Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

SMAD - A System for Multimodal Assistive Domotics

Thesis by

Flavio Ferrara

Student ID

787714

Advisor: **Prof. Matteo Matteucci** (Politecnico di Milano)

Co-advisor: **Prof. Teodiano Freire Bastos Filho** (Universidade Federal do Espirito Santo)

A.A. 2013 – 2014

Back in 2008, it was hard to tell what that fresher was up against. Lots of experiences followed one another, and wonderful people accompanied through this peculiar path. Let's focus on the recent past, still remembering that what I am is the result of the whole journey, not just the very end.

So, thanks to Matteo Matteucci, who trusted me going to develop this work 10.000 km far away.

Thanks to the whole Politecnico di Milano, which pleased my desire to discover new places and cultures.

A deep acknowledgment to Teodiano Bastos, who allowed this work to happen in the first place and warmly welcomed me at UFES. Thanks to Alexandre and everyone else at UFES Intelligent Automation Laboratory for supporting me and softening those long days at lab.

Heartfelt thanks to all the Italian and Brazilian friends and my parents, for being such an essential part of my life so far. I need this bond to be stronger and stronger, in order to face what's next.

E obrigado você, que me aguentou e me deu força, e que merece um lugar muito especial.

FF

Abstract

During the last decades there has been major growth in the application of technology for assisting persons with disabilities. The main goal is to alleviate their discomfort in a variety of tasks, mainly routine activities, by providing more self-sufficiency and reduce the dependency on external help. A large part of everyday life is spent at home. The use of common home accessories seems perfectly natural to most of people; nevertheless, it could require a lot of time and effort for individuals with motor disabilities. Therefore, supporting persons with mobility impairments when at home would be very valuable.

Several technologies originally developed in medical area are now employed in the context of assisting individuals with motor disabilities. Examples of this are Electromyography (EMG), Electrooculography (EOG) and Electroencephalography (EEG). Such techniques can drive the development of interfaces for Human-Computer Interaction (HCI) in order to supply the subject with useful and rehabilitating technological tools.

In this work, an assistive system called SMAD is presented to control home devices from a wheelchair through biological signals such as EMG and EEG. The wheelchair monitor displays a menu through which the user selects the desired device and operate it. Albeit in the literature there are plenty of approaches to use biological signals in Human-Computer Interaction, many of them lack in realizing a tool that would be usable at home by a regular user with disabilities. SMAD focuses on end-user features such as ease of setup and use as well as low cost and customizability. Moreover, unlike many approaches in literature, the functional design of the interface has been driven by an user-centered performance analysis. Additionally, the system has been conceived to be adaptable to the user's degree of disability. This means that, if she still presents muscle voluntary control skills, they should be exploited.

Two scenarios are presented in this thesis, using respectively EMG/EOG and EEG. Nevertheless, it will be possible to implement different techniques just plugging the new components into the system, exploiting the remaining of the architecture with no effort. Experimental results

show satisfying performance for both EMG/EOG and EEG cases. Offline analyses exhibited very reliable and fast identification of user intention. After offline analyses, the system has been tested in realistic online sessions. The user succeeded in employing SMAD to operate home devices, simulating common needs of persons with disabilities.

In conclusion, the main contributions of this work are:

- Realizing an assistive system that employs various biological signals in order to help persons with motor disabilities to handle home accessories from a wheelchair.
- Designing an interface based on performance from the end-user point of view and proposing it as a basic block to implement different kinds of control paradigms.

Ampio estratto

Negli ultimi decenni si è assistito a una sensibile crescita nell'applicazione di nuove tecnologie per l'assistenza alle persone con disabilità, o tecnologie assistive. L'obiettivo principale è di alleviare il disagio nello svolgere diverse azioni, soprattutto quotidiane, fornire maggiore autosufficienza e ridurre la dipendenza dall'aiuto esterno. Gran parte della vita di tutti i giorni si svolge a casa. L'uso dei comuni accessori casalinghi sembra perfettamente naturale alla maggior parte delle persone; tuttavia, può richiedere molto tempo e impegno per individui con disabilità motorie gravi. Di conseguenza, supportare tali persone quando sono nella propria casa è un importante elemento di ricerca nel campo delle tecnologie assistive.

Varie tecnologie basate su segnali biologici, sviluppate originariamente in campo medico, sono oggi impiegate nell'assistenza a portatori di handicap motori. In letteratura troviamo esempi come Elettromiografia (EMG), Elettrooculografia (EOG) e Elettroencefalografia (EEG). Queste tecniche possono essere adottate nello sviluppo di interfacce per l'interazione uomo-computer (*Human-Computer Interation* o HMI), dotando il soggetto di uno strumento tecnologico utile e riabilitante.

In questa tesi, viene presentato un sistema assistivo chiamato SMAD per controllare dispositivi casalinghi da una sedia a rotelle, mediante segnali biologici come EMG e EEG. Il monitor della sedia visualizza un menu attraverso il quale l'utente può selezionare il dispositivo desiderato e azionarlo. Sebbene in letteratura si trovi un'abbondanza di approcci che utilizzano segnali biologici nell'interazione uomo-computer, molti di questi peccano nella realizzazione di strumenti che possano essere utilizzati da un comune utente disabile nella propria casa. Questa tesi, invece, si concentra su proprietà orientate all'utente finale come facilità di installazione e uso, basso costo e personalizzabilità. Inoltre, a differenza di molti approcci in letteratura, il progetto funzionale dell'interfaccia è stato guidato da un'analisi di prestazioni centrata sull'utente. In aggiunta, il sistema è stato concepito per essere adattabile al grado di disabilità dell'utente. Ciò significa che le restanti capacità di controllo volontario dei muscoli possono essere sfruttate.

In questa tesi sono presentati due scenari, usando rispettivamente EMG/EOG e EEG, per impartire comandi e quindi azionare dispositivi; il primo conta su gesti facciali e movimenti degli occhi, mentre il secondo necessita solo che l'utente fissi degli stimoli pulsanti sullo schermo. Allo stesso modo, è possibile sviluppare differenti tecniche di controllo e semplicemente inserire il nuovo componente nel sistema, utilizzando il resto dell'architettura senza necessità di ulteriore adattamento.

In conclusione, i principali contributi sono:

- Realizzare un sistema assistivo che impieghi vari segnali biologici per assistere persone con disabilità motorie nell'uso di accessori casalinghi.
- Permettere che il sistema sia utilizzabile senza addestramento e senza costi o tempi di sistemazione proibitivi, dato che è impiegato un dispositivo di cattura dei segnali economico e facile da usare.
- Progettare un'interfaccia basata sulle prestazioni dal punto di vista dell'utente e proporre quest'ultima come componente di base per implementare vari tipi di paradigmi di controllo.

Entrambi gli approcci di interazione presentati sono stati testati in sessioni sperimentali. I risultati ottenuti sono stati pienamente soddisfacenti. Per quanto riguarda il controllo per mezzo di movimenti del viso e degli occhi, l'esito dei test effettuati con otto soggetti senza disabilità ha dimostrato che, tramite il sistema, è possibile esprimere da 14 a 53 bits di informazione al minuto. Quattro soggetti hanno partecipato agli esperimenti impiegando la modalità relazionata all'attività cerebrale. I risultati indicano che anche questo tipo di interazione è pienamente utilizzabile, potendo esprimere fino a 15 bits al minuto.

Nel Capitolo 1 vengono esposti gli obiettivi e le premesse alla base di SMAD. Inoltre, presentiamo una breve indagine dei lavori pionieristici e dello stato dell'arte nel campo dell'interazione uomo-macchina usando segnali biologici.

Il Capitolo 2 presenta gli aspetti generali del controllo assistivo per mezzo di segnali biologici. Definiamo il concetto di Trasduttore di Segnali

Biologici (*Biological Signals Transducer* o BST) che riveste un ruolo centrale in SMAD. Quindi, descriviamo le diverse strategie che il BST può adottare. Il capitolo termina focalizzandosi su un tipo particolare di BST chiamato interfaccia cervello-computer.

Nel Capitolo 3 si espone l'architettura di SMAD e il framework all'interno del quale il sistema è stato sviluppato. Inoltre, si analizza alcuni dei requisiti funzionali e la struttura modulare impiegata. Infine, l'ultima sezione introduce le metriche necessarie a valutare le prestazioni del sistema.

Il Capitolo 4 descrive il modulo relativo alla Control Interface (CI). Per prima cosa, illustriamo gli aspetti generali del modulo e dell'interfaccia grafica. Il capitolo continua con l'analisi su cui si basa il design funzionale della CI e con la descrizione delle scelte implementative più importanti. L'ultima sezione sottolinea l'importanza di una CI comune per valutare e comparare le prestazioni di BST diversi.

Il Capitolo 5 è dedicato ai due moduli BST sviluppati in SMAD. Il primo BST presentato impiega EMG e EOG per riconoscere gesti del viso e movimenti degli occhi. Il capitolo introduce anche Emotiv EPOC, il dispositivo di cattura dei segnali senza fili usato per entrambi i BST. La sezione seguente si occupa del secondo BST, che traduce i segnali EEG in comandi per il modulo della CI. Tutti gli stadi del processamento dell'EEG sono descritti in dettaglio.

Tutti gli esperimenti e i test svolti per questa tesi sono riassunti nel Capitolo 6. Sono documentate le varie sessioni sperimentali, insieme alla discussione dei risultati ottenuti.

Per concludere, il Capitolo 7 riporta le considerazioni finali riguardanti l'intero lavoro, con suggerimenti e idee per future sperimentazioni e miglioramenti.

Contents

List of Figures

List of Tables

List of Acronyms

BCI Brain-Computer Interface – a human-computer interface that acts as a non-muscular communication and control channel monitoring brain activity.

BST Biological Signal Transducer – the functional block that converts biosignals into control signals useful for HMI.

CCA Canonical Correlation Analysis – a statistical method to evaluate the correlation between sets of variables.

EMG Electromyography – measures electrical current induced inside the muscles during their contraction by neuromuscular activities.

EOG Electrooculography – a technique for assessing the corneo-retinal standing potential between the front and the back of the eye.

EEG Electroencephalography – measures electric field induced during synaptic excitation within the neurons of the brain.

HMI Human-Machine Interaction – a discipline concerned with the design, evaluation and implementation of the interfaces between people and machines.

MSI Multivariate Synchronization Index – an indicator for estimating the synchronization between multivariate signals.

PSDA Power Spectral Density Analysis – a method for analyzing a signal in the frequency domain.

SSVEP Steady-State Visual Evoked Potential – a BCI paradigm based on the user gazing visual stimuli that keep repeating at a certain frequency.

Introduction 1

Nowadays technology is increasingly pervading our life. With technological advance we obtain more and more tools and we gain brand-new capabilities; it is enough to think about the wave of mobile applications, which allows us to even increase the regular faculty of communicating and perceiving the world. From the viewpoint of persons with disabilities, what would be more useful than fancy apps are technological means through which recover lost faculties. This need led off the field of *assistive technologies*, a wide research area that is intended to increase, maintain or improve functional capabilities of individuals with disabilities [1].

More specifically, this work addresses the problems of persons with motor impairments. In Italy, according to the Instituto Nazionale di Statistica (ISTAT), the number of citizens older than six with motor disabilities reaches 1.3% of the total population, with peaks of 1.7% among women and 9.6% among octogenarians[1]. These people are likely to spend a substantial part of their time at home; for this reason, a key issue is how to improve the quality of this time and enhance their self-sufficiency. In this work, an assistive

[1] ISTAT: La disabilità in Italia. `http://www3.istat.it/dati/catalogo/20100513_00/arg_09_37_la_disabilita_in_Italia.pdf`

system called SMAD is proposed that allows the user to operate several home accessories at a distance, for example from the living room to the bedroom, in an autonomous and relatively easy way. The system setup should be straightforward too, consisting only in plugging the desired accessories into a central management box.

A significant resource to let a user exploit assistive systems like SMAD are biological signals (hereafter *biosignals*), that is, the electrical activity of human organism that can be measured and recorded. There are a variety of biosignals depending upon the tissue, organ or cell assortment which generates them. Examples of those are:

- Electrocardiogram (ECG) produced by the heart.
- Electromyography (EMG) produced by skeletal muscles. A subset of them, namely surface EMG (sEMG), determines muscle function by recording the activity above the skin.
- Electrooculography (EOG) produced in the eye.
- Electroencephalography (EEG) produced by brain cells.

Biosignals hold information generated voluntarily or not by the subject. By way of automatic signal processing it is possible to extract such information and use it to build an assistive system. Since the first attempts in doing so, the progress has been considerable, but controlling things with thought is still an unrealistic demand. There is a series of major problems involved with processing biosignals. First, the cost and complexity of medical devices used to record them. Second, the high level of noise and subjective artifacts in such signals, which makes the identification harder. Then, the difficulty in developing modular components, since often the optimal solution suits only the specific problem it addressed. Last but not least, as always when human subjects are concerned, several ethical questions exist that the researcher needs to take care of.

In this work, some of these problems are studied in order to propose feasible solutions. The adoption of a low-cost and easy-to-use recording device is evaluated to verify if it can substitute typical medical equipment. This work tries to adopt and develop robust algorithms and operating conditions, as well as to promote independent modules allowing the reuse of standard components.

Two control paradigms are offered, one employing sEMG together with EOG, the other employing EEG. The latter is often called Brain-Computer Interface (BCI) due to the fact that no voluntary muscle activation is needed. The goal of developing two distinct prototypes is to show that various control modalities can be developed and used as independent components, so that the most adequate modality can be chosen depending on the user's degree of disability. It is worth noting that the rest of system is transparent to the alteration of the control modality, that is, it does not affect in any way its behavior.

Moreover, it is often the case that assistive system are difficult to compare against each other. This is ascribable to the lack of common metrics and the deep differences between one system and another. The research should look for reliable means to standardize components: it would assist the comparison between assistive systems, thus helping the progress in the field. In this work, a metric from the BCI field is adopted and generalized with the aim of driving the design of the system. The very same metric can assess the system performance using different control paradigms, providing a plausible indicator for evaluating them.

In concluding, the main elements of this work are:

- Utilizing a consumer device to relieve the user from complex and costly medical devices for recording biological signals.
- Proposing an assistive control system with a high reliability and a good inter-subject robustness.

- Developing modular components that can be reused within other projects and constitute a basis for comparison between systems.

1.1 Aims and general assumptions

Realizing assistive systems that exploit biological signals is a very active line of research all around the world. A number of applications have been proposed by several groups, presenting impressive results. Nevertheless, a crucial issue to face is how to make their way out of the laboratory and actually become products that can assist people's demands. A number of questions turn this process hard to deal with: a large inter-subject variability in performance, the complexity of the signal acquisition and several external variables that are hard to control - think about the user hair, in the case of BCI systems. Moreover, it frequently happens that performance is measured by a biased metric, showing optimistic or anyway not significant results. As a consequence, even with state-of-the-art approaches, assistive technologies struggle to become a handy tool. In [2] a survey among BCI users with disabilities found that none of them could imagine using a BCI in everyday life. The main problems were related to the low-speed and the inconvenience of recording equipment. The study started from and was driven by BCI research. Analyzing related problems and common solutions, we realized that it is straightforward to generalize some concepts, originally proposed in BCI field, to other kinds of biological signals.

The goal of SMAD is to provide the end-user with a practical daily assistive platform, as close as possible to a product that works out of the box. The system design has adopted an user-centric view. This is not so usual in the field: many times the focus is on proposing innovative techniques or advancing the state of the art of a certain method.

To make SMAD a useful and sufficiently user-friendly tool, we have defined the following aspects:

- All the adopted techniques for obtaining biological electrical activity rely on non-invasive sensors. This means not entering or penetrating the body. Moreover, an easy-to-use and low-cost equipment has been preferred to more traditional ones, such as medical-grade amplifiers.
- For the user not to be confused and tired while managing the system, the number of possible commands has been kept low.
- No training or calibration period is required to utilize the system. The user is able to give commands since the very first moment the system is turned on.
- Every error performed by the system contributes to discourage the user; thus, a more delayed action is preferred over performing an erroneous one sooner.
- The system is intended to be part of a complete platform that includes the effective operation of devices as well as the monitoring of system and user data.

To fulfill the goal presented above, it is essential to pay attention to system design as well as realistic performance evaluation. In particular:

- The system development follows common guidelines from software engineering. Functionally distinct components have been decoupled as much as possible. Logical code has to be separated from data structures making the user customization immediate.
- A well-grounded metric has been employed for avoiding unrealistic off-line analyses.

Finally, the modularity of SMAD can also potentially support the research in the field. The module in charge of command identification is decoupled by the one responsible of command interpretation and actuation. The latter provides an interface of remote procedures; thereby, other projects can benefit from some functional components just calling them from their own platform.

Indeed, command interpretation does not need to involve any knowledge about command identification, i.e., how the user intention is extracted from biosignals. In particular, we argue that the development of a common interpretation component would be very valuable for the evaluation of the command identification module: it may be part of a standardized performance evaluation procedure, that is, a benchmark, against which to test different command identification paradigms in a direct, non-ambiguous fashion.

1.2 Related works

Back in the XIX century, the first ones to record electrical signals emitted by muscular nerves were Carlo Matteucci and Emil Du Bois-Reymond using a galvanometer. Since then, most of research involved EMG, EOG and EEG.

EMG turns popular starting since the 1990s. In many cases it is utilized to control artificial prosthesis, such as in [3]. More recently, sEMG has been employed in input device emulation. For example, in [4] the forearm muscles were used to move a mouse cursor and click. An analogous work, but with facial muscles, can be found in [5]. These works show good performance, being EMG relatively easy to measure and fast to process. They also have the advantage of being non-invasive. On the other hand, there is an obvious issue: it is necessary for the user to have voluntary control of muscles.

In many neural diseases that affect motor area of the brain, the voluntary muscle contraction is lost, but the individual is still able to move the eyes [6]. Two techniques are commonly applied to interpret eye movements: camera tracking or EOG analysis. Camera-based approaches depend on video cameras (or infrared) and image processing software. Their main limitations are due to illumination and eye closure, although in recent past good results have been achieved (see [7]). Conversely, EOG measures the electric potential between the cornea and the retina. In such a way, it is possible to detect eye movements even in total darkness and even if eyes are closed. EOG has already been adopted in [8] for controlling a robotic harm and in [9] to drive a robotic

wheelchair. Moreover, there are examples of projects combining EMG and EOG with success, such as [10]. This is the approach we follow in this work.

In some situations, for example in patients with Amyotrophic Lateral Sclerosis (ALS), spinal cord tumors, or acute muscular dystrophy, there is no remaining voluntary muscle control. If the cognitive abilities are intact, cerebral activity can be used for communication and control, realizing what is called a Brain-Computer Interface (BCI). Usually, BCIs employ EEG to record neurons' electrical activity on the scalp. In 2000 the first meeting about BCI took place and the distinctive traits of such systems were presented in [11]. During the next decade several applications and a variety of systems have been proposed implementing various paradigm. Some examples are: environment control with visual evoked potential ([12]), robot control by motor imagery ([13]), keyboard simulation ([14]), assistive speller ([15]) and wheelchair navigation ([16], [17]). Furthermore, BCI systems have been realized that exploit residual muscle control abilities. Such approaches, which use BCI technology as an additional channel, are called *hybrid BCIs*. An example of combining EMG with EEG activity is presented in [18].

A modern survey of the status of research and the main issues about BCI can be found in [19]. According to the authors, there exist four areas for BCI application in assistive technologies, namely, "Communication and Control", "Motor Substitution", "Entertainment", and "Motor Recovery". The work presented in this thesis fits into the first one: "Communication and Control".

At Universidade Federal do Espirito Santo (UFES), where we realize SMAD, several assistive systems based on biosignals have been developed. A presentation of a multimodal interface for a robotic wheelchair is offered in [20]. The wheelchair is suitable for people with various degrees of disabilities, by allowing several modalities of user control, including through biosignals. Furthermore, the user has the possibility to directly command the wheelchair or rely on autonomous navigation.

In particular, this thesis is inspired by the BCI control of the robotic

wheelchair presented in [21] and on the extension presented in [22], which improves the signal processing and employs a hybrid system to turn on and off the system. Both these works adopt the Steady-State Visual Evoked Potential (SSVEP) paradigm due to its high reliability and speed.

1.3 Publications

The following scientific publications have so far resulted from this work:

- Flavio Ferrara, Alexandre Bissoli, Teodiano Bastos-Filho. "Designing an Assistive Control Interface based on Utility". *Proceedings of the 1st International Workshop on Assistive Technology IWAT 2015*, Vitoria, Brazil, on CD-ROM, 142–145, 2015.

- Alexandre Bissoli, Flavio Ferrara, Mariana Sime, T. Bastos-Filho. "A Multimodal Assistive System to Operate a Smart Environment". *Proceedings of the 1st International Workshop on Assistive Technology IWAT 2015*, Vitoria, Brazil, on CD-ROM, 103–106, 2015.

