Signal Process.*, 2005(1):3152–3155, 2005.

[PN01] G. Pfurtscheller and C. Neuper. Motor imagery and direct brain-computer communication. *proceedings of the IEEE*, 89(7):1123–1134, 2001.

[PNB05] G. Pfurtscheller, C. Neuper, and N. Birbaumer. *Motor cortex in voluntary movements*, chapter Human brain-computer interface, pages 367–401. CRC Press, riehle a, vaadia e. edition, 2005.

[PNFP97] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and Clinical Neurophysiology*, 103:642–651, 1997.

[PNG+00] G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, H. Ramoser, A. Schlogl, B. Obermaier, and M. Pregenzer. Current trends in Graz brain-computer interface (BCI) research. *IEEE Transactions on Rehabilitation Engineering*, 8(2):216–219, 2000.

[PNM+03] G. Pfurtscheller, C. Neuper, G.R. Muller, B. Obermaier, G. Krausz, A. Schlogl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, M. Wortz, G. Supp, and C. Schrank. Graz-BCI: state of the art and clinical applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):1–4, 2003.

[PNSL98] G. Pfurtscheller, C. Neuper, A. Schlogl, and K. Lugger. Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE Transactions on Rehabilitation Engineering*, 6(3):316–325, 1998.

[PPNS02] R. Palaniappan, R. Paramesran, S. Nishida, and N. Saiwaki. A new brain-computer interface design using fuzzy ARTMAP. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10:140–148, 2002.

[PR99a] W.D. Penny and S.J. Roberts. EEG-based communication via dynamic neural network models. In *Proceedings of International Joint Conference on Neural Networks*, 1999.

[PRCS00] W.D. Penny, S.J. Roberts, E.A. Curran, and M.J. Stokes. EEG-based communication: a pattern recognition approach. *IEEE Transactions on Rehabilitation Engineering*, 8(2):214–215, 2000.

[PSL+07] G. Pfurtscheller, R. Scherer, R. Leeb, C. Keinrath, C. Neuper, F. Lee, and H. Bischof. Viewing moving objects in virtual reality can change the dynamics of sensorimotor EEG rhythms. *Presence*, 16(1):111–118, 2007.

[PSVH03] J. A. Pineda, D. S. Silverman, A. Vankov, and J. Hestenes. Learning to control brain rhythms: Making a brain-computer interface possible. *IEEE transactions on neural systems and rehabilitation engineering*, 11(2):181–184, 2003.

[QDH04] L. Qin, L. Ding, and B. He. Motor imagery classification by means of source analysis for brain computer interface applications. *Journal of Neural Engineering*, 1(3):135–141, 2004.

[QLC05] J. Qin, Y. Li, and A. Cichocki. ICA and committee machine-based algorithm for cursor control in a BCI system. *Lecture notes in computer science*, 2005.

[Rab89] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[RAERL05] R. Ron-Angevin, A. Díaz Estrella, and A. Reyes-Lecuona. Development of a brain-computer interface (BCI) based on virtual reality to improve training techniques. *Applied Technologies in Medecine and Neuroscience*, pages 13–20, 2005.

[RBG+07] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. Leong Teo, Q. Zeng, C. Laugier, and M. H. Ang Jr. Controlling a wheelchair indoors using thought. *IEEE Intelligent Systems*, 22(2):18–24, 2007.

[Rei05] E.L. Reilly. *Electroencephalography: basic principles, clinical applications, and related fields*, chapter EEG recording and operations of the apparatus. Lippincott Williams & Wilkins, ISBN 0781751268, E. Niedermeyer and F. Lopes da Silva eds edition, 2005.

[RG08] A. Rakotomamonjy and V. Guigue. BCI competition III: Dataset II - ensemble of SVMs for BCI P300 speller. *IEEE Trans. Biomedical Engineering*, 55(3):1147–1154, 2008.