- Richard J. M. G. Tello , Alexandre L. C. Bissoli, Flavio Ferrara, Sandra Müller, Andre Ferreira, Teodiano F. Bastos-Filho, "Development of a Human Machine Interface for Control of Robotic Wheelchair and Smart Environment" in *11th IFAC Symposium on Robot Control SYROCO*, Brazil, Aug. 2015, submitted.

1.4 Structure of the thesis

The outline for the chapters of this thesis is as follows:

Chapter 2 presents the general aspects about assistive control exploiting biological signals. From the notion of biosignals we define the concept of Biological Signals Transducer (BST) that is central in SMAD. Then, we describe different strategies available to define the BST behavior. The

chapter ends focusing on a particular type of BST, called brain-computer interface.

Chapter 3 describes the architecture of SMAD. The first section presents the general framework within which the system has been developed. Then, we analyze the functional requirements and modular structure. Some details about implementation are provided too. Finally, the last section introduces and explains the metrics needed to evaluate the system performance.

Chapter 4 is about the Control Interface (CI) module. Firstly, a section illustrates general aspects about CI module and the Graphical User Interface. The chapter continues with the formal analysis behind CI functional design. Next, the main implementation choices are described. The last section underlines the importance of a common CI module for evaluating BST performance.

Chapter 5 presents the two BST modules developed in SMAD. The first BST employs EMG and EOG in order to recognize facial gestures and eye movements. The chapter also introduces Emotiv EPOC, the wireless recording equipment used for both BSTs. The next sections concern the second BST, which translates EEG signals into commands for the CI module. All the stages involved in EEG processing are described in details.

Chapter 6 summarizes all the experiments and tests that have been carried out for this thesis, along with the discussion on the experimental results.

Chapter 7 reports final considerations about the entire work and suggestions for future tests and improvements.

Assistive control by means of biological signals 2

This chapter covers various topics about the realization of assistive systems that allow control through biological signals.
The first section introduces the notion of biological, bioelectrical signals and how to measure such signals. It also examines the properties of the biosignals used in this work.
The second section defines the concept of Biological Signal Transducer (BST), while the third and fourth parts present the different techniques available for the BST to interpret user signals and provide device control.
Finally, the fifth section explores a particular kind of BST, i.e., the so-called Brain-Computer Interface.

2.1 Biological signals

Usually the main concerns about motor disabilities are just about the loss of mobility, which is indeed a major issue. Although, the loss of communication and control abilities may represent a problem as important as that. Motor disabilities can indeed affect hands or mouth, precluding the subject from talking or using machines. In other words, these impairments provoke a deficit in the capability of expressing information. Assistive technologies can truly help subjects with motor disabilities to recover such capabilities. Biological signals are an essential information source for motor-impaired individuals. Through signal processing techniques, we can extract from them information related to the subject intention. This information can be conveyed by the subject using voluntarily or spontaneous physiological phenomena. For example, some systems exploit involuntarily cerebral reflexes to convey information about user intention.

Biosignals consist of any signals generated in living organisms. In practice, the term biosignals is often used to identify bioelectrical signals, that is, electrical activity generated by membranes and cells. Different types of cell have the ability of generating electrical potential. For example, in the nervous system electrochemical reactions induce small electric currents. The role of such signals in nerves is to carry information and control various functions of the human body ([23]).

2.1.1 Acquiring biological signals

There are several methods to measure and record biosignals, including non-electrical ones. These methods can be divided in two categories: invasive methods and non-invasive methods. Invasive methods rely on micro-electrodes implanted beneath the skin. For example, invasive EMG needs subcutaneous electrodes on the harm or the leg, while Electrocorticography (ECoG) places electrodes directly on the cortical surface. Albeit the sensibly higher quality of

recorded signals, methods like ECoG are rarely used in engineering in virtue of the delicate surgery required. In the case of invasive EMG the surgery is easier, but the subject still needs to undergo a medical treatment.

Hence, a major effort has been made in developing non-invasive techniques. The existing approaches vary both in the type of biological information measured and the technology used. For instance, neuroimaging includes techniques using magnetic field for depicting images of the nervous system; we can cite functional magnetic resonance imaging (fMRI) for measuring oxygenation levels and magnetoencephalography (MEG) for measuring electrical currents. However, in order to use biosignals as a communication and control channel it is very useful to have a high temporal resolution. Thus, the system being able to provide an output by recording and analyzing a small segment of data, in the order of seconds or less, would be a great advantage. Moreover, the equipment for collecting the signal should be kept as portable and non-invasive as possible. For these reason, the preferred techniques are based on electrodes that measure electric currents on skin or skull surface.

Since most researches on biosignals involve non-invasive electrodes, this kind of recording equipment has become widespread. Manufactures commercialize various type of electrodes. A fundamental difference is between *active* and *passive* ones. Active electrodes are furnished with built-in circuitry that amplifies the electrical signals. Passive electrodes have no inbuilt circuitry and rely on an external amplifier. For passive electrodes to reach the same signal quality of active ones, they need skin preparation as well as conductive gel or saline solution to help reduce impedance.

Passive electrodes with conductive gel imply a long setup time and relatively high inconvenience, e.g., removing the gel from hair. The setup complexity may represent a major problem for the user. A survey among ALS patient presented in [24] suggests that the setup time should be more than fifteen times shorter than the expected period using the system. In other words, for a 5 hours continuous use, the user should be subject to a setup time shorter than 20 minutes.

Active electrodes, also called *dry*, are making their way into the labs. In [25], a valid comparison between active and passive electrodes for recording EEG demonstrates that using dry electrodes does not entail a sensibly loss in performance. An analogous work for EMG can be found in [26].

Recently, research focused on the increasing demand for wearable devices. For instance, [27] presents a review of wearable EEG technology and a survey among neurologist that confirms the need for more comfortable devices for biosignals acquisition.

2.1.2 EMG, EOG and EEG signals

As previously introduced, this work focuses on three specific bioelectrical signals: electromyography (EMG), electrooculography (EOG) and electroencephalography (EEG). Figure 2.1 sketches the three cases.

The EMG signal measures electrical current induced inside the muscles during their contraction by neuromuscular activities. The less invasive method for recording EMG consists in placing electrodes on the surface of the skin; it is called *surface EMG*, or sEMG. An in-depth examination of sEMG nature can be found in [28]. Even if the recorded electrical activity owns a very small potential (in the order of microvolts), by sEMG it is possible to grasp muscular patterns invisible with naked eye. Several techniques were proposed that permit a relatively precise detection of discrete events in sEMG ([29]). On the other side, some weakness exist too. sEMG lacks a common methodology to guide electrode placement; thus, the same pattern may be represented differently between one study and another. Most systems monitor only a few muscle sites; this strongly limits the rich information conveyed by the neuromuscular system. Finally, even if way less invasive than alternative techniques, sEMG employes electrodes and leads that can influence user's postures and movement habits.

Electrical activity is generated inside human eyes too. Electrooculography (EOG) measures the corneo-retinal standing potential between the front and

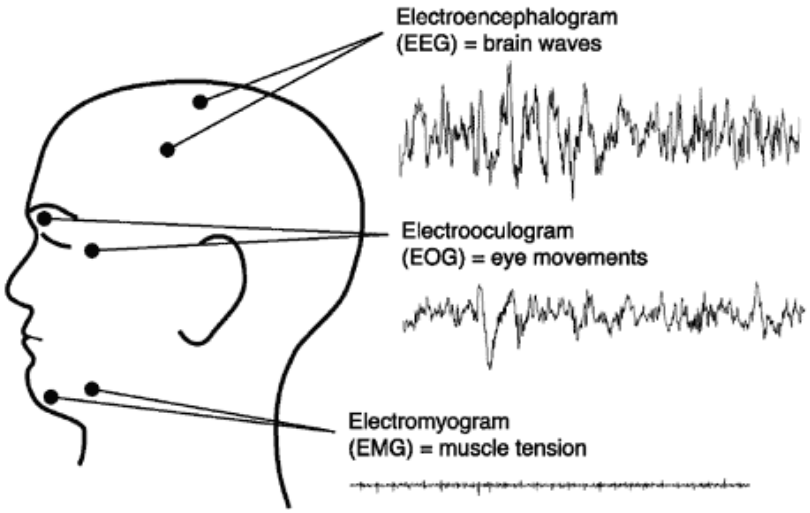


FIGURE 2.1: Description of EMG, EOG and EEG sources on the human head.

the back of the eye. By positioning electrodes next to the eyes, a potential can be captured when the eye moves from center position toward one of the sensors. The electrodes can be placed on the lids, i.e., above and below the eyes, or on the side of the eye. Depending on the placement, the electrodes would more efficiently recognize vertical movements of the eyes, when they are on the lids, or horizontal movements when they are on the sides. EOG is a relatively simple and robust technique. Obviously, it assumes that the user has the ability of voluntarily control the movements of the eyes.

Finally, a large part of the electrical activity induced by human body stays inside the brain. Through electrodes put on the skull, electroencephalography (EEG) measures the electric current that flows during synaptic excitation within the neurons of the brain. Such a current suffers strong attenuation for which the skull is the main responsible; therefore, large populations of neurons are needed to generate an adequately strong potential. As a consequence, EEG is characterized by poor spatial resolution: each electrode records signal within a range of various centimeters around it, and this represents a problem when trying to identify spatial patterns in the brain activity. Moreover, the noise internally produced by the brain mixes with the noise produced over the

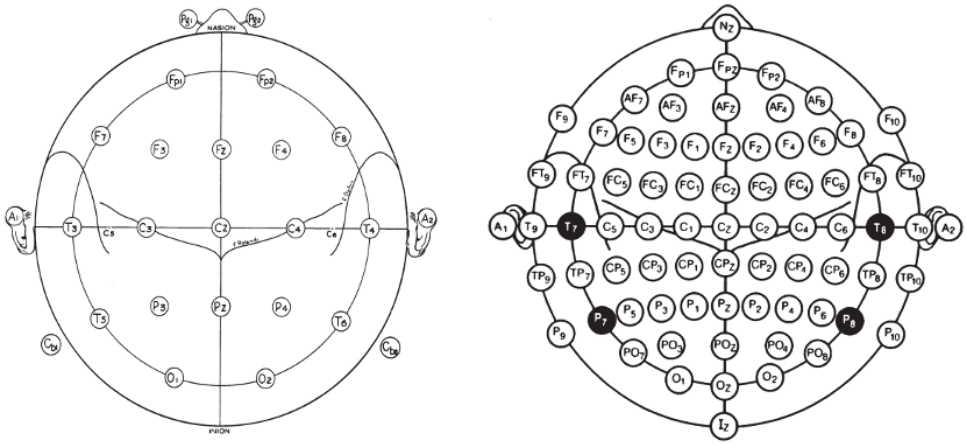


Figure 2.2: On the left, the original 10-20 system. On the right, the extension also known as 10-10 system. Adapted from [31].

scalp, making EEG signal processing a very delicate task.

A recognized standard in electrode placement for EEG exists. Originally 21 locations were proposed with established distances, in a system known as *10-20 international system* ([30]). Thereafter an extension to the original system was proposed which involved an increase of the number of electrodes up to 74 and that is currently the standard of the American Electroencephalographic Society ([31]). Figure 2.2 shows these standards. The electrode locations are named according to the brain lobe and a number that identifies the hemisphere. The letters F, T, C, P and O stand for Frontal, Temporal, Central, Parietal and Occipital respectively. The extended system offers more granularity. For example, new locations exist, e.g., AF (Anterior Frontal), as well as combinations of the classical ones such as FC (FrontoCentral). To identify the hemisphere, even numbers indicate right hemisphere, odd numbers indicate left hemisphere while the letter "z" refers to the midline.

More recently, extensions have been proposed that further increase the electrode density ([32]). Nonetheless, most systems limit the number of channels to a few dozens using 10-10 system.

2.2 Biological Signal Transducer (BST)

Most definitions of signal agree about the following: a signal is a quantity varying in time or space that conveys some kind of information. Biosignals are signals in this sense since they convey information generated voluntarily or not through neuro-physiological phenomena over time. Nonetheless, they are not immediately utilizable for tasks such as communication and Human-Machine Interaction (HMI) in general. The main issues are:

- Noise and artifacts generated inside user body, which are undesired collateral activity registered by the electrodes.
- Noise due to ambient and electrical equipment.
- Inherent instability of biosignals.

Therefore, biosignals need to be processed in order to extract information for the desired application. Such a processing transforms the raw biosignals into other signals meaningful for HMI. Let's try to isolate the functional block responsible for this transformation. It can be imagined as a transducer: analogously to an electrical or physical transducer, it converts a signal from one form (bioelectrical) to another (for example, a logical on/off switch). That characterizes what we hereafter call a Biological Signal Transducer (BST). This kind of transducer allows the HMI through biosignals, as depicted in Figure 2.3.

Usually, HMI occurs by means of some kind of muscular actions. Transducers are the system components that translate such muscular actions into a control signal. This is true for traditional and more innovative forms of HMI. For instance, when writing with a keyboard, the muscular action of pressing keys is translated into characters generation. Similarly, newer input devices like Microsoft's Kinect use a more complex transducing system, but still exploit user gestures and movements realized through muscular activity. For some individuals, such as persons with disabilities or elderly people, these muscular actions are impossible to achieve or too demanding. In such cases, a BST may

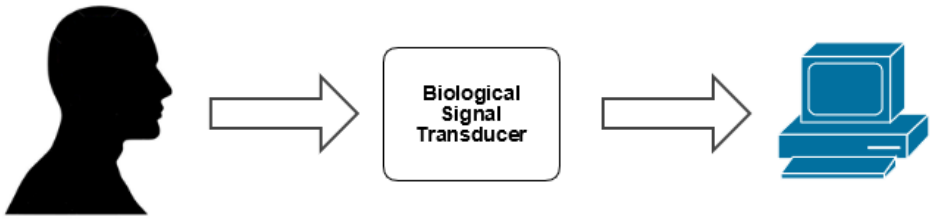


Figure 2.3: Biological Signal Transducer as the tool that permits Human-Machine Interaction.

constitute an essential tool for assistance and rehabilitation. A BST needs just the minimum amount of muscular activity, sometimes nothing at all. Yet, since we leave the conventional interaction channels, serious problems exist to recognize the user intention in a reliable way.

It is important to note that the concept of BST is independent of the type of biosignals one can use. The nature of a specific biosignal may alter the processing internal sub-modules, but does not affect the general BST abstraction. For example, EMG-based and EEG-based BST likely employ different signal processing pipelines; still, the BST continues to transform the input bioelectrical signal into an output control signal.

The idea of BST is also independent of the application, that is, the operations the machine performs after the interaction with the user. BST does nothing, but providing an output signal that can be meaningful for some applications but not for others. Moreover, after the application and the domain of HMI has been defined, distinct but compatible BSTs can be used interchangeably without affecting the rest of the system.

2.3 BST translation

By means of the translation performed by the BST the properties of the input biosignal are altered. A biosignal originates from an analog source and therefore it is continuous in time, although it is often subject to quantization during the recording process. Instead, the control signals that the BST provides could

assume various forms depending on the preferred approach. Two cases can be distinguished:

- When BST output is a continuous control signal we talk about *regression*.
- When BST output represents a logical class out of a set of possible ones the approach is called *classification*.

With regression the time continuity of the input signal acquired from the user is preserved, although with different features and often normalized. It offers the advantages of having an infinity of possible values in output and offering a more natural behavior. In fact, the control signal might show higher intensity in response to a stronger indication of user intention in the biosignal. Regression is particularly appropriate in Input Device Emulation (e.g., mouse emulation in [33]), which is an important design criteria for assistive technology [34]. On the other side, some substantial drawbacks exist with regression. One fundamental property desired for a control signal is being stable, in the sense that it does not exhibit sudden variations caused by noise or other accidental reasons. When using regression, to ensure stability for the control signal may be difficult since in the continuous domain it is often troublesome to handle noise and to realize algorithms that offer robustness with respect to the high instability of most biosignals.

The most popular approach is classification. Applying classification requires to fix the number of possible output values, that is, the different patterns the algorithm looks for. In many cases classification algorithms, or *classifiers*, are provided with training data, a set of data containing observations whose class is known. The classifier thus learns how to distinguish the features of the presented patterns. The training data can belong to the intended user of the system; in this way, the classifier will adapt to the peculiarity of that user's signals.

There are simple classifiers that do not need any training examples. They merely discriminates among the classes using features extracted by the algorithm and according to fixed rules. For example, suppose that each feature

expresses to which degree an example belongs to a certain class. A basic rule-based classifier may assign the class corresponding to the maximum feature value. The main advantage of such classifiers over trained ones is that the user can start issuing commands to the system in no time; the classifier performance ought to express a global behavior, roughly similar among all users. As a shortcoming, *untrained* classifiers do not adapt to the user at all and their performance will not improve after prolonged user practice.

For the purposes of this thesis, the proposed BSTs utilize classification by means of an untrained classifier. Section 5.2.2 describes in details the approach adopted.

2.3.1 BST control paradigms

The interaction between the user and the BST substantially affects the user experience with the system. How the interaction protocols are defined can influence user perception of usability, engagement and/or fatigue. Intuitively, the less bounded the interaction, the more positive the experience. This means that ideally the system should be designed to leave the user free to interact at any moment, without timings or rigid protocols. Since the described autonomy of interaction is extremely difficult to achieve, often a trade-off between user convenience and implementation issues is sought. We can identify three interaction modalities: synchronous control, asynchronous control and hybrid control, or system-paced.

Synchronous control

When the system drives the interaction we talk about synchronous control. Through some sensory channel, the user is notified when the BST is accepting input; thus, it is possible to give commands just within time windows defined by the system. It is the most common approach, being the simplest to implement, but at the same time it causes much fatigue and frustration. In fact, the user needs to comply with the system timings. Within the predefined time windows, the system is continuously analyzing user biosignals and waiting for

valid commands. Two problems can occur. First, the user may try to issue a command when the system is not "listening". Second, when in an input time period, the user needs to be very careful: involuntary idle activities may be interpreted as commands and may result in unattended actions.

Asynchronous control

Also called *self-paced*. With asynchronous control the BST is always available to receive commands. The user decides when to give a command; during user idle, the system is supposed to stand still. This means that the system should robustly recognize when the user is idle, thus issuing no command, even for long periods of time. This approach is the most convenient from the viewpoint of user autonomy. Nevertheless, asynchronous control is more difficult to manage, since user signals while idle are very similar to the signals while performing voluntary control.

Hybrid control

Also known as *system-paced*. The user has the possibility to activate or disable intentional control. When control is active, this modality is equivalent to synchronous control. However, the user can decide to temporarily disable the command triggering system. That allows the user to stay idle without worrying about provoking involuntary commands. It is an advantage from the user viewpoint and it is easier to implement than the asynchronous approach. Yet, it is important to equip the user with a way to turn the system back on; this is recognized as a difficult problem. Further details on this problem are provided in Section 3.2.

With the aim of making the user as comfortable as possible while using the system, in this work a system-paced approach has been chosen. Section 3.2 presents the details about system behavior.

2.3.2 From user commands to device control

HMI aims to enhance the user's ability to express information for the purpose of communicating with the outside world and controlling external devices. For communication, information needs to be encoded in commonly understood symbols and exhibited, e.g., well-known latin letters or small iconic images. For control, even if the final goal is not to express messages but to operate devices, systems use internal symbols for encoding the possible actions as well as the devices. Thus, in both cases, expressing a large amount of information leads to a high number of encoding symbols. How to translate the output from BST into a message encoded in a rich alphabet?

When converting the BST signal into such an alphabet, two strategies are mainly adopted: *direct conversion* or *symbol mapping*.

Direct conversion

In direct conversion, all symbol choices are available anytime the user can issue a command. In other words, there is neither hierarchical nor sequential structure for the options: they are organized in a flat fashion. Direct conversion could be implemented with both continuous and discrete BSTs. In the first case, the options are graphically organized and the BST output signal is used to lead the cursor. When the cursor stands over the desired option, the user is able to select it through another channel, or waiting some time idle. In the case of discrete BST, direct conversion consists of a simple one-to-one mapping between BST logical classes and symbol choices. Despite being the most efficient strategy, the number of classes that the BST supports is likely much less than the number of symbols in output.

Symbol mapping

In symbol mapping, we suppose some kind of indirect relationship between the BST output and the set of symbols. In other words, the BST output, or user commands, do not correspond directly to a symbol meaningful for

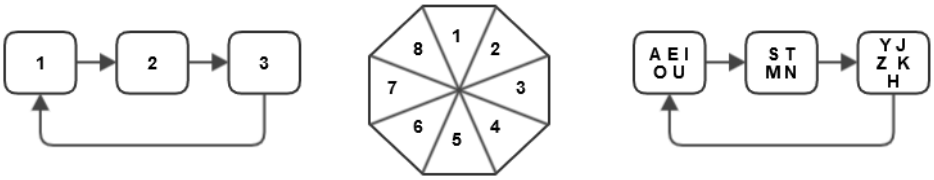


FIGURE 2.4: Option scanning approaches. On the left, the simple linear scanning. In the middle, a simple scanning with options organized circularly. On the right, an example of recursive scanning. After selecting a group, the options inside it will populate the menu for a further selection. For instance, when selecting the first group, another menu will appear with five options: A, E, I, O or U.

the application of interest. Despite being possible to use mapping with a continuous BST, it is more commonly employed with discrete commands. Very often BSTs support only few logical classes. Hence, more symbols are indeed available than user commands. In this case, symbol mapping could be quite inefficient, but it is the only feasible way. In fact, it can be used even with a binary user choice.

A particular kind of symbol mapping is option scanning. The interface presents the user with a set of options organized in a menu. The menu structure defines how to navigate between the options. Scanning could be *simple* or *recursive*. In simple scanning, the single symbol options are presented one at a time and are scanned linearly or circularly. In recursive scanning, multiple symbol options are grouped and presented for selection. The selected group is expanded and the options contained are scanned recursively until the desired one is reached. Figure 2.4 provides examples of different approaches. Recursive scanning alleviates the slowness of simple scanning and it has been adopted in various assistive speller, such as [15] and [35].

Taking into consideration the assumption of a reduced number of user commands, i.e., BST output, and the possibility of an unbounded number of devices, SMAD employs option scanning though a menu. This menu, presented in details in Chapter 4, combines simple and recursive scanning.

2.4 BCI as BST

When the BST input consists only of brain activity we talk about Brain-Computer Interface (BCI). This activity is usually recorded through EEG. A BCI does not rely on any muscular channel as brain's output pathway ([11]). Yet, it may depends on some conventional channel (e.g., control of eye gaze direction) to elicit the message sought. In this case the BST is intended as a substitute of the operating part of the neuromuscular system.

In contrast with EMG-based BST, a BCI-based BST is directly connected to the source of nervous signals instead of peripheral nerves. The difference is the same as providing a shortcut around a neural break and providing an entirely new path. However, this path should not be imagined as a straightforward equivalence, such as thinking about words to produce mechanical enunciation of such words. The conversion is often more cumbersome. In many scenarios, the user would spell a letter by thinking about some movements or by reacting to an event. A BCI may transform nervous activity that would produce no muscular action into an actual result affecting the external world. This is done according to different paradigms.

In the following, the diverse paradigms used in EEG-based BCI are presented and briefly discussed. The last subsection is devoted to the technique employed in this thesis.

2.4.1 BCI paradigms

Information related to different neuro-physiological phenomena can be extracted from EEG recordings. Some phenomena are deliberately controlled by the user, while others are due to involuntary reactions to external stimuli. Examples of the former are Event-Related Desynchronization (ERD) or motor imagery, and Slow Cortical Potentials (SCP). The most common technique that exploit involuntary reflexes are Event-Related Potentials (ERP), especially P300 and Steady-State Visually Evoked Potentials (SSVEP). Being the paradigm employed in this thesis, more details about SSVEP are provided.

ERD and Motor Imagery

In awake people, when idle, motor cortical area displays a periodic EEG activity in the 8-12 Hz range, called mu rhythm (μ). μ is often associated with 18-26 Hz activity, called beta rhythm (β). Movement, or preparation for movement, usually coincides with a reduction in μ and β rhythms. This reduction has been labeled as Event-Related Desynchronization or ERD. After performing the movement the opposite phenomenon happens, i.e, a rhythm increase or Event-Related Synchronization (ERS). What is more important for BCI use is that the occurring of ERD and ERS is not tied to the physical realization of the movement; they happen also with imagined movement, namely *motor imagery*. Several BCI systems have been developed exploiting this paradigm, the first being presented in [33]. Such systems rely especially on hands and feet motor imagery, because these movements are relatively easy to envision and provoke an ERD/ERS classifiable with good accuracy.

Slow Cortical Potentials

Slow cortical potentials (SCPs) are potential shifts in the EEG with a very low frequency. It has been demonstrated that users can learn to voluntarily produce negative and positive shifts when provided with sensory feedback of their brain potential. Negative SCPs are typically associated with functions involving cortex activation. Positive SCPs, on the contrary, correspond to reduced cortical activation [11].

P300

P300 are a kind of Event-Related Potentials evoked by the presentation of uncommon or surprising stimuli, when interleaved with other frequent or routine stimuli. Approximately 300 ms after the unexpected stimulus, or *oddball*, is presented, a positive peak (P300) appears in EEG recording. Employing P300 does not require initial user training: it is an innate reflex that triggers involuntarily. At the same time, it has been argued that P300 is likely to change

over time. Hence, the translation algorithm needs to take care of possible user adaptation.

SSVEP

Another form of ERP are Visually Evoked Potentials (VEP). VEPs are elicited by visual stimuli such as repeated flashes or reversal graphical patterns. These potentials reflect the cortex mechanisms for vision and visual pathways. The physiological models and theories at the basis of VEPs are presented in details in [36]. At least three different visual pathways contribute to the formation of VEPs. The role of each pathway is linked to characteristics of the stimuli, like colors and contrast.

When the stimulation is brief, the visual system reacts to sudden changes in the input with transient responses. Therefore, the elicited VEPs are called transient VEPs. A longer stimulation using periodically reversal stimuli has demonstrated to produce a stable VEP of small amplitude, called steady-state VEP or SSVEP. Figure 2.5 illustrates the difference between transient VEPs and SSVEP. The presence of frequency components that remain constant over a long time period distinguishes SSVEP from transient potentials; this appears more evidently looking to the spectral content of SSVEP.

Transient VEPs are useful for studying brain's visual dynamics, for example the time that it takes for a visual stimulus to travel from the eye to the occipital cortex. Yet, SSVEP are much more effective while dealing with many applications, including HMI. In fact, according to [36], SSVEP is less affected by artifacts coming from EMG and EOG contamination.

Resuming, SSVEPs are periodic electric potentials induced by flickering visual stimuli. The visual stimulus blinking at a certain frequency elicits a response in the brain at the same frequency plus its even harmonics. Thus, when several EEG trials are averaged during SSVEP, the signal shows a stationary spectrum with characteristic peaks at the stimulus frequency and even harmonics.

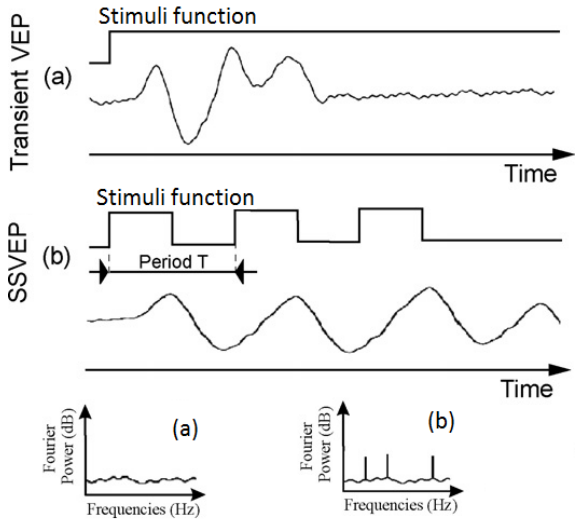


FIGURE 2.5: Differences between transient VEP and Steady-State VEP. Adapted from [36].

Most researches agree that SSVEP is generated in the occipital area of the cortex, especially the primary visual area (V1). Three main components are often distinguished: the low-frequency response with a maximum around 10 Hz, the medium-frequency response in 15–25 Hz range and high-frequency response in 40–60 Hz range. Lower frequency responses are characterized by higher latency. Hence, the lower the stimulus frequency, the slower will be the elicitation of SSVEP. Moreover, the low-frequency component does not immediately disappear after turning off the stimulus.

Albeit intensive research has been done, the mechanism underlying SSVEP is not yet fully understood. Literature disagrees about the frequency range that elicit SSVEPs. While there is consensus on 3-40 Hz range [36], in [37] SSVEP has been reported SSVEP in 1-100 Hz range. Other authors [38] set the minimum frequency for SSVEP to 6 Hz. Moreover, there exist open questions about stimuli type and the medium through which they are displayed. Systematic studies proposing consistent comparison among different media and types (LED and LCD, checkerboard and plain), although available, are still insufficient. Finally, photosensitive epilepsy issues related to the use of SSVEP are seldom studied.

2.4.2 SSVEP-based BCI

The interest in developing a SSVEP-based BCI started back in 1970s with the work of Jacques Vidal and David Regan, according to [11] and [36]. However, the first application of SSVEP in actual BCIs dates back to around 2000, by Cheng and Gao ([39]) and Middendorf ([40]). The general idea behind these works is to represent the possible user commands as distinct flickering stimuli at different frequencies. The user selects one of the commands by gazing the corresponding flicker and the system infers which stimulus the user selected by analyzing EEG containing the generated SSVEP. This approach led the majority of SSVEP-based BCI proposed since then.

SSVEP-based BCIs have been widely studied since they present some serious advantages over the other previously introduced paradigms. First, being SSVEP driven by an external source (the stimulus) it provides more robustness and stability than internally modulated phenomena. Indeed, SSVEPs have been demonstrated to be robust to noise and artifacts. Second, as SSVEPs are unconscious reflexes of the visual cortex, the user does not need any preliminary training in order to user the BCI. Moreover, some systems have been developed that offer a high number of user commands; for example, up to 12 in [41].

Of course, SSVEP has some weaknesses too:

- A prolonged use of the BCI is likely to result in eye fatigue and tiredness.
- Most SSVEP-based BCIs are *dependent*, that is, depend on the control of eye gaze direction. Nevertheless, recently some studies have proposed innovative technique to realize *independent* SSVEP-based BCIs (see [42] and [43]).
- Flickering stimuli may be dangerous for individuals with photosensitive disorders.

Brain-Computer Interfaces employing SSVEP have been developed by different research groups around the world, the most important ones being Gao's

group at Tsinghua University, Beijing (China) and the group in Bremen University (Germany).
At Tsinghua University the pioneer work presented in [39] was the first of several results. In 2003, they applied SSVEP-based BCI to environmental control through an infrared remote ([12]). Thereafter, the group focused on common properties of SSVEP in order to develop a practical BCI usable by most people. This work is presented in [38] and it has been tested with 11 individuals with disabilities. Moreover, in 2008 the group launched a new SSVEP recognition method, based on statistics, that proved itself to outperform state-of-the-art methods [44]. Finally, they implemented this method in an online BCI [45].
At Bremen University the first results were presented in 2007. In [46] they modeled the SSVEP response and proposed a novel technique for detecting SSVEP using multiple channels. Later on, an online BCI that requires no training was developed and it demonstrated high performance with 37 subjects, out of which 8 with disabilities ([47]).

Even though each research line has its own peculiarities, we can identify some general issues and common aspects. Such aspects are analyzed in the following.

Visual stimulation

SSVEP response strongly depends upon the characteristics of the visual stimuli. A survey of the stimulation method can be found in [48]. Commonly used stimuli fall into three categories:

- *Light* stimuli employ light sources, such as LEDs, modulated at a certain frequency.
- *Plain* stimuli consist of some graphics (i.e., rectangle, arrow, or letter) that appears and disappears from a fixed background at a certain rate.
- *Reversal* stimuli are obtained by the alternation of graphical patterns, the most common being checkerboards.

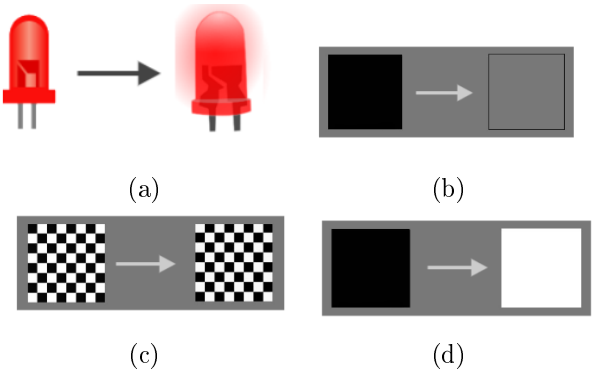


Figure 2.6: Examples of stimuli used to elicit SSVEP. (a) light stimuli. (b) plain stimuli. (c) and (d) two examples of reversal stimuli.

For each type, size, color and contrast may vary. Plain and reversal stimuli are usually colored in black and white. Figure 2.6 shows examples of different kinds of visual stimuli.

There is no common agreement in literature about which kind of stimuli elicits a more pronounced SSVEP. In fact, a direct comparison of the performance of different systems is often challenging due to the many external variables (other than stimulation) that may affect BCI's performance. Light stimuli are generally thought to be more effective than the others [48]. Yet, they are more difficult to implement since they need extra hardware, e.g., various LEDs and some device for synchronizing the stimulation. On the contrary, both plain and reversal stimuli are rendered on a computer screen.

Assuming that a monitor is available, plain and reversal stimuli only rely on software development. It is important to note that plain stimuli elicit an SSVEP at the frequency of one full cycle, i.e., the inverse of the period corresponding to one episode of graphic appearance and disappearance. This frequency is measured in Hertz (Hz). Conversely, reversal stimuli elicit an SSVEP at the frequency of one pattern alternation (measured in *reversal per second* or rps). While some studies stated that reversal stimuli can produce a stronger SSVEP [36], others affirm exactly the opposite. In [49], a study involving 6 subjects found a statistically significant increase in SSVEP detection using a plain stimulus with respect to checkerboard.

Hence, no general conclusion can be drawn about the impact of stimuli type on SSVEP intensity; color, frequency and size of the stimuli influence SSVEP response as well. For what concerning the color of the stimuli, colored squares or checkerboards (for example green and blue) provide high comfort-performance ratio, so as in [50]. Although, colored stimuli raise the risk for photosensitive epilepsy convulsions [51].

In this work, only black-white plain stimuli have been considered. For their design, the approaches described in [52] and [53] have been followed. Further details are provided in 5.2.1.

Stimulation frequency

The choice of the stimulation frequencies is one of the most important problems when dealing with SSVEP-based BCIs. As briefly stated before, the stimulus frequencies used in SSVEP research fall into three bands: low (less than 12 Hz), medium (12-30 Hz) and high(40-60 Hz). These frequencies sometimes overlap with the alpha cerebral rhythm (α, 8-13 Hz), that is, a natural periodic brain activity present in visual cortex, similar to those introduced in Section 2.4.1.

There is no consensus about the optimal frequencies for SSVEP-based BCIs. Since this is very individual dependent, a subjective calibration is sometimes employed. In [54], the user undergoes a screening session in order to select the most suitable frequencies in 5-30 Hz range. Still, many studies try to use fixed frequencies and avoid any training or calibration period. In [38], two criteria for selecting the stimulus frequency are reported: avoiding false positive; efficiency of frequency detection. Choosing the stimuli frequencies within the alpha band may lead to many false positives, that is, spontaneous EEG activity that triggers undesired commands. At the same time, the chosen frequencies should elicit an efficiently detectable SSVEP. For some subject this problem is not trivial, since the majority of EEG energy belongs to the alpha region.

SSVEP amplitude is often larger in α band, near 10 Hz ([48]). Several studies adopt frequencies in this band. Nevertheless, the spontaneous α rhythm

in EEG signal may likely produce false positive, that is, commands triggered without user intention. To avoid frustrating the user, it is necessary to avoid as much as possible false positives.

Spectral power of EEG decreases with frequency. Therefore, the response to high frequency stimulation has less absolute power than the response to low-frequency one. Signal-to-Noise Ratio (SNR) is frequently the preferred metric for measuring the effectiveness of a certain stimulus frequency. In general, most BCIs use frequencies in low and medium range, although the chosen values vary sensibly. Examples are [38], [46], [47]. In [55], the authors stated that the optimal frequency range is 12-18 Hz, followed by 8-14 Hz. However, there are cases in which higher frequencies are employed with good results, e.g., in [44].

This work was inspired by previous research on SSVEP presented in [21]. In the cited work, the stimulus frequencies adopted were reversal stimuli at 5.6, 6.4, 6.9 and 8 rps.

Signal processing

The essential role of signal processing in SSVEP-based BCI is basically to detect the existence of SSVEP in EEG signal and to determine the corresponding dominant frequency. The dominant frequency should be the current stimulation frequency, that is, the one the individual is focusing on. The stimulation induces a peak in EEG power spectrum corresponding to the dominant frequency, which normally is more pronounced than the peaks at other frequencies. In general, in order to identify the frequency sought two types of features are exploited:

- **Frequency domain features**. Such features are extracted by the frequency representation of EEG signal, namely the EEG power spectral density (PSD). In some cases, examining the power content at the stimulus frequency and its harmonics is sufficient for the BCI to detect the right command. Generally, however, the PSD is less explicit for the

presence of much background cerebral activity. PSD analysis will be described in details in Section 5.2.3.1.

- **Time domain feature**. Features extracted in the time domain capture EEG signal amplitude over time. For example, we can compare the signal amplitude with a reference signal pulsing at predetermined frequency. In this way it is possible to better isolate the SSVEP component sought. Further explanation in Section 5.2.3.2 and 5.2.3.3.

System architecture 3

This chapter regards the overall architecture of the assistive system.
The first section presents a global framework and a functional model around which the system has been designed.
The second section is about the process of designing and engineering the system and its components.
Finally, the third section gives a reader an overview of the means for evaluating the performance of the system.

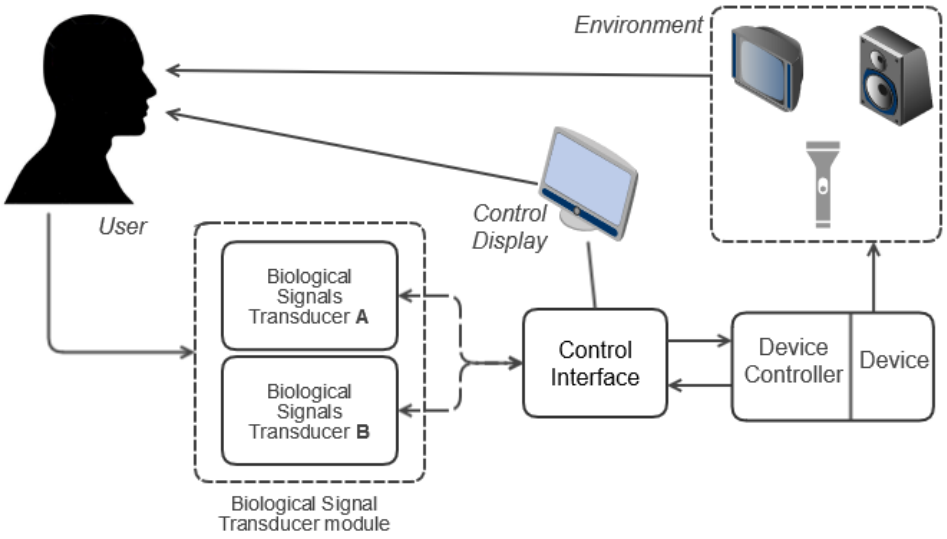


Figure 3.1: Functional model of the multimodal control system.

3.1 General framework

Assistive systems for HMI based on biosignals were proposed with various techniques and applications. Despite each work presents unique properties, most of them fit into a common framework. SMAD adopted the framework depicted in Figure 3.1, and adapted from the one proposed in [34] for BCI systems.

Observing the model, a loop structure can be recognized. It starts from the User, whose biosignals are the primary input, and it ends with the Environment that is affected by the actions of the system. Along this path, three main modules can be identified: the *Biological Signal Translator* module, the *Control Interface* module and the *Device Controller/Device* module. Eventually, the loop is closed through user feedback of different kinds. The communication between the modules may be bidirectional, for a module to know the outcome of a command, but the information flow within the system is not supposed to go the other way round, i.e., from the Devices to the User.

The first module, hereafter called Biological Signals Transducer (BST), acquires biosignals and translates them into an output signal. In Figure 3.1

it appears as a black box. Usually a BST comprehends many sub-modules. However, looking at this module independently of its internal organization turns it unconstrained from the kind of biosignals it uses. As it can also be seen from Figure 3.1, various possible BSTs may be used interchangeably. In fact, in [34] has been pointed out that this component may be replaced in systems other than BCIs, e.g., by a EMG-based one. This leads to the possibility of directly compare various BCI approaches, distinct biosignals based controls (such as EMG or EOG) or even a combination of different channels as in [18]. Furthermore, the BST usually provides device-independent output.

The second module, called Control Interface (CI), is responsible of translating the BST output into semantic control signals significant for the desired aim. This module is independent of the type of BST: as long as logical output are provided, it is able to translate them. Building the system in such a semantic decoupled way leads to several advantages in an assistive application. First, the possibility to adapt the controlling paradigm according to the degree of disability of the individual. Second, a well-designed CI would let the community objectively compare existing and novel technologies and help the progress of the field. Nevertheless, while a lot of research focused on improving BST control techniques, little attention has been paid to methodologies and good practices for designing such Control Interfaces.

The CI often holds knowledge about the application domain and context; for example, natural language processing software for spelling systems [15] or trajectory planning for wheelchair autonomous navigation [56]. It is important to take it into consideration while assessing system performance; further explanation is provided in Section 3.3. Finally, the CI may be equipped with a Control Display. The role of the Control Display is to exhibit a Graphical User Interface (GUI) showing the Control Interface current state, that is, the user feedback about the performed actions within the context of interest, e.g., environmental control.