[RGC+07] Y. Renard, G. Gibert, M. Congedo, F. Lotte, E. Maby, B. Hennion, O. Bertrand, and A. Lécuyer. OpenViBE: An open-source software platform to easily design, test and use brain-computer interfaces. In *BCI Meets Robotics: Challenging Issues in Brain-Computer Interaction and Shared Control, MAIA workshop*, page 49, 2007.

[RGMA05b] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado. Ensemble of SVMs for improving brain computer interface P300 speller performances. In *International Conference on Artificial Neural Networks*, 2005.

[RH97] Y. D. Rubinstein and T. Hastie. Discriminative vs informative learning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[RJ91] S. J. Raudys and A. K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, 1991.

[RMGP00] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, 2000.

[RS07a] B. Rivet and A. Souloumiac. Extraction de potentiels évoqués P300 pour les interfaces cerveau-machine. In *Proc. GRETSI*, pages 625–628, 2007.

[RS07b] B. Rivet and A. Souloumiac. Subspace estimation approach to P300 detection and application to brain-computer interface. In *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society (IEEE EMBC)*, pages 5071–5074, 2007.

[RTNS06] S. Rezaei, K. Tavakolian, A. M. Nasrabadi, and S. K. Setarehdan. Different classification techniques considering brain computer interface applications. *Journal of Neural Engineering*, 3:139–144, 2006.

[SBRS99] V.J. Samar, A. Bopardikar, R. Rao, and K. Swartz. Wavelet analysis of neuroelectric waveforms: A conceptual tutorial. *Brain and Language*, 66(1):7–60, 1999.

[Sch05] A. Schlögl. Results of the BCI-competition 2005 for datasets IIIa and IIIb. Technical report, Institute of Human-Computer Interface Graz University of Technology, 2005.

[SCWM06] A. Schwartz, X. Cui, D. Weber, and D. Moran. Brain-controlled interfaces: Movement restoration with neural prosthetics. *Neuron*, 52(1):205–220, 2006.

[SD06] E.W. Sellers and E. Donchin. A P300-based brain-computer interface: initial tests by ALS patients. *Clin Neurophysiol.*, 117(3):538–548, 2006.

[SEGYS07] T. Solis-Escalante, G.G. Gentiletti, and O. Yanez-Suarez. Detection of steady-state visual evoked potentials based on the multisignal classification algorithm. In *3rd International IEEE/EMBS Conference on Neural Engineering, 2007. CNE '07*, pages 184–187, 2007.

[SGM+03] P. Sajda, A. Gerson, K. R. Müller, B. Blankertz, and L. Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, 11(2):184–185, 2003.

[SKSP03] A. Schlögl, C. Keinrath, R. Scherer, and G. Pfurtscheller. Information transfer of an EEG-based bran-computer interface. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, 2003.

[SLBP05] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of Neural Engineering*, pages L14–L22, 2005.

[SLMA06] J. Sreng, A. Lécuyer, C. Mégard, and C. Andriot. Using visual cues of contact to improve interactive manipulation of virtual objects in industrial assembly/maintenance simulations. *IEEE Trans. Vis. Comput. Graph.*, 12(5):1013–1020, 2006.

[SLP97] A. Schlogl, K. Lugger, and G. Pfurtscheller. Using adaptive autoregressive parameters for a brain-computer-interface experiment. In *Proceedings 19th International Conference IEEE/EMBS*, pages 1533–1535, 1997.

[SLS+08] R. Scherer, F. Lee, A. Schlögl, R. Leeb, H. Bischof, and G. Pfurtscheller. Towards self-paced brain-computer communication: Navigation through virtual worlds. *IEEE, Transactions on Biomedical Engineering*, 55(2):675–682, 2008.

[SM04] S. Solhjoo and M. H. Moradi. Mental task recognition: A comparison between some of classification methods. In *BIOSIGNAL 2004 International EURASIP Conference*, 2004.

[Smi97] S. W. Smith. *The Scientist & Engineer's Guide to Digital Signal Processing*. ISBN 0-9660176-3-3, 1997.