The Device Controller is responsible for carrying out the commands received from the CI by translating them into physical signals required to operate

the Device and the Environment. The Device Controller is not aware whether the control commands has been provided using assistive or traditional input devices; nevertheless, its design shall be affected by the peculiarity of assistive technology. For example, the Device Controller for a biosignals-controlled wheelchair should avoid sharp turns and high speed. The Device is constituted by the necessary hardware to physically achieve the desired effect. Different works have used specialized hardware such as wheelchairs, or general purpose computers.

Figure 3.1 shows two feedback paths within the architecture that allow the user to monitor the system state. The Environment state is fed back to the user through sensory channels and it is compared with his or her intention. The Control Display, as well, provides a visual feedback about the Environment state in order to facilitate user interaction with the system. In this way, the model encourage the so-called *biofeedback*. Through biofeedback the individual is able to learn how to adapt the physiological activity for the purpose of reaching the desired goal. Therefore, it supports the user in developing and maintaining a new skill: the proper control of specific electrophysiological signals.

Finally, in Figure 3.2 the reader is given a picture of the robotic wheelchair the user will sit on while using the system. The manufacturer (Freedom[1] company) furnished the wheelchair with a joystick-based navigation. A PC has been embedded just behind the seatback, connected to a LCD display located in front of the user. Both BST and CI modules will run on the PC. An Arduino Mega[2] board is connected to the computer through USB port and will be responsible of wireless communication with the environment. Device Controller module will run on the Arduino board. In the following more details about this module are provided.

[1] `www.freedom.ind.br/`
[2] `http://arduino.cc/en/Main/arduinoBoardMega`

FIGURE 3.2: UFES robotic wheelchair. The Freedom wheelchair has been equipped with a computer behind the seat and a touchscreen monitor mounted on an aluminum structure.

3.2 Engineering the system

A software engineering approach drives the system design, with the aim of better understanding the overall system behavior and modular structure. Such an approach helps to develop complex system, by unambiguously defining the responsibilities of different components and how they interact with each other. As an important advantage, subsequent modifications of the system will take sensibly less time and effort. It is important to remark that this practice is uncommon in the field of assistive system development. Although assistive systems are unlikely as complex as large IT systems (e.g., enterprise application software), proper design considerations shall be an essential starting point. Instead, very often, assistive systems are monolithic or divided into components whose role is loosely analyzed. An example of that is the lack of well-established packages that can be easily integrated into brand-new sys-

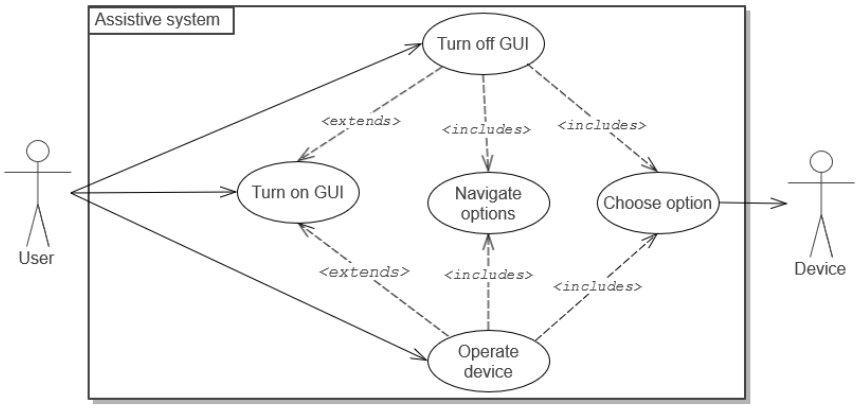


Figure 3.3: SMAD Use Case diagram.

tems with minimum effort. A remarkable exception in BCI field is BCI2000 project[3].

The first aspect under investigation is requirements engineering. This phase addresses the elicitation and specification of user requirements, that is, functions or capabilities that the system should provide to the user. The result is showed in Figure 3.3 as an UML Use Case Diagram. The User, i.e., the individual with disabilities, is associated with three use cases: Turn On Graphical User Interface (GUI), Turn Off GUI and Operate Device. The first two use cases are related to the control paradigm: as introduced in Section 2.3.1, the system-paced control paradigm implies a modality for the user to turn on and off command triggering. Although the GUI is just a visual feedback mean, the reader can consider it as tied to the user intentional control: a command can be issue only when the GUI is actually displayed, and conversely. Therefore, turning on the GUI is implicit in the other operations; this is the meaning of the *extends* notation in the diagram.

It is immediate to see from Figure 3.3 that two use cases *include* other use cases; for example, the execution of Operate Device contains as a part of it the execution of Navigate Options. This derives from the nature of the Control Interface, given that it implements option scanning translation. In order to operate a device, the user needs to navigate through the menu options until

[3] `http://www.schalklab.org/research/bci2000`

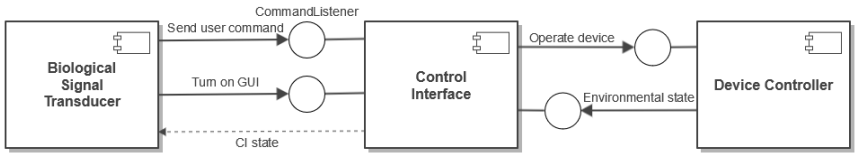


FIGURE 3.4: SMAD component diagram. The interface symbols represent the way different components interacts with each other. The dotted line means that CI module is just returning data after some interface method has been called.

reaching the desired one and then confirming the intention by choosing the option. As the diagram depicts, only when the user chooses an option the operation will affect a device in the external environment. The same goes for turning off the interface, since it is merely one of the possible options within the menu. On the other side, the user needs no other operations to turn on the GUI; this is due to the fact that, when the GUI is hidden, the normal command triggering is inhibited. Thus, the BST is responsible for providing a direct method for the user to turn on the GUI without relying on the menu.

We have presented the user requirements of SMAD; now, the focus turns on the components that make up the system and the interfaces between them. Since the attention is on the assistive system, the user and the external environment, including the devices, are leaved out of the analysis. As introduced while presenting the general framework, the main components are three: BST module, CI module and Device Controller module. Figure 3.4 depicts the interaction between them.

BST module processes user biosignals through a classification approach. When the pattern corresponding to *Turn on GUI* is identified, BST module calls the proper procedure exposed by the CI interface. Analogously, in the case of patterns corresponding to user commands, e.g., navigate or choose menu options, BST module produces a logical value that is passed to CI through the *CommandListener* interface. It is worth noting that this interface is mono-directional: it does not allow for CI module to start communicating on its own with the BST. Nevertheless, BST module can get information from the

CI as values returned by CommandListener procedures; this is particularly useful to synchronize system behavior. CI module interprets the logical values received from the BST. When the received value specifies to actually operate a device, the CI will pass the command on to the Device Controller, specifying some identifier of the device the user intended to operate. This communication channel is bi-directional (see Figure 3.4): Device Controller module can autonomously start sending data to the CI. Such a design choice enable the Device Controller to update the CI on the environmental state. For instance, the Device Controller may signal that a device is out of order and the CI would disable the corresponding option.

In the following, the three components are individually examined more in-depth.

Biological Signal Transducer

In this thesis two distinct BST modules are proposed, employing different modalities of control by means of biosignals. Hence the definition of *multimodal* system. Yet, it is not possible to use the two approaches simultaneously. The system works with one BST at a time; it is required to choose the most suitable modality according to the degree of disability of the user. The first BST module processes EMG and EOG signals with the aim of identifying determined facial expressions and eye movements. The second BST module is a proper BCI control: it deals with EEG recording of cerebral activity. They are treated in details in Section 5.1 and 5.2, respectively. For both modes, the BST uses a classification approach and supplies the CI with a logical value representing the detected class. Both BSTs produce same logical values for commanding the CI, thus the choice of which BST to use is transparent for the CI.

The interface between BST and CI is implemented by means of a Remote Procedure Call (RPC) protocol. Therefore, it creates total decoupling between the two components: they can communicate disregarding implementation details. For example, if a BST has been realized in MATLAB while the CI uses

Python, the only required effort to enable communication is developing a RPC client for MATLAB or using a software library.

Going back to the *Turn on GUI* use case, the problem of implementing a mechanism to turn on GUI and consequently intentional control, is difficult. It depends heavily on the kind of biological control the user is actually using. For example, [57] proposed a method when using EEG-based control. Hence, the development of a technique to use for turning on GUI rests on the type of BST used. Each module should implement a custom technique to address the problem. Instead, turning the system off will be one of the options the user can select while the system is running independently of the chosen BST.

Control Interface

The CI module receives control output signals from the BST and converts them into semantic control signals for the Device Controller. As well as for the BST, the control signals transmitted to the Device Controller may be continuous or discrete. Usually, even when commanding the continuous movement of some robotic device, the Device Controller works with discrete input, as in [13, 21, 16]. See [34] for more examples. Hereafter, we consider the CI to produce a discrete output for the Device Controller. This output is encoded as 3 characters long strings according to a protocol agreed with the Device Controller; further details in Section 4.3.

Furthermore, CI module drives GUI behavior. The GUI is displayed on the Control Display; in particular, the LCD display showed in Figure 3.2. The GUI structure is described in Section 4.1. Since the LCD display can be conceived as a part of the Device (the wheelchair), the Control Display is also a part of the Device. This setup has been pointed out to have some drawbacks (see [34]): the display may limit the environmental feedback and the user is dependent on the Device to present the GUI. However, the proposed setup is feasible with only one display and the mechanism that regulates the Control Display has been kept separated from the Device Controller. Therefore, looking at the Control Display is not unavoidable depending on the kind of BST used; for

example, the BST based on facial expressions proposed in this thesis can work even with no display. Yet, having a GUI is very helpful and convenient while using the system.

Device Controller and Device

The Device Controller module plays the role of receiving a command from the CI and operate the corresponding device. For the purpose of this thesis, the design and implementation of Device Controller module is treated only briefly, since this module was developed in other works of the research laboratory. It is the only component that does not run on the wheelchair's computer. Instead, it is actually made up of two Arduino boards. The first board is connected to the wheelchair's computer USB port. CI module interacts with this board through the USB connection, simulating a serial interface. Therefore, the channel is naturally bi-directional: both sides can start the communication, provided that the other side is listening. The second board is installed into an acrylic box which the devices are connected to. Each device is connected to the box through a socket; each socket has an associated electrical relay. The opening or closing of a relay implies turning on or off the matching device.

The two boards communicates via Radio Frequency (RF). When the first board receives a command, it passes it on to the second board that activates the corresponding relay, thus switching the state of a device. In addiction, the first board is equipped with an infrared (IR) transmitter. This is employed to operate devices such as television and air-conditioner. Figure 3.5 depicts the overall behavior of the Device Controller.

3.3 System performance evaluation

An essential ingredient for the progress of a research field is the adoption of commonly accepted, reliable and valid methodologies to evaluate the results. Within assistive technologies that allow HMI by means of biosignals, several factors make the assessment of performance more difficult. Hereafter, perfor-

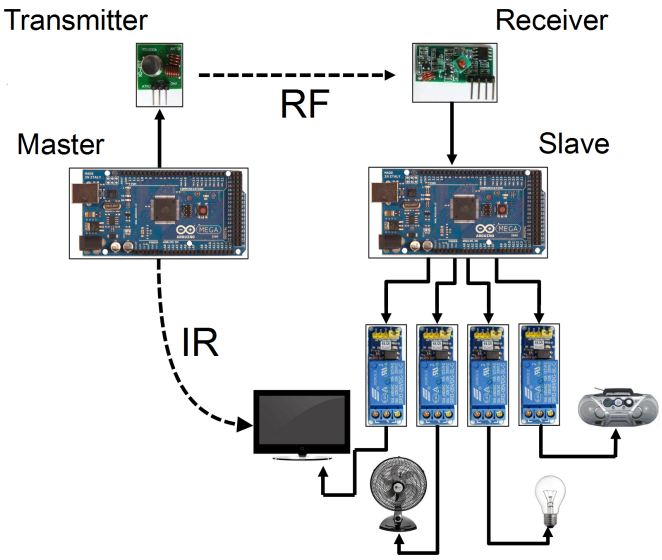


FIGURE 3.5: Device Controller wireless communication system. The master Arduino board is equipped with RF and IR transmitters, whereas the slave board has only RF receiver.

mance is intended not only regarding technical measures (e.g. speed) but also parameters relative to the overall user experience. Being still nascent technologies in the commercial and everyday life spheres, the diversity of approaches among research laboratories produce performance reporting far from uniform. Even using the same metrics, different assumptions may turn comparison between works cumbersome. The human factor involved in such systems (the *Human-* part of HMI) contributes to complicate the process of homogenization. When it comes to evaluate user experience, rating-based approaches are not standardized and open-ended questionnaires are more qualitative than quantitative.

Continuing with the idea at the basis of SMAD, some metrics are adopted that have been initially proposed for BCI systems, but can be easily extended to more generic BSTs. That makes comparison between distinct modalities of biosignal-based HMI, including BCIs, straightforward and immediate to understand. [58] offers a comprehensive tutorial on BCI performance metrics. In the following, the ones used in this thesis are analyzed in details.

Confusion matrix and accuracy

Confusion matrix is a classic way of describing the result of a multi-class classification problem. A confusion matrix is a square matrix that report counts for the true classes on the rows and for the predicted classes on the columns. The elements of the matrix can represent absolute counts or percentages of examples assigned to a certain class with respect to the total number of examples.

The element n_{ij} in the matrix indicate how many samples of class i have been predicted as class j. Thus, diagonal elements n_{ii} represent the number of correctly classified samples, while elements n_{ij} represent samples of class i that have been incorrectly classified in class j.

The most commonly used metric resulting from confusion matrix is the *classification accuracy* (p), defined as:

$$p = \frac{\sum_{i=1}^{M} n_{ii}}{N} \tag{3.1}$$

where M is the number of classes and N is the total number of samples.

Using classification accuracy presents some shortcomings. It does not take into consideration classes occurring with different frequencies: less frequent classes will weight less in the final result than more frequent classes. Furthermore, accuracy does not allow the comparison of classifiers with a different number of classes. This is due to the bias related to chance accuracy: a classifier with 2 classes has 50% probability of correctly classify a sample by chance, while a classifier with 3 classes has only 33%.

Cohen's Kappa coefficient

Cohen's Kappa coefficient (k) is a way to express the degree of agreement between classification. It is computed as:

$$k = \frac{p - p_e}{1 - p_e} \tag{3.2}$$

where p is the accuracy computed as in (3.1), while p_e is the chance accuracy computed as:

$$p_e = \frac{\sum_{i=1}^{M} (\sum_{j=1}^{M} n_{ij} \sum_{j=1}^{M} n_{ji})}{N^2} \tag{3.3}$$

Equation (3.3) can be seen as the sum of the matrix by row times the sum of the matrix by column, divided by the total number of samples.

Cohen's Kappa is equal to zero when the accuracy matches the chance probability, while it is 1 with perfect accuracy. Although it solves the problem of dealing with classifiers with different number of classes and it provides more sound estimation of classifier performance, Kappa coefficient is much more rarely used than accuracy.

Information Transfer Rate

Information Transfer Rate (ITR, sometimes called bit rate) is a performance measure widely used in BCI literature, proposed in [11]. It comes from Shannon's information theory, in particular from the concept of mutual information ([59]). ITR naturally applies to discrete BCIs; nonetheless, there is no theoretical obstacle to generalize its application to discrete BSTs. Simplifying mutual information computation, the formula for ITR was defined as:

$$B = \left(\frac{1}{c}\right) \left[\log_2 M + p \log_2 p + (1-p) \log_2 \left(\frac{1-p}{M-1} \right) \right] \tag{3.4}$$

where c is the time per trial, p is the classifier accuracy and M is the number of possible choices, that is, the number of logical values the BST produces.

ITR has been proposed as a metric to turn performance reporting of BCI systems uniform. In fact, the large majority of works in BCI field reports the achieved ITR. Nonetheless, it suffers from a variety of drawbacks. Firstly, since it focuses on BST output, it does not consider how CI behaves. Thus it assesses only BST performance. Furthermore, as pointed out in [60], mutual information - and so ITR - is a theoretical limit for the considered channel. For the actual channel throughput to reach this limit, in telecommunication complex techniques have been devised, none of which applicable in the domain of

HMI by means of biosignals. Therefore, ITR may be unrealistic for measuring the real performance of a practical BST.

Utility

As well as for ITR, the Utility metric has been proposed, in [60], within BCI field of research with the name *BCI Utility*. However, being discrete is the only property required for the BCI transducer. Thus, we can naturally extend the concept of Utility to a generic discrete BST. The aim of this metric is to compute the average benefit the user achieves using a given system. Noteworthy, the focus is on the entire system, not only on the BST: computing the Utility depends on both the BST performance and the CI behavior. In [60], Utility is defined as the expected average benefit over time:

$$U = \mathbb{E}\left[\lim_{T\to\infty} \frac{\int_0^T b(t)\mathrm{d}t}{T}\right] \tag{3.5}$$

where $b(t)$ is a benefit function. $b(t)$ assumes positive (or negative) values depending on whether the output at time t matches or contradicts the user intention. For a discrete BST, (3.5) reduces to:

$$U = \frac{\mathbb{E}\left[b_k\right]}{\mathbb{E}\left[\Delta t_k\right]} \tag{3.6}$$

where b_k is the benefit corresponding to the kth output and Δt_k is the time passed since the previous output. This formulation can be interpret as the average benefit brought by any correctly emitted output divided by the expected time needed to emit it. Note that benefit is related to the system output, not the BST's. Therefore, Utility is maximum for systems that produce actions conform to user intention in the shortest interval of time.

In practice, computation of U demands the definition of two quantities: benefit and time. The benefit term offers flexibility, because the authors are free to define the most appropriate criterion to express and compare their results. In [58] two examples are provided:

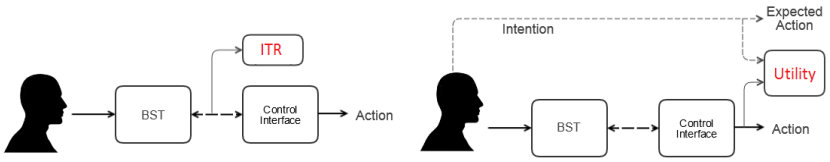


FIGURE 3.6: Difference in approach between ITR and Utility. While the ITR measures the capacity of the BST–CI channel, the Utility assesses the agreement between the action performed by the system and the one the user is expecting.

- A correct output could product a positive unitary benefit (+1). Therefore, U would measure the average number of correct outputs over time, expressed in 1/time.
- In order to compare Utility with ITR, one could quantify benefit as the information conveyed when a correct output is produced. In this case, U has the unit of bits/time.

On the other hand, the time term is strictly tied to the BST performance and the CI strategy. Therefore, it helps aggregate the performance of the system in a consistent and realistic way. Being the definition of this term CI-specific, it needs to be derived in a closed form; Section 4.2 provides the derivation for the CI used in this work.

Finally, Figure 3.6 tries to illustrate the difference between Utility and ITR. While ITR just assesses the capacity (not the actual throughput) of the channel between BST and CI, the Utility introduces into discourse the user intention and how the system is able to support it. Therefore, a fast and precise BST could have extremely high ITR, but would not be optimal for the user if the CI was badly designed. Therefore, Utility gives the researchers a mean to consistently drive the design of the entire assistive system and to optimize its performance.

Control Interface 4

This chapter focuses on Control Interface module and how it interprets user commands to produce actions that affect the home devices.
We start describing the main premises behind CI design. The section shows the GUI displayed to the user and the different types of menu options.
The second section explains the formal reasoning that has driven CI design, in order to optimize system performance. Then, we give some details about the main implementation choices that leave CI module easy to customize and adapt.
Finally, the last section is about the importance of employing a common CI to turn the evaluation and comparison of different BSTs easier.

4.1 CI module and GUI

The Control Interface module fulfills two main functions: it acts as a bridge between the BST and the specific application, e.g., the environmental controller, and it provides the user with a graphical feedback (GUI) about the system state. The design of the CI was based on two fundamental principles:

P1. The user should be able to realize every possible task, i.e., operating any device and turn off the system, with a very limited set of logical commands.

P2. The interface behavior plays an essential role in determining the proper trade-off between performance and number of commands.

In order to fulfill P1, in SMAD we opted for an option scanning-based strategy. The choice of this approach allows to create a semantic mapping between logical control signals from the BST and semantic control signals that are meaningful for the Device Controller. An example is translating the same *Operate* logical command into semantic commands such as *Operate lamp* and *Operate radio*. This choice has justifications from both user and technical points of view. For an assistive system to be really valuable, it needs to be easy to use by technically non-skilled individuals. Keeping low the number of possible tasks helps the user to minimize fatigue and confusion between them. Furthermore, a key aspect of control system based on biosignals is the user adaptation to the system, that is, how the user develops better ways to perform the possible tasks. This adaptation will be probably simpler as long as the number of tasks is limited. Finally, the pattern recognition techniques used within BST module are likely to ensure more reliability with a small number of classes. On the other hand, a menu has some performance drawbacks that are needed to be taken into consideration. Intuitively, the bottleneck is represented by the option farthest from the one initially selected.

Figure 4.1 offers an example of the GUI, as shown in the wheelchair display. The currently selected option is highlighted in the center, while the other pos-

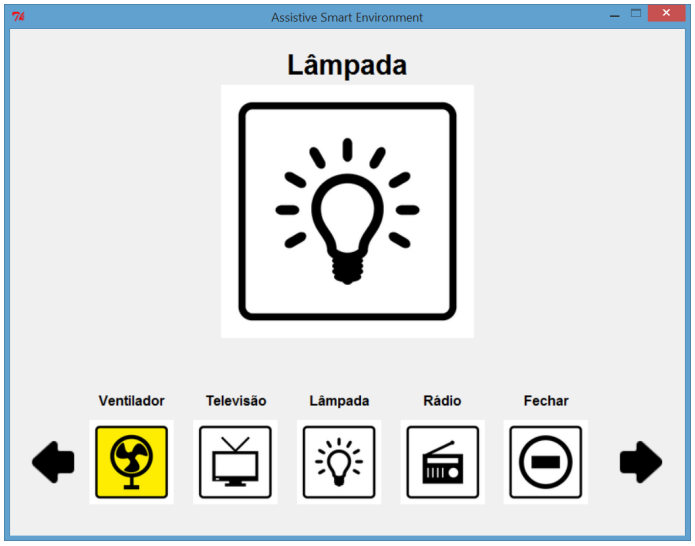


FIGURE 4.1: Screenshot of the GUI.

sibilities are listed below. Just one option can be highlighted, hence operated, at a given time. A kind of circularity is implicit in the menu: from the last option (the fifth) it is possible to navigate to the first option with one step, and the other way around. When an option is highlighted, the user has one command available to select it. Triggering that command means to switch the state of the corresponding device. The GUI notifies the user about the devices currently turned on. For instance, in the situation described by the screenshot above, the fan has been turned on and then the user has navigated up to the lamp.

There are three kinds of options:

(a) Options that, when selected, produce a switch of the device power state.

(b) Options that, when selected, produce an action independently of the previous state of the device.

(c) Options related to operations that affect no device but the GUI.

The fifth option shown in Figure 4.1 is an example of type *c*. It is related to the operation of turning off the GUI. By operating it, the user will hidden the window. The GUI will remain hidden until a *Turn on GUI* command is triggered again by the BST.

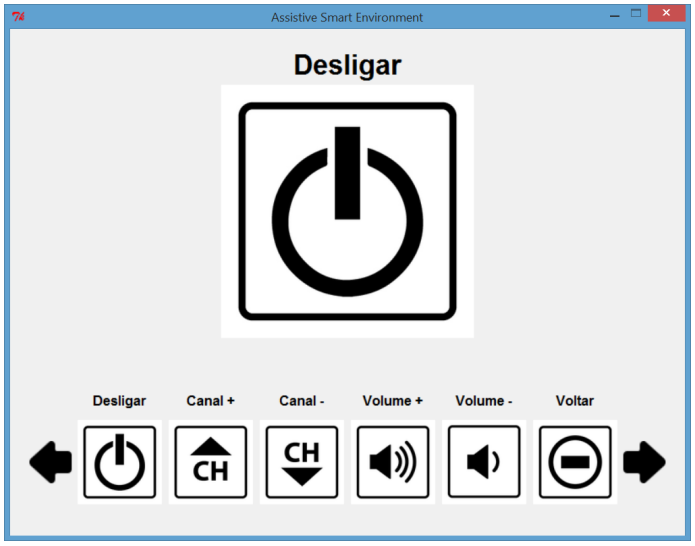


Figure 4.2: Screenshot of the TV sub-menu.

The presented menu is not flat: some option may have a sub-menu coupled with its activation. A sub-menu has the same appearance and behavior as the main menu. Moreover, there is always the possibility of going back to the main menu. In Figure 4.2 the sub-menu linked to the television is displayed. The options contained in this menu are mostly of type *b*. They generate actions that alter television state, e.g., by changing the channel, producing the same effect whatever the previous state was.

4.2 Design based on Utility

A crucial step when optimizing CI performance is to find out the optimal number of logical commands, which are the input for the CI. This choice affects the way the User navigates the menu. An original contribution of this work is to employ Utility metric (see Section 3.3) to drive this part of the CI design. Three interaction approaches were compared:

Mono-directional navigation Only two logical commands are provided: one to choose to operate the currently highlighted option (*Do* command) and one to navigate to the next option, e.g., rightward. (*Next* command).

Reversible navigation In addiction to the two command described above, an *Undo* command is provided to cancel the last action performed and restore the previous state.

Bi-directional navigation The mono-directional navigation is expanded with a third command (*Previous*) to navigate to the previous option, e.g., leftward.

As previously introduced, Section 3.6 defines Utility for a discrete BST. By using that definition, the Utility for SMAD is obtained as the average benefit b_d brought by any correctly operated device divided by the expected time T_d needed to operate it:

$$U = \frac{b_d}{T_d} \tag{4.1}$$

Similarly to [60], the following assumptions were considered to simplify the derivation:

A1. The accuracy p of the BST is constant over trials.

A2. Each trial is independent of the previous one.

A3. The different devices have equal probability of being operated.

A4. For all options, the *Do* command switches the state of the corresponding device. For instance, if the selected device is off, a *Do* command will turn it on, and vice versa.

In order to be comparable with classic ITR, the benefit is measured in terms of bits of information. Supposing the CI exhibits N options, under A3 the information carried by the device choice will be $b_d = \log_2 N$ bits. It is worth noting that ITR is not affected by the characteristics of the CI. Therefore, there exists a remarkable difference between Utility and ITR in this case, because the ITR only depends on the number of BST logical commands, which in general will be different from N. Hereunder the derivation of Utility in closed form is provided for the three navigation approaches presented.

Let's focus first on the time T_c needed to carry out a single command. Normally it would be equal to the duration of a trial. Nonetheless, for each trial, it may be the case that the BST misinterprets the signal producing the

wrong command. This happens with a probability $(1-p)$. When it happens, T_c will be longer because the user has to both recover the error and give the desired command again. Assuming c the duration of a single trial, with an accuracy of p, the expected time T_c needed to complete a single command becomes:

$$T_c = pc + (1-p)\left(c + T_r + T_c^{(1)}\right) \tag{4.2}$$

where T_r is the expected time to recover an error. Under bi-directional and reversible navigation approaches, it is easy to see that only one additional command is required to undo the erroneously classified one if A4 holds. In fact: *a)* supposing bi-directional approach, an additional *Do* recovers an erroneously performed *Do*, while a *Previous* command restores an erroneously performed *Next*, and vice versa; *b)* supposing reversible approach, all the user needs to do to recover an error is to perform an *Undo* command.

Therefore, considering A1 and A2, we have $T_r = T_c^{(1)} = T_c$. This leads to:

$$T_c = c + (1-p)\,2T_c \tag{4.3}$$

The actual time T_d to operate the desired device is related to the distance d between the actual highlighted option and the desired one. To operate once the desired device, the user needs to achieve d *Next* (or *Previous*) commands and one *Do* command, leading to $d+1$ total trials.

$$T_d = (d+1)\,T_c = (d+1)\,[c + 2\,(1-p)\,T_c] \tag{4.4}$$

Let's write equation (4.4) in iterative formulation

$$\begin{aligned} T_d &= (d+1)\,[c + 2\,(1-p)\,T_c] = (d+1)\,[c + 2\,(1-p)\,(c + 2\,(1-p)\,T_c)] = \\ &= (d+1)\left[c + 2\,(1-p)\,c + 4\,(1-p)^2\,T_c\right] = \\ &= (d+1)\sum_{i=0}^{\infty}(2-2p)^i \end{aligned} \tag{4.5}$$

this series converges to

$$T_d = \frac{(d+1)\,c}{2p-1} \tag{4.6}$$

if $p > 0.5$. If this condition is not met, the expected time to correctly operate a device goes to infinite and the system is not usable. From the general formula of Utility and (4.6), and setting $b_d = \log_2 N$, we finally obtain

$$U = \frac{(2p-1)\log_2 N}{(d+1)\,c} \tag{4.7}$$

Note that, even if (4.7) holds for both bi-directional and reversible approaches, there still is a difference between the two regarding the value of d. In fact, under the former d would assume integer values in $[0, N/2]$ if N is even (or in $[0, N/2+1]$ if N is odd), whereas under the latter d will take values in $\{0, 1, \ldots, N\}$.

Going back to the mono-directional navigation approach (m), the time for command is as follows

$$T_c^{(m)} = c + (1-p)\left(\frac{N+2}{2} T_c^{(m)}\right) \tag{4.8}$$

due to the error recover time T_r. If the system triggered an unexpected *Do*, another *Do* is needed to recover (so $T_r = T_c$). However, if the system triggered by error a *Next*, the user would need to navigate the entire menu to come back to the desired option, leading to $T_r = N \cdot T_c$. As a result, $T_r^{(m)} = \frac{N+1}{2}$. Utility formula derives as follows

$$U^{(m)} = \frac{(4p-3)\log_2 N}{(d+1)\,c} \tag{4.9}$$

if $p > 0.75$, with d assuming values in $\{0, 1, \ldots, N\}$.

In order to compare the presented navigation approaches, an actual biological signals translator can be simulated with accuracy p to compute the Utility values for each strategy. Let's imagine an interface with $N = 6$ options; the trial duration c will be assumed unitary, so that the computed value can be immediately compared with ITR. As previously illustrated, the Utility is dependent on the desired option and the current distance d to it. Figure 4.3 shows this dependency. The three curves represent the three interaction approaches, respectively. In all cases the first option is currently highlighted, so the distance to it is zero. Depending on the position of the desired option

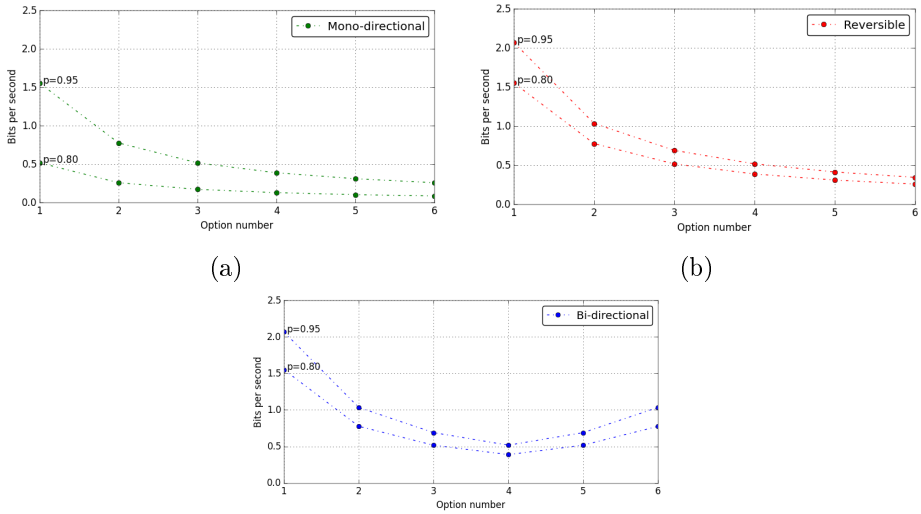


Figure 4.3: Relation between Utility and the position of the desired option. The option n. 1 is highlighted. The menu contains six options. Two curves are plotted for different values of accuracy. Diagram (a) shows the mono-directional navigation approach, (b) the reversible approach and (c) the bi-directional one.

(desired option n. 1, 2, 3, and so on), the value of Utility changes. Using mono-directional (Figure 4.3(a)) as well as reversible navigation (Figure 4.3(b)), d goes from zero to five; instead, introducing *Previous* command (Figure 4.3(c)) d goes from zero to three: the sixth option is at distance equal to 1.

A sensible decay in performance is observed when trying to operate farther devices, in terms of distance to option. Nevertheless, by allowing the navigation in reverse order, the bi-directional approach reduces the decay and the worst case happens for $d = N/2$. To show the global performance, hereafter the arithmetic mean of the Utility for the possible values of distance d will be considered as Mean Utility.

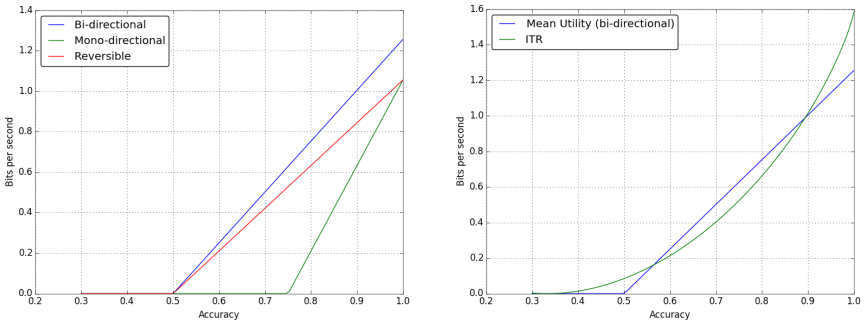
It is important to note that the recursion implied in (4.3) entails a threshold below which the system cannot be used. Figure 4.4(a) depicts the difference between the Mean Utility using the various navigation approaches. For $p < 0.5$, the output is more likely wrong than correct, hence, on every trial, the system would operate an undesired device or navigate in the opposite direction than

expected. With mono-directional navigation this aspect is even more relevant, since the Utility derivation entails that the system is usable (i.e., the series converges) only if $p > 0.75$. Figure 4.4(a) also shows that, as one would expect, mono-directional and reversible navigation approaches are eventually equivalent with a perfect accuracy because *Undo* command becomes useless. The bi-directional curve shows better Utility and suggests that performance will decay in a smoother way for individuals with less accuracy. This is the case in most situations when the error rate is significant, around 10-15%. Moreover, the great variability in performance among individuals often leads to observe poor accuracy for at least few subjects. In summary, bi-directional navigation has been chosen as menu navigation approach due to the better performance and still reduced number of logical commands.

Figure 4.4(b) helps to understand the difference between Utility and ITR. Let's focus firstly on the range of $p < 0.5$. The ITR formula returns a little yet positive value, indicating that a small quantity of information is expressed through the system. However, as demonstrated before, when $p < 0.5$ the system is not usable, since it will continue to misinterpret user intention. Secondly, ITR is linked to the number of classes the BST recognizes rather than the number of symbols the CI uses to encode its output. Therefore, the ITR will be limited to 3 user commands while the Utility consider the benefit brought by N devices.

4.3 Implementation

The CI module has been implemented keeping in mind some important aspects. The code shall be easy to develop and maintain. Being cross-platform would help the migration of SMAD on a new operating system. The GUI logic should be kept separate from communication logic and overall interface behavior. Finally, the importance of using free software was considered too, for the sake of simplifying system setup and relieve the user from expensive software licenses.



(a) (b)

Figure 4.4: Analysis of Mean Utility with respect to accuracy. In (a) the comparison between mean Utility using three navigation approaches for a menu with 6 options. In (b) the comparison between mean Utility, using bi-directional approach, and ITR.

The module has been developed using Python 2.7 following the object-oriented programming paradigm. Therefore, the portability is guaranteed on any platform the interpreter is available on and the code is easily maintainable and extensible. In Figure 4.5 a simplified class diagram is shown in order to describe the main entities. *Control* class is the central class. It starts the communication with the Device Controller (Arduino plate) through serial port. Then, it creates the main menu panel, launches the RPC server and it displays the GUI. The *GUI* class encapsulates all graphical logic, while *CommandServer* class encloses the RPC server management. It is worth focusing on the *Panel* class, the class that builds the logical structure of a menu panel, e.g., the main menu with fan, television, lamp and radio. Special attention has been paid to leave panel definition and modification as flexible as possible, so as such operations will not involve modifying the Python code itself.

During its initialization, the *Panel* class reads a JSON[1] file containing the description of existing panels. Each panel is identified by an unique id and it contains a series of items, that is, menu options. Options have in turn the following properties:

Mandatory

[1] `http://json.org/`

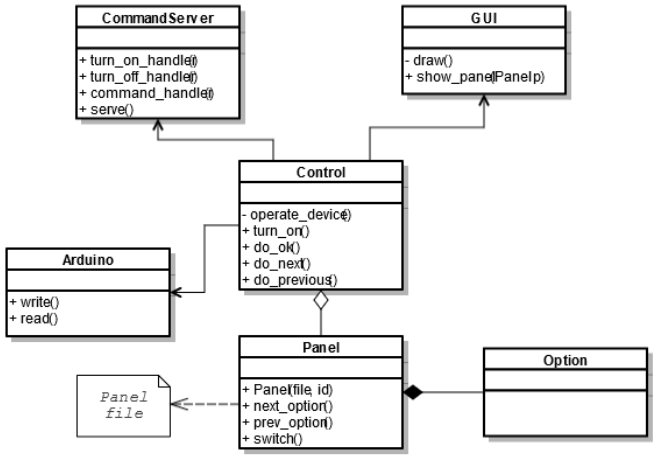


FIGURE 4.5: Class diagram of CI module.

- "id" : the unique option identifier.
- "name" : the label associated to the option.
- "imgOff" : the image displayed when the device is off (default at start).

Optional

- "imgOn" : the image displayed when the device is on.
- "deviceOperation" : what operation will be triggered on the corresponding device. This option may assume one out of the following values: *a)* "switch" if the device power state will be switched *b)* "device_on" if the device will be turned on *c)* "device_off" if the device will be turned off *d)* "device_modify" if the option modifies some properties of the device without affecting its power state. Selecting a "device_on" option on a device that is already on will produce no effect.
- "message" : a three char long string that will be transmitted to the Arduino plate, in order to recognize which device to operate.
- "panelId" : a panel identifier corresponding to the next displayed panel. This property is needed when the option will change the currently displayed panel, e.g., in the case of a sub-panel.
- "objectId" : an option identifier. This property is needed when the "deviceOperation" will affect an option defined in another panel. For

instance, when, from the television panel, the user wants to turn off the television itself, defined in the main panel.

- "turnOff" : associated with any value, this property means that, when selected, the option will hidden the GUI.

In Appendix A, the reader is give a concrete example of the JSON code that specify the main menu and the television sub-menu.

Finally, a global configuration file is provided to further facilitate the process of migrating the system on another platform. It sets resources path, RPC server host and port, and other global properties.

4.4 Common CI as a BST benchmark

CI module relies on RPC paradigm to guarantee complete decoupling between the BST and itself. Hence, a brand-new BST can be just plugged in to be used together with the existing CI. This property takes on great importance, essentially for one reason: it allows the comparison between BSTs. The progress toward actually usable human-machine interfaces through biosignals is obstructed by the difficulty in comparing different solutions. The variety of applications, experimental settings and metrics create a confused picture where lots of new system are proposed often without broadly recognized advances. Some attempts have been made in order to simplify the problem. For example, the ITR metric for BCI has been proposed as an application-independent objective measure for comparing different BCIs; yet, considering the drawbacks presented in Section 3.3, it did not totally fulfill its promise.

The concept of Utility metric helps to restate that for analyzing BST performance it is unavoidable to consider CI behavior as well. Indeed, the adoption of Utility to express the performance of BST and CI together would be of great value, since it is a realistic measure to summarize the usefulness of the system for the user. As stated, Utility assesses the performance of the tied combination of BST and CI. Increased Utility may come from an improvement to the BST accuracy or the CI strategy. Thus, the only way to reliably

compare distinct BSTs is by setting a fixed common CI. As proposed in [34], developing common CIs should have leaded to establish benchmarks for BST. Surprisingly, this did not happen, as no component has been proposed yet to meet this need. Several research fields have had benefit from standard benchmarks as an empirical method to ascertain actual improvements brought by novel works. Some guidelines are presented hereafter to frame the problem of designing such a benchmark.

How a benchmark for BST may be conceived? BSTs are automatic systems, but their users are an integral part of the final product. Therefore, two scopes need to be taken into account:

System performance Indicators of how well does the system support the user intention, e.g., accuracy or Utility. To relieve the dependency of this indicators from the current individual and make them express properties of the system itself, the measures shall be calculated with respect to several subjects and then averaged.

User experience Quantities related to the user experience while and after interacting with the system. An important example is user fatigue, which is a typical effect of biosignal-based HMI. Assessing such quantities is essential to get an overall evaluation of the assistive system, since they can sensibly contradict system performance metrics. For example, SSVEP BCI is considered the most performing BCI paradigm because of high bit rate with minimal training. At the same time, this paradigm is very demanding for the user: it provokes serious eye fatigue. Other usual examples of quantities regarding user experience are usability, user engagement, user work load. A system necessitating a long training or calibration session will cause more work load, whereas a BST with fast user feedback will likely be more engaging.