[Smi02] L. Smith. A tutorial on principal components analysis, 2002.

[SMN+04] R. Scherer, G. R. Muller, C. Neuper, B. Graimann, and G. Pfurtscheller. An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering*, 51(6):979–984, 2004.

[SNG05] S. Solhjoo, A. M. Nasrabadi, and M. R. H. Golpayegani. Classification of chaotic signals using HMM classifiers: EEG-based mental task classification. In *Proceedings of the European Signal Processing Conference*, 2005.

[SNP02] A. Schlogl, C. Neuper, and G. Pfurtscheller. Estimating the mutual information of an EEG-based brain-computer interface. *Biomed Tech*, 47(1-2):3–8, 2002.

[SSL+07] R. Scherer, A. Schloegl, F. Lee, H. Bischof, J. Janša, and G. Pfurtscheller. The self-paced Graz brain-computer interface: Methods and applications. *Computational Intelligence and Neuroscience*, 2007, 2007.

[Sto05] J.V. Stone. *The Encyclopedia of Statistics in Behavioral Science*, chapter Independent Component Analysis. Wiley, BS Everitt and DC Howell (eds) edition, 2005.

[Sun07] S. Sun. Ensemble learning methods for classifying EEG signals. *LECTURE NOTES IN COMPUTER SCIENCE*, 4472:113–120, 2007.

[TAA08] A. Tzelepi, R. Ron Angevin, and A. Amditis. Neural activity related to visual cues and saccade preparation for operating a brain conputer interface. In *INTUITION international conference*, 2008.

[TAH08] H. Touyama, M. Aotsuka, and M. Hirose. A pilot study on virtual camera control via steady-state VEP in immersing virtual environment. In *Proceedings of the third IASTED conference on Human Computer Interaction*, pages 611–615, 2008.

[TDAM06] R. Tomioka, G. Dornhege, K. Aihara, and K.-R. Müller. An iterative algorithm for spatio-temporal filter optimization. In *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course 2006*, pages 22–23, 2006.

[TGP04] G. Townsend, B. Graimann, and G. Pfurtscheller. Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(2):258–265, 2004.

[TGW06] M. Thulasidas, C. Guan, and J. Wu. Robust classification of EEG signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1):24–29, 2006.

[TH07b] H. Touyama and M. Hirose. Steady-state VEPs in CAVE for walking around the virtual world. In *HCI*, volume 6, pages 715–717, 2007.

[THS+01] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A.T. Helser. VRPN: a device-independent, network-transparent VR peripheral system. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55–61, New York, NY, USA, 2001. ACM Press.

[TLK04] A. Tzelepi, A. Lutz, and Z. Kapoula. EEG activity related to preparation and suppression of eye movements in three-dimensional space. *Experimental Brain Research*, 155(4):439–449, 2004.

[TMK+03] K. Tanaka, K. Matsunaga, N. Kanamori, S. Hori, and H.O. Wang. Electroencephalogram-based control of a mobile robot. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 2, pages 688–693, 2003.

[TMW05] K. Tanaka, K. Matsunaga, and H.O. Wang. Electroencephalogram-based control of an electric wheelchair. *IEEE Transactions on Robotics*, 21(4):762–766, 2005.

[TR04] K. Tavakolian and S. Rezaei. Classification of mental tasks using gaussian mixture bayesian network classifiers. In *Proceedings of the IEEE International workshop on Biomedical Circuits and Systems*, 2004.

[TRM06] L.J. Trejo, R. Rosipal, and B. Matthews. Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):225–229, 2006.

[Vap99] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transaction on Neural Networks*, 10(5):988–999, 1999.

[VHMM00] M. Varsta, J. Heikkonen, J. R. Millan, and J. Mourino. Evaluating the performance of three feature sets for brain-computer interfaces with an early stopping MLP committee. In *Proceedings. 15th International Conference on Pattern Recognition*, 2000.

[Vid73] J. J. Vidal. Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering*, pages 157–180, 1973.

[VML+07] G. Vanacker, J.R. Millán, E Lew, P.W. Ferrez, F. Galán Moles, J. Philips, H. Van Brussel, and M. Nuttin. Context-based filtering for assisted brain-actuated wheelchair driving. *Computational Intelligence and Neuroscience*, 2007:Article ID 25130, 12 pages, 2007.

[VMS+06] T.M. Vaughan, D.J. McFarland, G. Schalk, W.A. Sarnacki, D.J. Krusienski, E.W. Sellers, and J.R. Wolpaw. The wadsworth BCI research and development program: at home with BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):229–233, 2006.

[VSCP04] C. Vidaurre, A. Schlogl, R. Cabeza, and G. Pfurtscheller. A fully on-line adaptive brain computer interface. *Biomed. Tech. Band, Special issue*, 49:760–761, 2004.

[Wan92] L. X. Wang. Fuzzy systems are universal approximators. *IEEE International Conference on Fuzzy Systems*, 1992.

[WBM+02] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, and T.M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

[WDH04] T. Wang, J. Deng, and B. He. Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns. *Clinical Neurophysiology*, 115(12):2744–2753, 2004.

[WGG05] W. Wu, X. Gao, and S. Gao. One-versus-the-rest (OVR) algorithm: An extension of common spatial patterns(CSP) algorithm to multi-class case. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005*, pages 2387– 2390, 2005.

[WLA+06] J.R. Wolpaw, G.E. Loeb, B.Z. Allison, E. Donchin, O.F. do Nascimento, W.J. Heetderks, F. Nijboer, W.G. Shain, and J. N. Turner. BCI meeting 2005–workshop on signals and recording methods. *IEEE Transaction on Neural Systems and Rehabilitation Engineering.*, 14(2):138–141, 2006.

[WM04] J.R. Wolpaw and D.J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc Natl Acad Sci U S A*, 101(51):49–54, 2004.

[WMB+04] N. Weiskopf, K. Mathiak, S.W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer. Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering*, 51(6):966–970, 2004.

[WMNF91] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78:252–259, 1991.

[WMV00] J.R. Wolpaw, D.J. McFarland, and T.M. Vaughan. Brain-computer interface research at the wadsworth center. *IEEE Transactions on Rehabilitation Engineering*, 8(2):222–226, 2000.

[Wol92] D.H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[Wol07] J.R. Wolpaw. Brain-computer interfaces as new brain output pathways. *J Physiol*, 579:613–619, 2007.

[WZL+04] Y. Wang, Z. Zhang, Y. Li, X. Gao, S. Gao, and F. Yang. BCI competition 2003–data set IV: an algorithm based on CSSD and FDA for classifying single-trial EEG. *IEEE Transactions on Biomedical Engineering*, 51(6):1081–1086, 2004.

[Yag04] R.R. Yager. On the retranslation process in Zadeh's paradigm of computing with words. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(2):1184–1195, 2004.

[YHS05] Y.P.A. Yong, N.J. Hurley, and G.C.M. Silvestre. Single-trial EEG classification for brain-computer interfaces using wavelet decomposition. In *European Signal Processing Conference, EUSIPCO 2005*, 2005.

[YTI03] E. Yom-Tov and G. F. Inbar. Detection of movement-related potentials from the electro-encephalogram for possible use in a brain-computer interface. *Medical and Biological Engineering and Computing*, 41(1):85–93, 2003.

[Zad96a] L. A. Zadeh. Fuzzy sets. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh*, pages 19–34, 1996.

[Zad96b] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.

[ZL06] M. Zhong and A. Lécuyer. Automatic elimination of ocular and muscle artifacts in EEG recordings based on blind source separation. Technical Report PI 1817, IRISA, 2006.

[ZLGL08] M. Zhong, F. Lotte, M. Girolami, and A. Lécuyer. Classifying EEG for brain computer interfaces using gaussian processes. *Pattern Recognition Letters*, 29:354–359, 2008.