For the sake of comparison, quantitative benchmark components would aid unambiguous interpretation. System performance measures are naturally quantitative, while it is not the case for user experience ones. However, sev-

eral automatic techniques for quantifying user experience indicators through user biosignals have been developed. EEG signal has been used to accurately classify various cognitive states. Specifically, in [61] an algorithm has been proposed to quantify user fatigue while driving, by analyzing spectral features of EEG, in particular delta, theta, alpha and beta bands. By using thresholds on magnitude in such bands, the authors succeeded in classifying user state into early, medium and extreme fatigue phases. A similar spectral band-based approach has been used in [62] to find out an optimal index of user engagement by means of EEG signal. Four candidate indexes have been compared and one based on theta, alpha and beta bands showed to be highly correlated with individual task engagement.

In summary, the point has been made that a common CI could constitute the basis for an useful quantitative BST benchmark. By setting a series of goal to achieve using the CI, researches will be able to compare both scopes of BST evaluation. Moreover, applying novel EEG-based technique, automatic quantitative assessment of user experience may be coupled with system performance analysis, providing a more comprehensive picture of the degree of usefulness of the system.

Biological Signal Transducers 5

This chapter presents two BST modules that translate different biosignals into commands for the CI module.
The first BST considered is based on EMG and EOG signals. It interprets facial gestures and eye movements and maps them into valid user commands. The wireless headset called Emotiv EPOC, which has been used to record user biosignals, is subsequently introduced.
The second section illustrates the other BST developed for SMAD. This BST deals with user cerebral activity when the subject is gazing visual flickering stimuli. The BST module processes user EEG signal and extracts the current stimulus the subject is looking at.

5.1 BST module using EMG/EOG

A notable number of individuals with disabilities that affect motor skills preserves the control of facial muscles. Similarly, elderly persons may experience motor impairments, but still succeed in governing simple facial expressions. For such subjects a BST module is proposed that employs sEMG and EOG signals in order to detect facial gestures. The user shall perform one out of three possible facial gestures to trigger the corresponding command. Therefore, after the recording equipment installation on head, the user triggers command by muscle activation. Through an RPC client, the command is transmitted to CI module. The latter takes care of providing visual feedback by altering menu state.

5.1.1 Emotiv EPOC

Emotiv EPOC is a wireless consumer equipment for EEG recording that offers a Software Development Kit (SDK)[1]. EPOC consists of a plastic headset with flexible plastic arms that guarantee the right positioning of the electrodes. The electrodes are modular, allowing for fast placement. It makes no use of conductive gel: the experimenter just needs to drop saline solution on felt pads provided by the manufacturer. Figure 5.1 is a picture of the headset and the so-called *hydration box* for storing and wetting the felt pads. The headset communicates with a USB wireless receiver dongle, connected to the computer, using a proprietary encrypted wireless protocol.

The plastic arms help headset installation, albeit at the same time force the electrodes to fixed locations. There are 14 available channels, plus 2 CMS/DRL references. See `http://www.biosemi.com/faq/cms&drl.htm` for more details on these references. In Figure 5.2 the available sensor locations are depicted. It is very important to note that two alternative references are provided, in case the contact of the P3/P4 locations was poor. These alternative references are located just behind the ears, touching the mastoid bone. Although the

[1] `https://emotiv.com/epoc.php`

FIGURE 5.1: Emotiv EPOC headset. On the left, the electrodes inside the hydration box.

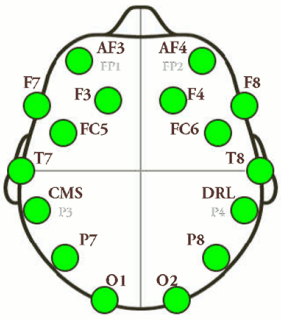


FIGURE 5.2: Emotiv EPOC electrode locations.

fixed locations limit the possible applications of EPOC, they are sufficient for recording facial sEMG and EOG, especially on AF3/AF4 and F7/F8 locations. The sampling rate is 128 Hz, even if internally the signal is sampled at 2048 Hz and then downsampled; the bandwidth is 0.2 - 45 Hz, with digital notch filters at 50Hz and 60Hz.

In addition to the EPOC headset, Emotiv supplies researchers with a complete SDK that offers a variety of functions based on the signals recorded on the scalp. The *Expressiv* suite concerns muscle control. This suite supports a range of face and eye movements including looking left/right, clenching, raising brow and blinking. The *Affectiv* suite reports changes in the subjective

emotions experienced by the user, in terms of Engagement, Instantaneous Excitement, and Long-Term Excitement. Finally, the *Cognitiv* suite processes cerebral activity to identify the user's intent actions over a physical or virtual object. The Emotiv Control Panel offers a practical GUI for the three packages. Concerning this BST, the focus is on the Expressiv suite. A prototype has been developed that relies on this suite for signal processing of sEMG/EOG.

Emotiv EPOC has already been used in BCI context due to the low-cost and ease of install. For instance, in [63] BCIs and EPOC are framed within the context of innovating computer game development. In [64], EPOC is used for a SSVEP BCI presenting satisfying results. Nevertheless, a series of factors hampers developing actual assistive technology exploiting this headset:

- Fixed electrode locations prevent some kind of BCI applications. For instance, a BCI based on motor imagery (see Section 2.4.1) could hardly achieve optimal performance without electrodes on primary motor cortex, such as C3 and C4.

- The equipment is subject to sensible depreciation. After an year (of intensive use) in the research lab, the four EPOCs purchased are well-thumbed. The gold plate electrodes are coated with is sensible to oxidization; this effect is mainly due to the salt present in the conductive solution. So, special care is needed, both for the headset and the sensors. At the same time, the plastic structure is quite fragile. Besides the risk of breaking the headset during installation, it becomes less close-fitting with prolonged use. This has a negative effect on the quality of contact, especially for sensors on temporal lobes (T7/T8).

- Given the relatively low sampling rate (128 Hz), it is often necessary to collect several seconds of signal in order to make pattern recognition reliable. Compare with most high-precision biosignals recording devices that reach 500-600 Hz: EPOC needs up to three times the time to collect the same amount of signal samples.

It is useful to bear these drawbacks in mind to better understand how to properly use such an equipment.

5.1.2 User tasks

As introduces, each of the three user tasks corresponds to a facial movement or expression. The three following tasks were defined: moving eyes horizontally (left/right), raising brow, prolonged clench. Among the expressions supported by Expressiv suite, these tasks guarantee high accuracy and are relatively easy to carry out; classification results are presented in Section 6.1.2. The user navigates through the options by raising brow and moving eyes, whereas the prolonged clench is used to act the highlighted option.

The system does not need to be trained on the specific user: the facial expressions are characterized by signal features quite similar among most subjects. Yet, each individual performs the three gestures in a subtly different personal fashion. Thus, to expect a new user be able to achieve an optimal performance from the first trial is unrealistic. However, our tests revealed that, for most users, just a very brief adaptation period was required to figure out the best form to perform the gesture. This adaptation period consists in 2 or 3 minutes when the user is trying to execute a command and observe a visual feedback, that is, the Emotiv Control Panel. The Emotiv Control Panel shows a sketchy 3D head reproducing the same gestures the user is actually performing; it is a sort of simulation of the actual use. Although this task is not required, it is recommended considering that it can highly accelerate the development of the user's control skills.

Emotiv's SDK performs signal processing and delivers logical values corresponding to the facial expressions. However, Expressiv suite does not treat the same way all the user tasks used. In the case of clenching and raising brow, a continuous value is produced indicating the degree of intensity of the movement. In the case of moving eyes left or right, the software supplies just a switch discrete value (0/1) for the left and one for the right. Therefore, recalling that BST module needs to transmit three logical values to the

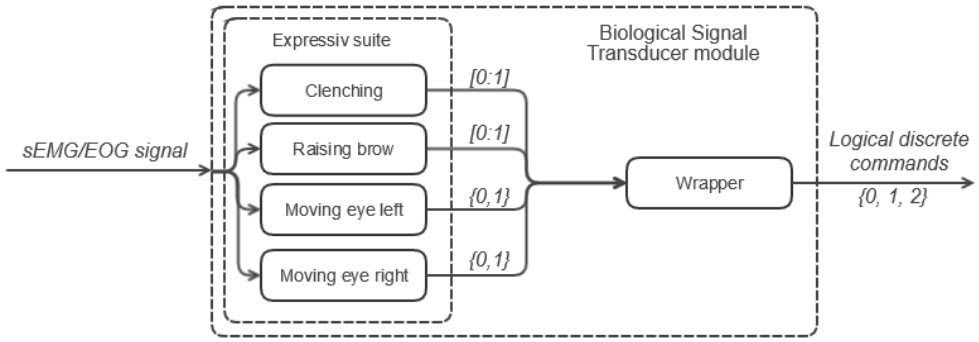


Figure 5.3: Functional structure of BST module using Emotiv's SDK. $[n:m]$ notation indicates any continuous value between n and m, while $\{a, b, ...\}$ means a discrete value belonging to that set.

CI, a software wrapper is needed in order to make BST output uniform. In Figure 5.3 the BST scheme is depicted. The wrapper could be developed by interfacing with the Dynamic Linked Library (DLL) included with the SDK. Alternatively, Emotiv furnishes an application, called *Emokey*, that allows to set thresholds on continuous values and combine facial expressions. It simulates key pressing generating as output keyboard characters. These characters can be easily converted into numbers to be passed through RPC to CI module. Since it is up to the Control Interface to correctly interpret these values, there is the necessity of a shared mapping between BST and CI. The mapping shall define the association between logical values and user commands consistently to the CI interpretation. For instance, both the BST and the CI need to agree whether the number 0 or 1 corresponds to the *Do* user command (see Section 4.2).

5.2 BST module using SSVEP BCI

For many individuals, the motor disabilities preclude voluntary control of facial muscles. For these users to use SMAD, an alternative BST module has been developed based on SSVEP BCI. As described in Section 2.4.2, SSVEP is a reactive response of the cerebral cortex to a visual stimulus presenting a recurrent pattern at a certain stimulation frequency. The BCI presented in this

thesis is based on the SSVEP BCI previously developed in UFES lab and presented in [21] for commanding a robotic wheelchair. Among the contribution of the cited work, the authors have provided an EEG signal database that stores SSVEP recording trials of 17 people, out of which 5 with disabilities. In this thesis, the EEG database has been exploited as a valuable tool for evaluating and improving signal processing techniques. The SSVEP recognition methods that perform better on the database can be indeed used for the actual BST module. Nonetheless, there are some differences between the data recorded in the database and the data collected for our BCI. Foremost, in [21] a medical device has been used instead of Emotiv EPOC. It offers higher sampling rate and a cleaner EEG signal thanks to the conductive gel. Secondly, in this work the number of stimuli has been reduced from four to three and the stimulation method have been changed. On the other side, the stimulation frequencies has been kept the same in order to compare the results.

From the definition of BCI follows that the user is not required to possess any residual muscle control. However, some functionalities of this BST module may be based on minimum muscle control. For example, an efficient method for turning on the interface has been proposed in [57] that uses eye closing. When individuals close their eyes for prolonged time (more than 1 second), the method detects a substantial variation in the energy of the alpha component of EEG. The cited method has been implemented in [22] by UFES lab for turning on and off visual stimulation in a system-paced SSVEP BCI. This implementation has not been included in SMAD, but it could be added as a further work.

5.2.1 User stimulation

The SSVEP brain response strongly depends on the characteristics of visual stimuli. In fact, works on BCI that employ such a paradigm need to describe clearly the type of stimulation used and how it is generated. The type of stimulation has to be chosen considering the strength of SSVEP response, the user safeness with respect to photosensitive disorders, and the minimization of

eye fatigue. The stimuli generator should guarantee certain features, such as precision and stability of stimulation frequencies. In this work, the focus has been also posed on the flexibility of stimuli generator, that is, the effort needed to setup the generator and migrate it from one installation to another. *Light* stimuli, implemented with light sources like LEDs, require additional hardware setup; the same happens with stimuli driven by external hardware, such as the FPGA subsystem used in [21]. Therefore, although LED stimulation elicits a stronger SSVEP, LCD stimulation is the recommended choice for an SSVEP-based BCI with few flickers ([53]). In SMAD, the user is presented with three *plain* stimuli displayed on the LCD monitor the wheelchair is equipped with.

In order to ensure reliable stimulation on LCD computer screens, the stimuli generator shall be synchronized with the refresh rate of the monitor. This limits the number of flickers and the choice of frequencies. For example, on a common monitor with 60 Hz refresh rate, the usable stimulus frequencies within 6 Hz and 8 Hz are only 6 Hz (10 frames per period), 6.66 Hz (9 frames per period) and 7.5 Hz (8 frames per period). To overcome this limitation, the approach presented in [52] has been followed. The authors showed how the visual graphical pattern can be updated after a variable number of frames, realizing a stimulus at any desired frequency. In fact, the stimulus signal at frequency f can be computed as

$$stim(f,i) = square\left[2\pi f\left(\frac{i}{RefreshRate}\right)\right] = \left|sin\left[2\pi f\left(\frac{i}{RefreshRate}\right)\right]\right| \tag{5.1}$$

where i is the frame index. $stim(f,i)$ signal can be used to drive the appearance and disappearance of a plain stimulus, as well as the alternation of patterns in pattern reversal stimuli. The stimuli generation program has been developed under Microsoft Windows platform using C++ and Microsoft DirectX 9.0 library.

Choosing stimuli frequencies is another key problem while using the SSVEP paradigm. This work adopted three of the four frequencies used in [21] – 6.4, 6.9 and 8.0 rps – because of two reasons. Firstly, to be able to compare additional results with the one obtained using the extensive EEG database recorded

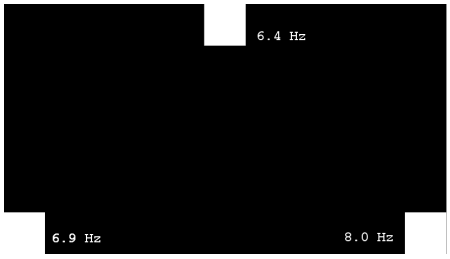


FIGURE 5.4: The three stimulation flickers used to elicit SSVEP response. Each white square appears at the rate indicated by the label next to it. These labels do not appear in the original screenshot.

with such frequencies. Moreover, these frequencies have demonstrated to elicit good SSVEP response as well as not to trigger photosensitive epileptic seizures. In [21], four *pattern reversal* stimuli – checkerboards – have been employed, at 5.6, 6.4, 6.9 and 8.0 rps. It is noteworthy that pattern reversal stimuli elicit an SSVEP response at the frequency of one alternation (see Section 2.4.2). On the contrary, the plain stimuli implemented in SMAD are modulated at the very same frequencies that appear in the EEG signal, namely 6.4, 6.9 and 8.0 Hz. In practice, if one desires an SSVEP response at 8 Hz, the modulating sin in (5.1) needs to use $f = 4Hz$ in the case of pattern reversal stimuli, whereas $f = 8Hz$ in the case of plain stimuli.

In Figure 5.4 a screenshot shows the three plain stimuli. The stimuli color has been fixed to white on black background. This choice has been driven by considerations about photosensitive disorders and colored stimulation([51]). Furthermore, the tests have empirically supported the idea that a solid square graphic is less tiring to look at than a more complex graphic such as a checkerboard. The frequency spectrum of the top-center stimulus is plotted in Figure 5.5, confirming that it represents an almost ideal square wave with frequency $f = 6.4$ Hz. The spectrum has been obtained by sampling the modulating function $stim(6.4, i)$. An even more reliable mean to verify the stimuli precision is measuring the rate of the actual light emitted by the LCD screen, for instance through a photo sensitive diode.

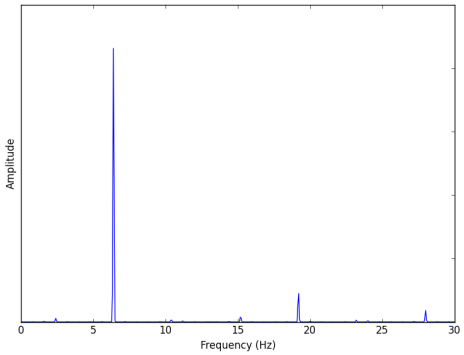


Figure 5.5: The frequency spectrum of the square wave corresponding to the 6.4 Hz stimulus. The reader can observe the dominant presence of the fundamental and third harmonic components.

5.2.2 EEG processing

The goal of EEG processing is to translate the subject's EEG signal into a discrete control signal that assumes one out of the three possible command values. The BST module handling EEG has been developed by the author, without relying on Emotiv's software. The procedures responsible for EEG signal processing have been designed and validated using a large amount of data coming from a prerecorded EEG database. Therefore, the EEG processing is presented in details and all the sub-modules are going to be analyzed.

EEG data provided by the acquisition stage forms a regular stream of samples, according to the sampling rate of the recording equipment. For this stream to be converted into discrete values, data windows with a fixed size are filled with the upcoming samples. EEG windows are analyzed one at a time; in other words, the processing pipeline is fed with data windows sequentially. Prior offline analysis is necessary in order to find out the optimal length of the windows. Larger windows would increase the accuracy of the BST, but to the detriment of speed. As saw when defining metrics like ITR and Utility, both accuracy and trial duration (or *time per command*) contribute to the overall performance of the system. After EEG acquisition, this BST module implements a slight variation of a signal processing pipeline quite common in

BCI field, composed by the following stages:

Pre-processing The raw EEG signal acquired from the recording equipment is initially manipulated with the aim of reducing noise.

Spatial filtering The EEG components corresponding to different electrodes are somehow combined to magnify the signal of interest and reduce background activity.

Feature extraction A number of features are computed and extracted from the filtered channels. These features should be useful to discriminate among the different patterns corresponding to different user commands.

Classification The extracted features are finally analyzed in order to identify the pattern performed, thus the output class.

In SMAD, this pipeline has been improved by segmenting the EEG signal window under processing and generating multiple classifications. An actual output will be triggered if and only if the global classification confidence exceeds a certain threshold. It is important to underline this aspect: not every user trial eventually produces an output. We want to avoid the user to experience unexpected action triggering. Since the signal may occasionally be very noisy and difficult to process, the choice in this case would be to produce no output at all and wait until the pattern identification becomes more reliable.

When no output can be produced, waiting until the next window is available would be very inefficient, especially for large EEG windows. For example, for 5-second windows, an unclassified trial would lead the user to keep gazing on the desired stimulus for at least 5 seconds more. Such a scenario may become very tiring and stressful from the user viewpoint. Therefore, a slightly more complex windowing structure is employed. Figure 5.6 provides a graphical representation. The BST processes a whole window of fixed length. If one pattern is recognized with enough confidence, the corresponding logical value is emitted as output and another window of the same length is collected. If the confidence value of the classification is below the given threshold, the

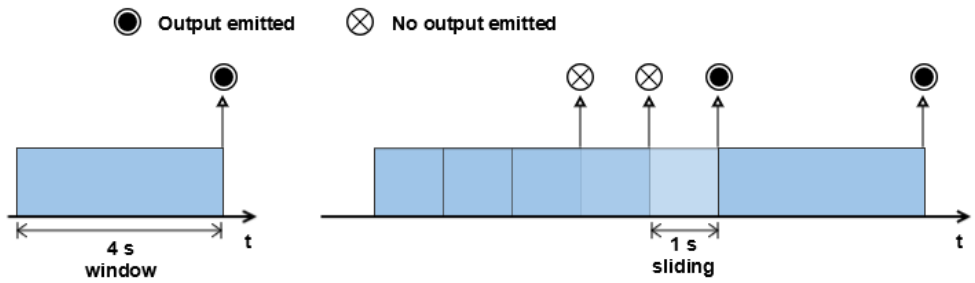


Figure 5.6: Pseudo-online protocol used to split EEG signal. Each window, indicated in blue, is passed to the signal processing pipeline and, if possible, produces an output. In case no output can be provided, the window is slid using one additional second of EEG data.

BST emits no output. In this case, one additional second of data is collected. Simultaneously, the first second of the original window is discarded. In other word, the window corresponding to the unclassified trial undergoes a 1-second sliding. After the sliding, the new window is in turn processed and a new classification is produced with the associated confidence value. The sliding continues similarly until a reliable classification can be achieved. Hence, in case of signal windows with low classification confidence, the BST attempts to provide an output once per second. The advantage over a fixed window structure is clear: when the output is missed for a 5-second window trial, the user does not need to keep gazing for an entire additional window because every second the BST has the possibility to recognize the pattern.

Clearly, the presented approach improves classification accuracy but has a negative drawback on the time needed by the BST to produce a command. Nonetheless, the experiments presented in Section 6.2 show that the overall impact on performance is often positive.

Figure 5.7 depicts the described EEG processing model. Hereafter, further details are provided about each sub-module.

Acquisition

Cerebral activity of the subject is recorded by means of the Emotiv EPOC headset introduced in 5.1.1. The Emotiv SDK does not provide the required

flexibility for collecting raw EEG data, especially online, i.e., with signals acquired and classified in real-time. Hence, an external open source driver called Emokit has been used[2]. The driver is responsible of decoding wireless transmission between the headset and the USB dongle and providing an intuitive API to access EEG data, both using C and Python code. Through the API it is possible to ensure the sensors contact quality and access the electric charge value of the fourteen electrodes, plus other data - e.g., an accelerometer.

The contact quality is represented by the electrical impedance of the electrodes: the less the impedance, the greater the contact quality and consequently the cleaner the signal. It is not always possible to achieve a good contact quality with all the electrodes; for example, often in our experiments the two temporal sensors offered quite poor contact quality. Therefore, a reduced number of sensors has been actually employed for collecting biosignals. In particular, four from the rear side of the scalp - O1, O2, P7, P8 - and two nearer to the front side of the skull - F3, F4. The sensors from occipital and parietal lobes are particularly relevant for SSVEP detection, since the primary visual area, where the SSVEP is elicited, is located in the occipital part of the skull. The choice of F3 and F4 has been driven by the need of recording EEG also from the front side of the skull to identify and cancel common noise patterns. Finally, the reference pair on the mastoid bone has been used. That pair demonstrated a perfect contact on every subject and offer almost total absence of electrical activity.

It is important to underline the difference within EEG acquisition between the dataset used for processing evaluation and the experiments with Emotiv EPOC. In the case of the dataset, an external medical-grade amplifier has been employed, connected to individual electrodes with no predetermined location. The sampling rate was 600 Hz and the sensors location were P7, PO7, P5, PO3, POz, PO4, P6, PO8, P8, O1, O2 and Oz, all in the rear side of the skull.

[2] `https://github.com/openyou/emokit`

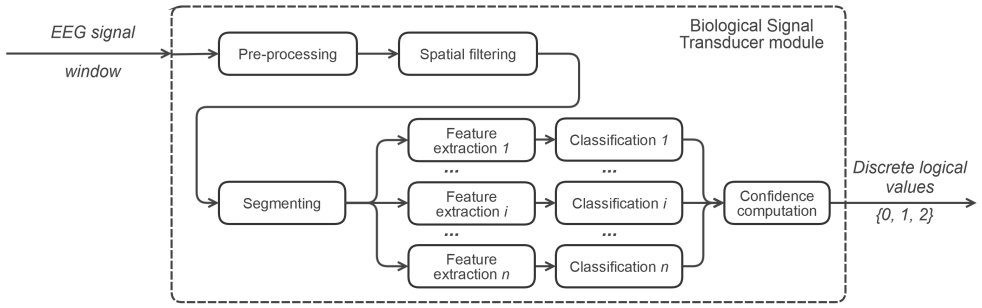


Figure 5.7: BCI processing pipeline.

Pre-processing

Traditionally, signal pre-processing applies simple methods in order to eliminate, or at least attenuate, the noise present in the signal and hence improve the signal-to-noise relation. One of the most common and straightforward approaches is frequency filtering. A frequency filter aims to exclude the signal components with undesired or unnecessary frequencies. A typical example is removing the power line component at 50/60 Hz.

For SSVEP detection the frequency band of interest has been fixed to 5-50 Hz. In the case of Emotiv EPOC, a band-pass filter is already implemented in hardware, providing a signal in 0.2 - 45 Hz bandwidth. Thus, it is sufficient to realize a digital high-pass filter to discard the low frequency components. In the case of the EEG dataset, the amplifier has no hardware filter, therefore a band-pass filter is required and the signal mean μ needs to be subtracted to cancel the DC component. In both instances, an equiripple FIR digital filter has been used with the following specifications:

Passband ripple 0.0575
Stopband amplitude 0.01
Density factor 20

Spatial filtering

As previously stated, EEG signal presents strong spatial characteristics. This means that most cerebral phenomena are associated to a particular area of

the scalp, hence recorded by different EEG electrodes. Each electrode collects electrical activity from sources next to it; nevertheless, the contribution of more distant sources is not negligible as it may represent up to the half of the electrode recording. This is an essential problem since the SSVEP is weak and affected by noise. A mean to reduce this contamination and obtain just the signal of interest is spatial filtering. Spatial filtering consists of applying a transformation to the original EEG channels, i.e., the electrodes, to get more meaningful channels for the detection of the patterns sought.

Some spatial filters are trained with preliminary data recorded from the subject. For example, Common Spatial Pattern filter (CSP) has been designed for separating a multivariate time-series in a linear combination of the original variables such that the variance for one class is maximized and simultaneously the variance for the other class is minimized. In the SSVEP case, CSP can be employed for maximizing the SNR of the evoked potential against the non-stimulus condition. CSP has been applied to SSVEP-based BCIs in various works, such as [54] and [55].

Spatial filters like the ones presented in [65] do not require calibration data. They try instead to combine the electrode channels in the most efficient way for the majority of subjects. Widespread approaches such as Common Average Reference filters (CAR) and Laplacian filters preserve the original number and spatial location of the channels. The resulting channels after filtering are still, for example, O1 and O2, but the SSVEP has supposedly been accentuated. [66] offers a discussion about spatial filters applied to SSVEP-based BCI. The bipolar setting is a well-known approach to remove common noise. The resulting channels represent the difference between two closely placed electrodes. Alternative spatial filters exist that provide output channels without any reference with the original locations, such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA). Nonetheless, they have not demonstrated themselves to be the best choice for SSVEP magnification.

In SMAD, the preferred spatial filter is CAR. This method, simple and

effective, produces EEG channels on which the widely distributed activity is reduced, including EMG artifacts and visual alpha rythms ([65]). Furthermore, it allows to clean all the original channels at the same time, making possible to exploit SSVEP detection techniques based on multivariate analyses, that is, considering multiple channels at once. CAR references all electrode to their common average. Therefore, the value of each channel is obtained by subtracting the mean of all EEG channels:

$$V_i^{CAR} = V_i - \frac{1}{n}\sum_{j=1}^{n} V_j \tag{5.2}$$

where V_j is the potential between the electrode j and the reference and n is the number of electrodes.

Segmenting

As Figure 5.7 depicts, the EEG processing pipeline performs multiple feature extraction and classification stages. The segmenting stage splits the EEG window that enters the BST. The segmenting is executed on temporal dimension: the spatial components, e.g., the channels, are left untouched. This stage produces several sub-windows, or *segments*, that will be individually passed to the feature extraction and classification stages.

Let's suppose having a t second EEG window. To allow accurate classification, it is important to avoid producing too small segments. With small segments, the information available for pattern identification might be insufficient. Therefore, the segments are kept of length $t - 1$. The residual second of data is used to create several segments by sliding. The BST divides the window in five segments, hence the sliding consists of 0.2 seconds.

Although the resulting segments will sensibly overlap, thus having most information in common, in this way one obtains five partially independent inputs for the next stages. If the identified pattern sensibly differ among the segments of the same window, it will be a good indicator that the recognition is not reliable enough for that window.

Feature extraction

This stage aims to find out the best or more relevant representation for SSVEP. As often in signal processing, feature computation and extraction is the most delicate part, since a meaningful representation crucially affects the pattern recognition performance. What is desired on feature extraction stage is to highlight the evoked potentials while reducing the remaining, background activity. For example, for 6.4 Hz stimulation the resulting features should point out the 6.4 Hz component instead of other periodical activity like α rhythm. As showed in Figure 5.7, in SMAD various feature extractors operate simultaneously on the same EEG signal window. Albeit the execution is not actually parallelized, the reader can view it as such from a functional point of view.

Three feature extraction methods have been tested in this thesis. Due to the importance of the role this stage plays, all methods are presented in details in the following section. Each technique adopts a different viewpoint to compute the best features. However, all methods returns a single value that represents the predominant frequency among the stimulation frequencies.

Classification

Eventually, the extracted features need to be associated with a certain class, that is, a category out of a set of possible ones. Conforming to the number of stimulation flickers, three classes are supported. The classes are represented by simple logical values, e.g., integer numbers 0, 1 and 2. Analogously to the BST using EMG/EOG, there must be a consistent and coherent association between such logical values and the CI user commands.

Since one basic assumption of SMAD is to use untrained classifiers (see Section 2.3), the BST employs a simple rule-based classification approach. Recalling that the feature extraction stage works on five segments, it will produce five different values indicating the dominant SSVEP frequency for each segment. The classification procedure is more easily explainable with an example. Let's suppose that $f_0 = 6.4$ Hz, $f_1 = 6.9$ Hz and $f_2 = 8$ Hz; the

output from feature extraction stage could be

Segment 1	*Segment 2*	*Segment 3*	*Segment 4*	*Segment 5*
f_0	f_0	f_1	f_0	f_0

The classification confidence is computed as following:

$$C_j = \frac{\sum_{i=1}^{5} g(s_i, f_j)}{5}, j = 0, 1, 2 \tag{5.3}$$

where $g(s, f)$ is an indicator function defined as

$$g(s, f) = \begin{cases} 1 & \text{if the segment } s \text{ produced the frequency } f \\ 0 & \text{otherwise} \end{cases}$$

Continuing with the example table depicted above, the confidence values would be

f_0	f_1	f_2
0.8	0.2	0

The classification stage will provide an actual output, indicating one of the three possible values, only if the corresponding frequency has its confidence value equal to 1. In other words, a window will be assigned to a class only if the totality of five segments, obtained from the window, have exhibited the same dominant frequency. In the provided example no output would be produced. This is a very conservative attitude, due to the propensity of SMAD toward accuracy instead of recognition speed. The user might experience long periods of time without output, hence prolonged trial duration, if the EEG signal is noisy or the elicited SSVEP too weak. Yet, it is immediate to see that decreasing the confidence threshold, e.g., to 0.8, is an easy way to shorten the trial duration while likely keeping a good accuracy.

5.2.3 Feature extraction for SSVEP

5.2.3.1 PSDA

PSDA method is the traditional and widely used frequency recognition technique in SSVEP-based BCI. It has been proposed in [38] to compute how the power of a signal or time series is distributed over frequencies. By computing the PSD of the EEG signal one obtains what is called frequency spectrum: its power distribution in the frequency domain. PSDA method aims to assess the signal to noise ratio (SNR) at the stimulus frequencies:

$$SNR_k = 10\log_{10}\left(\frac{nP(f_k)}{\sum_{m=1}^{n/2} P(f_k + mf_{res}) + P(f_k - mf_{res}}\right) \tag{5.4}$$

where f_k is the stimulus frequency, f_{res} is the frequency resolution of the PSD and SNR_k indicates the ratio of signal power at f_k - $P(f_k)$ - over the power at n adjacent frequencies.

SNR_k is computed for each stimulus frequency f_k and occasionally for their harmonics. $P(f_k)$ is obtained with methods such as *periodogram* or *Welch method*, applied to the temporal EEG data. The input to the transform may be a monopolar or bipolar channel, but it has to be an one-dimensional signal. This is a major drawback of PSDA. This method requires to know what is the single most informative channel of EEG signal. In fact, PSDA can indeed be used to find the optimal monopolar or bipolar lead, but doing that requires a period of calibration. Moreover, when SNR in the original signal gets lower, PSDA recognition accuracy falls rapidly.

For the sake of comparison among feature extractors, PSDA has been implemented in SMAD. To guarantee an high PSD resolution, 4096-points Welch method with Hann window has been used. Thus, $f_{res} = F_s/4096$ being the sampling rate F_s. The range containing $n = 20$ adjacent points is extracted as $[f_k - 10 * f_{res} : f_k + 10 * f_{res}]$. The range median is considered as $P(f_k)$. After the computation of SNR_k with the formula presented above, the dominant frequency is merely the one with largest SNR, or

$$O = \max_k SNR_k, k = 1, 2, 3 \tag{5.5}$$

5.2.3.2 CCA

CCA method has been proposed in [44] by the same laboratory than formerly developed PSDA; the authors stated in the paper that this technique outperforms traditional PSDA. CCA is a statistical method useful for multivariate analysis. While ordinary correlation analysis studies the relationship between two variables, CCA extends this approach to sets of variables. The aim of CCA is to find a pair of linear combinations of two sets such that the correlation between the sets is maximized. Such linear combinations are called *canonical variables*. The correlation coefficient quantifies the correlation strength between the sets. This process can be repeated as many times as the number of variables in the smaller set. That generates multiple correlation coefficients, although usually only the largest one is considered.

When applying CCA method to SSVEP, variables in one set are the multiple electrodes, or channels, that compose the temporal EEG signal, $X(i)$. The discrete time index i is used to stress the fact that the continuous signal has been sampled at F_s sample rate. The second set, $Y(i)$, represents the stimulus signal filtered by the brain. Being a periodic signal, the stimulus signal can be decomposed into the Fourier series of its harmonics

$$Y_k(i) = \begin{pmatrix} y_1(i) \\ y_2(i) \\ \vdots \\ y_{2h-1}(i) \\ y_{2h}(i) \end{pmatrix} = \begin{pmatrix} \sin(2\pi f_k i) \\ \cos(2\pi f_k i) \\ \vdots \\ \sin(2h\pi f_k i) \\ \cos(2h\pi f_k i) \end{pmatrix}, i = \frac{1}{F_s}, \frac{2}{F_s}, \ldots, \frac{T}{F_s} \tag{5.6}$$

where h is the number of considered harmonics, T is the number of samples and f_k is the fundamental frequency corresponding to the considered stimulus.

The goal of CCA is to find the canonical variables $x = X^T W_x$ and $y = Y^T W_y$ with maximum correlation. In order to solve this optimization problem, CCA method needs to compute the weight vectors, W_x and W_y, which maxi-

mize the correlation between x and y, according to the following constraints:

$$E[xx^T] = E[x^T x] = E[W_x^T X X^T W_x] = 1 \tag{5.7}$$

$$E[yy^T] = E[y^T y] = E[W_y^T Y Y^T W_y] = 1 \tag{5.8}$$

The correlation index ρ is calculated with respect to W_x and W_y as follows

$$\rho_k = \rho_{W_x, W_y}(x, y) = \frac{E[x^T y]}{\sqrt{E[x^T x]E[y^T y]}} = \frac{E[W_x^T X Y^T W_y]}{\sqrt{E[W_x^T X X^T W_x]E[W_y^T Y Y^T W_y]}} \tag{5.9}$$

for $Y = Y_k(i)$, that is, the reference signal at frequency f_k.

Note that CCA considers at the same time all the channels that compose $x(i)$, that is, the EEG signal. Using the electrodes known as containing SSVEP components, CCA is able to exploit all that data to compare the actual EEG signal with the reference periodic signal. Analogously to PSDA, the dominant frequency is obtained through CCA by looking at the largest ρ_k, or alternatively

$$O = \max_k \rho_k, k = 1, 2, 3 \tag{5.10}$$

Implementation-wise, CCA can be computed by means of Singular Value Decomposition (SVD) on the correlation matrix $C = corr(X, Y)$ [67].

5.2.3.3 MSI

MSI method, as well as CCA, is a recent multichannel detection technique. MSI has been proposed in [68] as a new detection method that may outperform PSDA and CCA. It is grounded on dynamical system theory, in particular on the so-called S-estimator. The S-estimator was originally proposed to measure the synchronization between cortex regions; in SSVEP case, it would act as the Multivariate Synchronization Index that indicates the synchronization between EEG signal and the stimuli periodic references.

Considering a multivariate signal, the S-estimator is tied to the entropy of the eigenvalues of the correlation matrix. Conceptually, the index is inversely

proportional to the embedding dimension: the more synchronized the data, the minimal its dimensionality ([68]). The dimensionality of data is minimal for a low entropy of the normalized eigenvalues. Therefore, the method expresses an high synchronization index when the entropy of the eigenvalues is small, and vice versa.

Analogously to CCA, let's define one matrix $X(i)$ for the EEG signal of size $N \times T$, where N is the number of channels and T is the number of samples, and the reference signal matrix $Y(i)$ of size $2h \times T$. h is the number of considered harmonics. Hence, the correlation matrix is computed as follows

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} \tag{5.11}$$

$$C_{xx} = \frac{1}{T} X X^T \tag{5.12}$$

$$C_{yy} = \frac{1}{T} Y Y^T \tag{5.13}$$

$$C_{xy} = C_{yx} = \frac{1}{T} X Y^T \tag{5.14}$$

where the reader should pay attention to avoid confusing the number of samples T with the matrix transpose sign.

In order to reduce the influence of autocorrelation matrices C_{xx} and C_{xy}, the correlation matrix is transformed as

$$R = \begin{bmatrix} I_{N\times N} & C_{xx}^{(-\frac{1}{2})} C_{xy} C_{yy}^{(-\frac{1}{2})} \\ C_{yy}^{(-\frac{1}{2})} C_{yx} C_{xx}^{(-\frac{1}{2})} & I_{2h\times 2h} \end{bmatrix} \tag{5.15}$$

where $I_{N\times N}$ and $I_{2h\times 2h}$ are identity matrices.

The autocorrelation is canceled out in (5.15). Then, the normalized eigenvalues $\lambda_1', \lambda_2', \ldots, \lambda_{N+2h}'$ of R can be computed as

$$\lambda_i' = \frac{\lambda_i}{\sum_{i=1}^{(N+2h)} \lambda_i} = \frac{\lambda_i}{tr(R)} \tag{5.16}$$

Finally, recalling the Von Neumann entropy of a matrix $H(A) = -\sum_i \lambda_i \log \lambda_i$, the synchronization index is defined as

$$S = 1 + \frac{\sum_{i=1}^{(N+2h)} \lambda_i' \log \lambda_i'}{\log(N+2h)} \tag{5.17}$$

The S index is zero when $C_{xy} = C_{yx} = 0$, that is, totally uncorrelated sets. On the other hand, when X and Y are perfectly correlated, R will have ones on the main diagonal and zeros elsewhere. Thus, only one eigenvalue is one and S is consequently one. In other cases, the value of S will vary from zero to one.

Similarly to PSDA and CCA, to obtain the dominant frequency is sufficient to calculate the synchronization index between the EEG signal and each reference signal $Y_k(i)$ and extract the maximum index. Mathematically, the output will be

$$O = \max_k S_k, k = 1, 2, 3 \tag{5.18}$$

where S_k is the S index computed between $X(i)$ and $Y_k(i)$.

Experiments, tests and results 6

This chapter illustrates the experiments performed using the two presented BSTs and the results.
Firstly, we focus on the EMG/EOG-based BST. The experimental settings and protocols are described, as well as the results.
In the second section, we concentrate on the EEG-based BST, or BCI. Tests and results concerning both offline analysis and online experiments are presented.
Finally, we discuss the results obtained and compare these results to similar works in literature for identifying possible room for improvement.

6.1 EMG/EOG tests and results

This section presents experiments performed and results achieved using EMG/EOG-based BST, introduced in Section 5.1.

This BST employs face gestures and eye movements. Thus, during tests the subjects were asked to perform such gestures to verify whether the system was able to recognize them correctly. Emotiv EPOC device acquires user biosignals and delivers the digital samples to the processing pipeline. Since the BST does not need a training period, the only preparation for test sessions consisted in putting the EPOC headset on the head of the subject. It is worth to recall that three gestures are considered: raising both eyebrows to navigate to the next option, moving eyes right or left to navigate to the previous option, and prolonged clenching to select the current option, thus operating the corresponding device.

The signal processing techniques have been already been validated because they are embedded in the Emotiv's SDK. For this reason, we skipped offline analysis, which is required for evaluate signal processing algorithms and choose the best parameters. All the tests performed with EMG and EOG paradigms are already online. This means that the gesture performed by the subject is classified in real-time and a feedback is immediately provided to the user.

6.1.1 Experimental methods

All experiments have been performed at the *LAI*, the Intelligent Automation Laboratory at Universidade Federal do Espirito Santo (UFES), from November 2014 to March 2015. This section presents the subjects involved, chosen experimental protocols and settings and performance metrics used to evaluate results.

Subjects

Eight healthy subjects participated to the tests on the EMG/EOG-based BST. We chose not to involve persons with disabilities in this preliminary testing

phase for two main reasons. Firstly, the system was still under development. Hence, a subject may had gone a long way to reach the lab and any malfunctioning would squander this effort. Secondly, we feel appropriate the presence of researchers with medical formation when performing experiments with subject with disabilities. Although results obtained with healthy persons are not guaranteed to hold when the system is used by individuals with motor impairments, extensive experimentation with subjects with disabilities has been scheduled as a future work.

The participants are students from LAI, seven males and one woman, all aged from 23 to 35. The presence of a long-haired woman is particularly important to test how hair cumbers the sensors contact quality. Almost every subject had some previous experience with HMI though biosignals, but none of them using face and eye movements. We refer to the subjects with ordinal numbers, i.e., $\#1, \#2, \#3, \#4, \#5, \#6, \#7, \#8$.

Experimental sessions

As previously introduced, for this BST the tests were carried out in *online* fashion. During an online session, in each trial the subject is supposed to perform a gesture; the system classifies the gesture and returns a visual feedback. In contrast, an *offline* session consists of recording several trials of user gestures without providing any feedback. The data is subsequently analyzed in order to classify the trials and adapt the parameters of the processing techniques. Since Emotiv's SDK internally processes EMG and EOG signals and identifies the gesture pattern without offering much customization, the offline phase can be skipped.

All trials for each subject have been recorded in the same session. This is because we empirically found that the classification of face and eye movements shows less inter-session variation than inter-subject. In other words, experimenting with several subjects is worth more than collecting various sessions from the same subject. The subjects were seated on a comfortable chair, in front of a laptop where the program was running.