[Zuc00] W. Zucchini. An introduction to model selection. *Journal of Mathematical Psychology*, 44:41–61, 2000.

[ZWG+07] D. Zhang, Y. Wang, X. Gao, B. Hong, and S. Gao. An algorithm for idle-state detection in motor-imagery-based brain-computer interface. *Computational Intelligence and Neuroscience*, 2007(1):5–5, 2007.

List of Figures

Résumé

Une Interface Cerveau-Ordinateur (ICO) est un système de communication qui permet à ses utilisateurs d'envoyer des commandes à un ordinateur via leur activité cérébrale, cette activité étant mesurée, généralement par ÉlectroEncéphaloGraphie (EEG), et traitée par le système.

Dans la première partie de cette thèse, dédiée au traitement et à la classification des signaux EEG, nous avons cherché à concevoir des ICOs interprétables et plus efficaces. Pour ce faire, nous avons tout d'abord proposé FuRIA, un algorithme d'extraction de caractéristiques utilisant les solutions inverses. Nous avons également proposé et étudié l'utilisation des Systèmes d'Inférences Flous (SIF) pour la classification. Nos évaluations ont montré que FuRIA et les SIF pouvaient obtenir de très bonnes performances de classification. De plus, nous avons proposé une méthode utilisant ces deux algorithmes afin de concevoir une ICO complétement interprétable. Enfin, nous avons proposé de considérer la conception d'ICOs asynchrones comme un problème de rejet de motifs. Notre étude a introduit de nouvelles techniques et a permis d'identifier les classifieurs et les techniques de rejet les plus appropriés pour ce problème.

Dans la deuxième partie de cette thèse, nous avons cherché à concevoir des applications de Réalité Virtuelle (RV) controlées par une ICO. Nous avons tout d'abord étudié les performances et les préférences de participants qui interagissaient avec une application ludique de RV à l'aide d'une ICO asynchrone. Nos résultats ont mis en évidence le besoin d'utiliser des ICO adaptées à l'utilisateur ainsi que l'importance du retour visuel. Enfin, nous avons développé une application de RV permettant à un utilisateur d'explorer un musée virtuel par la pensée. Dans ce but, nous avons conçu une ICO asynchrone et proposé une nouvelle technique d'interaction permettant à l'utilisateur d'envoyer des commandes de haut niveau. Une première évaluation semble montrer que l'utilisateur peut explorer le musée plus rapidement avec cette technique qu'avec les techniques actuelles.

Abstract

A Brain-Computer Interface (BCI) is a communication system which enables its users to send commands to a computer by using brain activity only, this brain activity being measured, generally by ElectroEncephaloGraphy (EEG), and processed by the system.

In the first part of this thesis, dedicated to EEG signal processing and classification techniques, we aimed at designing interpretable and more efficient BCI. To this end, we first proposed FuRIA, a feature extraction algorithm based on inverse solutions. This algorithm can automatically identify relevant brain regions and frequency bands for classifying mental states. We also proposed and studied the use of Fuzzy Inference Systems (FIS) for classification. Our evaluations showed that FuRIA and FIS could reach state-of-the-art results in terms of classification performances. Moreover, we proposed an algorithm that uses both of them in order to design a fully interpretable BCI system. Finally, we proposed to consider self-paced BCI design as a pattern rejection problem. Our study introduced novel techniques and identified the most appropriate classifiers and rejection techniques for self-paced BCI design.

In the second part of this thesis, we focused on designing virtual reality (VR) applications controlled by a BCI. First, we studied the performances and preferences of participants who interacted with an entertaining VR application, thanks to a self-paced BCI. Our results stressed the need to use subject-specific BCI as well as the importance of the visual feedback. Then, we developed a VR application which enables a user to explore a virtual museum by using thoughts only. In order to do so, we designed a self-paced BCI and proposed an interaction technique which enables the user to send high-level commands. Our first evaluation suggested that a user could explore the museum faster with this interaction technique than with current techniques.