Each session is composed of three runs of 30 repetitions associated to different tasks. Before each session, we checked the electrodes contact quality to ensure it was optimal. This can be done using the *Emotiv Control Panel* or a simple custom program exploiting *Emokit* library. The runs are interleaved with a one minute break. Therefore, a total of 90 trials per user has been considered. During the trial the participant is asked to perform a specific task, i.e., clenching, raising brows or moving eyes left/right. The monitor indicates what is the currently required task. Each run consists of 30 repetitions of the same task. This may have leaded to a sort of user adaptation; an improved protocol would randomize the order of tasks within a single run, keeping the total number unchanged.

Being the session online, there is no fixed time for each trial. Instead, the system is expecting a gesture for an undefined time, until a pattern is eventually classified (correctly or not). Hence, it is possible as well to record the time needed for the user to issue a command, that is, the time passed from the beginning of the user gesture to the classification of the command. After the system has classified the currently performed gesture, a 5 seconds break allows the user to return to resting state and briefly relax.

With the aim of realizing a prototype, we used *Emokey* as software wrapper to convert Emotiv's SDK output to discrete BST control values (see Section 5.1.2). For continuous clenching and raising brow values, Emokey requires to set fixed threshold above which to simulate key pressing. There are two threshold-based rules, one for each gesture. We opted to trigger an action when the correspondent value (e.g., clenching) is greater than 0.8 and the other one (raising brow) is less than 0.2. These thresholds are obtained empirically; we found them to provide good results with most subjects. For detecting eye movement, the SDK already provides a discrete number assuming values 0 and 1. Intuitively, the value 1 indicates occurrence of eye movement. Emotiv's SDK differentiates between moving eyes right or left; we map both movements to the same logical value, i.e., the same simulated key.

6.1.2 Results

Output messages displayed by the system during the experimental sessions have been used to estimate the performance of the BST and the overall assistive system. For each trial, the system returns the classified gesture. Hence, each trial generates an output that can be correct or wrong. By merely counting the number of correctly classified trials we can extract the BST accuracy and the Kappa coefficient. Moreover, at the end of each session the system provides the average of the trials duration. This time represents the average time, among all trials, needed by the user to trigger a command. The trial duration is required for computing metrics such as ITR and Utility. For the Utility computation, we considered a menu panel with $N = 6$ options, as indeed this is the maximum number of options allowed in the CI. We express ITR and Utility in bits over minutes, as it is common in literature; this is obtained simply multiplying the b/s value from the formulas times 60.

We initially treat each of the three user gestures independently. In fact, the three movements do not present an homogeneous classification accuracy; clenching, for example, is correctly classified more likely than raising brow. Table 6.1 summarizes the metrics presented in Section 3.3 for clenching, while Table 6.2 and 6.3 contain the same metrics for raising brow and moving eyes, respectively. All three gestures demonstrate satisfying results, especially in terms of speed of recognition. It can be seen that a command is triggered after a very small amount of time. In one case, it is even less than a second. Yet, since results may remarkably vary among users, we can observe than subject #1 was not able to perform raising brow. The outcome was always wrong. As a consequence, we ignored that run and the user is likely forced to use only the two remaining gestures.

It can be noted that clenching offers best performance, both in terms of accuracy and trial duration. As a consequence, clenching shows considerably higher values of ITR and Mean Utility. This is the main reason why in SMAD clenching represents the *Do* user command, i.e., the command that operates

Subject	**Clenching**					
	Correct trials	Accuracy [%]	Kappa	Duration [s]	ITR [b/m]	Mean Utility [b/m]
#1	30	100	1.0	1.06	89.71	71.13
#2	30	100	1.0	0.86	110.58	87.67
#3	29	96.67	0.95	1.29	62.37	54.55
#4	30	100	1.0	2.08	45.72	36.25
#5	28	93.33	0.9	3.09	22.62	21.14
#6	25	83.33	0.75	2.07	22.27	24.28
#7	26	86.67	0.8	3.86	13.76	14.33
#8	30	100	1.0	2.20	43.23	34.27
Average	28.5	95	0.93	2.06	51.28	42.95

Table 6.1: Performance metrics for the *clenching* gesture.

Subject	**Raising brow**					
	Correct trials	Accuracy [%]	Kappa	Duration [s]	ITR [b/m]	Mean Utility [b/m]
#1	x					
#2	27	90	0.85	1.35	45.15	44.68
#3	30	100	1.0	1.12	84.91	67.32
#4	28	93.33	0.9	1.87	37.37	34.94
#5	22	73.33	0.6	2.41	11.99	14.6
#6	24	80	0.7	1.55	25.67	29.19
#7	29	96.67	0.95	3.41	23.59	20.64
#8	19	63.33	0.45	2.81	5.77	7.15
Average	25.5	85.24	0.78	2.07	33.49	31.22

Table 6.2: Performance metrics for the *raising brow* gesture. Subject #1 was not able to achieve this gesture.

the currently highlighted option. In this way, when the user reaches the desired option and wants to operate it, there is a strong chance to succeed at the first trial. We believe that this element is essential for the aim of minimizing user frustration.

Analyzing the metrics for raising brow and moving eyes, the reader can immediately see a quite identical value of Mean Utility - and similar for ITR. Nevertheless, this is caused by different factors. Raising brow, as it appears clearly looking at accuracy and Kappa coefficient, does not guarantee high pre-

Subject	**Moving eyes**					
	Correct trials	Accuracy [%]	Kappa	Duration [s]	ITR [b/m]	Mean Utility [b/m]
#1	30	100	1.0	1.60	59.44	47.12
#2	30	100	1.0	1.58	60.19	47.72
#3	30	100	1.0	1.87	50.85	40.32
#4	22	73.33	0.6	6.32	4.57	5.57
#5	30	100	1.0	2.39	39.79	31.55
#6	29	96.67	0.95	1.43	56.26	49.21
#7	22	73.33	0.6	4.07	7.1	8.64
#8	29	96.67	0.95	3.67	21.92	19.18
Average	27.75	92.50	0.89	2.87	37.52	31.16

Table 6.3: Performance metrics for the *moving eyes* gesture.

cision in recognition. Indeed, the average accuracy is 10% less than clenching. This worsening can be ascribed to the similarity of muscular patterns between clenching and raising brow. On the other side, moving eyes offers a quite good average accuracy, whereas the trial duration increases substantially. This supports the idea that EOG is generally a reliable paradigm, but it is troublesome with some users and takes a long time for the eye movements to be recognized.

Figure 6.1 gives a general idea about the results obtained by the eight subjects. Each point corresponds to one run, during which a subject was repeating 30 times a single gesture. We can notice that the distribution of the points tend toward the upper left corner of the diagram, hence perfect accuracy and low recognition time (trial duration). In fact, the large majority of the trials showed at least 80% accuracy and less than 4 seconds to recognize the gesture. The size of the points is proportional to the Mean Utility. Therefore, we can observe how the Utility decreases both for the points with less accuracy and for the ones with higher recognition time.

In order to get the global performance for each subject, the metrics related to the three gestures need to be combined. A way could be the simple average; however, averaging may introduce a little bias, since the navigation commands (*Next* and *Previous*) are likely more frequent than the confirmation *Do* command. Therefore, we apply a weighted averaging approach to accuracy

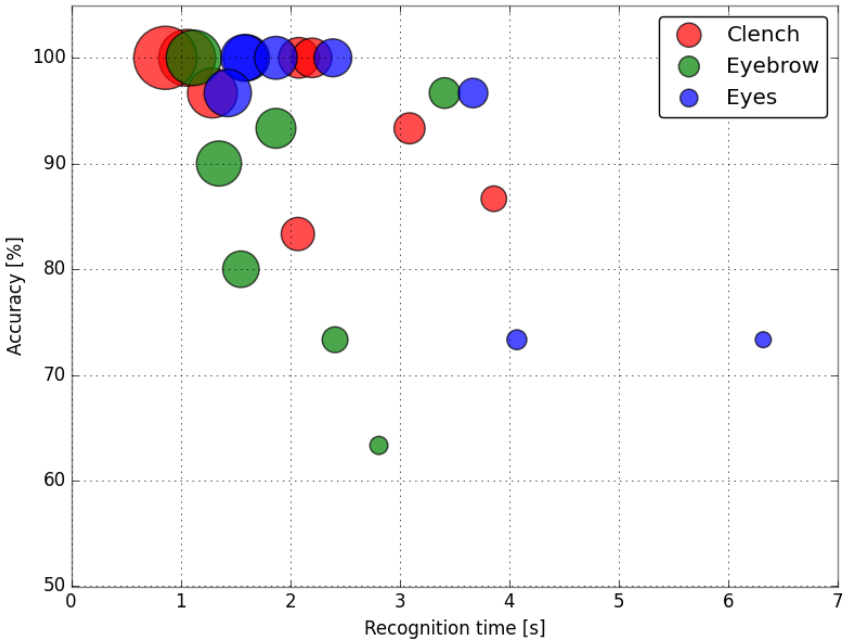


Figure 6.1: Classification accuracy of single experimental trials using EMG/EOG BST with respect to recognition time. Each point corresponds to a single subject performing a specific gesture. The radius of the points is correlated with the value of Mean Utility. The higher the Utility, the bigger the point.

and trial duration, where the metric values corresponding to raising brow and moving eyes weight twice than the clenching one. The remaining metrics are computed starting from the averaged values of accuracy and trial duration. Table 6.4 reports the summarized results.

The averaged accuracy and trial duration indicate a quite fast and reliable gesture recognition. Considering more carefully the values of ITR and Utility, we find again the situation graphically depicted in Figure 4.4(b). Note that the diagram expresses accuracy in the range $[0, 1]$, while the table in terms of percentage. For very large values of accuracy – near 1.0 or 100% – ITR provides an overestimation of the real performance. Indeed, for subjects #1, #2 and #3 ITR value is substantially higher than Utility. By contrast, ITR and Utility are closer for accuracy values near to 90%. Utility becomes eventually larger for lower accuracy, e.g., subject #8. As argued in Section 4.2, Utility offers a

Subject	**Overall BST**				
	Accuracy [%]	Kappa	Duration [s]	ITR [b/m]	Mean Utility [b/m]
#1	100	1.0	1.42	66.97	53.09
#2	96	0.94	1.34	58.33	51.76
#3	99.33	0.99	1.45	62.91	51.3
#4	86.67	0.8	3.69	14.39	14.98
#5	88	0.82	2.54	22.1	22.56
#6	87.33	0.81	1.61	33.91	34.96
#7	85.33	0.78	3.76	13.35	14.17
#8	84	0.76	3.03	15.66	16.92
Average	90.83	0.86	2.36	35.95	32.47

Table 6.4: Performance metrics for each subject performing the three EMG/EOG user tasks. The values are obtained through weighted average among the results corresponding to the three tasks.

better and more realistic assessment since it takes into account the number of menu options and the navigation behavior. Hence, the reader is suggested to focus on Utility values to get a reliable estimation of the quantity of information the user is able to express through the assistive system.

To give a more intuitive representation of the condensed results, Figure 6.2 reproduces the overall performance for the eight subjects. Again, the bigger the point, the higher the corresponding Utility. We immediately observe three users with optimum performance. Besides them, the accuracy is quite constant, between 84% and 90%, for the other subjects. We can take this range as appropriate with respect to user expectation. Yet, two or three users presents less satisfying results, mainly due to a longer recognition time. The system is still perfectly usable by these users, yet with poorer efficiency.

In concluding, totally realistic experiments have been carried out by connecting real lamp, fan, television and radio to the acrylic box (see Section 3.2). The users, through facial gestures and eye movement, have succeeded in performing the following sequence of actions:

1. Turn on lamp
2. Turn on fan

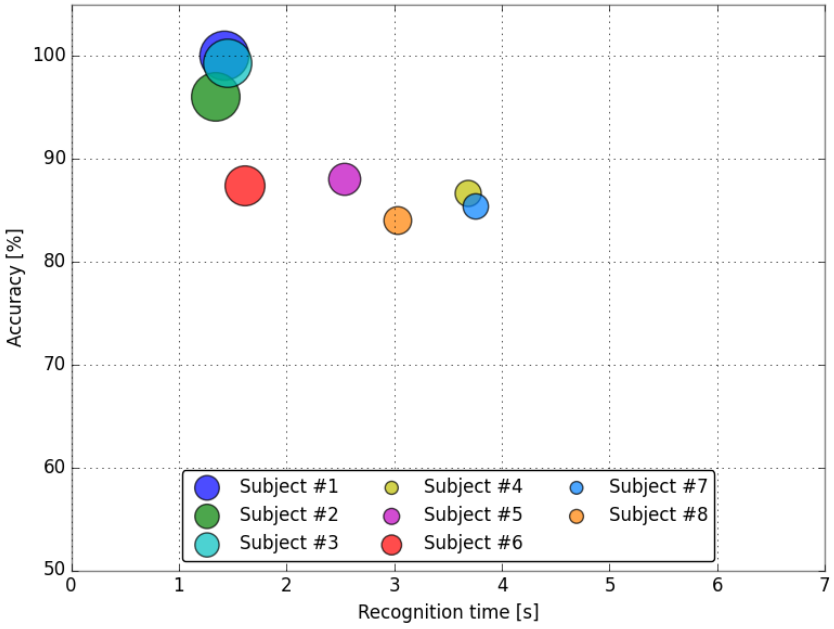


Figure 6.2: Classification accuracy of experimental sessions using EMG/EOG-based BST with respect to recognition time. Each point corresponds to one of the eight subjects. The radius of the points is correlated with the value of Mean Utility. The higher the Utility, the bigger the point.

3. Turn on TV
4. Change TV channel up twice
5. Turn TV volume up once
6. Go back to main menu and hide GUI

The entire sequence has been completed in less than three minutes and correct behavior of the whole system has been positively verified.

6.2 BCI tests and results

This section illustrates experimental tests and results using EEG-based BST, or BCI, presented in Section 5.2.

This BST processes cerebral activity under the SSVEP paradigm. SSVEP consists of periodic electrical components of EEG signal, resulting from user stimulation with periodically recurrent visual patterns. Hence, the experi-

ments involve the subjects looking at the visual stimuli while the system is capturing their EEG signal. Two types of analysis have been made: first offline, to try different processing techniques and parameters, then online for actually examining how the system works in practice. The offline analysis is in turn composed by two parts. We started from a prerecorded EEG database, provided to our lab by the research work presented in [21]. This database contains EEG recordings of 19 subjects during SSVEP stimulation. Thus, it is a very indicative base for evaluating the effectiveness of signal processing techniques.

After extensive tests on the cited dataset, we collected actual EEG recording sessions using the Emotiv EPOC and we applied the same processing methods. The frequencies used for SSVEP elicitation were the same used when recording the dataset; since for the dataset four stimuli were employed and in SMAD just three commands exist, we removed one stimulus relative to one of the frequencies. The stimulation method slightly changed, as explained in Section 5.2.1. Despite these variations, we were able to use the recorded data to optimize the processing techniques and their parameters. Finally, online sessions have been recorded. The subjects were asked to simulate a real utilization scenario, when they navigate the menu, turn on and off several devices and manipulate the television sub-menu.

6.2.1 Experimental methods

SSVEP database

As stated in the past section, the first tests have been carried out on a previously recorded EEG database. It contains EEG recording sessions of 17 subject, out of which 5 with some kind of disabilities. The disabilities affect not only the motor area, but also, for some subjects, the cognitive sphere. The database was recorded with four visual stimuli, hence four stimulation frequencies. Each subject was requested to look at one stimulus at a time for three runs. The first run consisted of gazing each stimulus for 20 seconds, for two

iterations. The second run was similar, but each gaze period lasted 10 seconds and the iterations are four. Finally, the third run was made of five iterations of 5 seconds gaze period. The four stimulation frequencies were fixed at 5.6, 6.4, 6.9 and 8.0 rps. Note that, being pattern reversal stimuli, their reference rate is measured in reversal per seconds instead than in Hertz. We refer to the subjects in the database with ordinal number from #1 to #17.

For this preliminary analysis to be as close as possible to the actual behavior of the BST, we apply the pseudo-online protocol described in Section 5.2.2. Thus, the EEG is analyzed in fixed windows that may, or may not, produce a value as output after passing through the signal processing pipeline. To clarify the terminology, hereafter we use the term *trial* to indicate the period during which the user is gazing a single stimulus for a varying amount of time (5, 10 or 20 seconds). A *window* is the fixed quantity of EEG signal that is passed to the signal processing algorithms, whereas a *segment* is each of the sub-fragment a windows is divided during processing, with the aim of providing several sub-classifications. The recognition time is the amount of time between the user starting to gaze a stimulus and the system producing an output – correctly or not. The minimum recognition time is equal to the length of the window, when all windows produce some output; yet, for some subjects the recognition time can grow if many windows do not trigger any output command. For an exhaustive explanation of the signal processing pipeline, see Section 5.2.2.

We used all the 10-second and 5-second trials, leading to a total of 9 trial per frequency and 36 trials per subject. Instead, we considered the 20-second trials unrealistic for an actual SSVEP-based BCI. Such a prolonged gazing period would be too tiring for the user. Furthermore, the SSVEP pattern would remain relatively stable for the entire period. This may likely lead to an overoptimistic performance evaluation. Intuitively, if the pattern is correctly identified analyzing the first windows, it would continue to produce correct outcome for most of the following ones. Since the BST accuracy is computed by considering the total number of correctly classified windows, the windows contained in the 20 second trials may bias the accuracy evaluation. We tested

the processing with windows from 2 to 5 seconds.

Each window is passed to the processing pipeline with the aim of SSVEP classification. The choice of pre-processing (frequency filtering) and spatial filtering algorithm (CAR) are illustrated in Section 5.2.2. Different feature extraction techniques were compared. First, the PSDA method has been employed to find out the more useful electrode for SSVEP detection, that is, the electrode that carries out most information. This is necessary to properly exploit the PSDA method, because it just processes the univariate data corresponding to one electrode. Then, we compared the results obtained with PSDA applied to the best electrode with the results given by multivariate analysis techniques, namely CCA and MSI. Only the occipital electrodes (O1, O2 and Oz) have been taken into account in these tests. The reason is to be found in the location of SSVEP, which many studies have shown is maximum in the occipital area of the scalp.

Offline experiments

Offline recording sessions have been recorded at the *LAI* during February and March 2015. Analogously to the first BST presented, only healthy subjects participated to the tests. The participants were four LAI students, all males, between 20 and 30 years old. Three of them had some previous experience with SSVEP-based BCI. We refer to the subjects as #18, #19, #20, #21. Figure 6.3 depicts subject #19 while performing a session.

During each session, we placed Emotiv EPOC headset on the subject. The sensors contact quality (i.e., the impedance) has been always checked through *Emokit* library. While the Emotiv SDK furnishes a graphical indication using colors to express the degree of electrodes contact, Emokit provides numerical values, thus a more detailed information. We empirically found that values from 4000 to 8000 correspond to optimal quality. Therefore, all experiments have been performed after checking the employed electrodes to have such a contact quality value. Differently from the EEG database, the visual stimuli are three, thus with three different stimulation frequencies. Each session

Figure 6.3: Subject performing SSVEP offline session.

consists of two runs separated by a two minute break for the user to rest the eyes. Each run is in turn composed by four iterations gazing at each stimulus with gaze period of 20 seconds. In total, we get 8 trials per frequency and 24 trials per subject. Although, as previously stated, 20-second trials may lead to overoptimistic performance evaluation, we opted for this trial size to obtain as much data as possible having a reduced number of subjects and due to the lack of time.

The EEG processing is kept unchanged with respect to the tests on the database. For the sake of comparison, we applied the same feature extraction techniques – PSDA, CCA, MSI. One important difference is that, with EPOC headset, Oz location is not available but just O1 and O2. Furthermore, since EPOC sampling rate is considerably lower than the acquisition equipment used for recording the database, we had to analyze larger data windows to obtain satisfying results. Hence, the window lengths considered range from 4 to 8 seconds.

Online experiments

Online sessions are the most realistic simulation performed with this BST. In these tests we implemented the pseudo-online protocol, used for offline analysis, in a real-time fashion. Two subjects (#19 and #21) have been asked to issue a series of commands with the aim of simulating a common user scenario. In particular, the scenario was defined as follows:

1. Turn on lamp
2. Turn on fan
3. Turn on TV
4. Change TV channel up twice
5. Turn TV volume up once
6. Go back to main menu and hide GUI

This sequence needs to be translated into the corresponding navigation and confirmation commands, meaningful for the CI. For further details about the definition of the commands see Section 4.2. We refer to the commands as **D** for *Do*, **N** for *Next* and **P** for *Previous*. Depending on the direction chosen while navigating the options, the command sequences that the two subjects were supposed to perform are:

Subject #19 : **N N D P P D N D N D D N N D N N D N N N D**

Subject #21 : **N N D P P D N D N D D N N D N N D P P D**

After the system has recognized a command, a 2-seconds break allows the user to gaze another stimulus and rest the eyes.

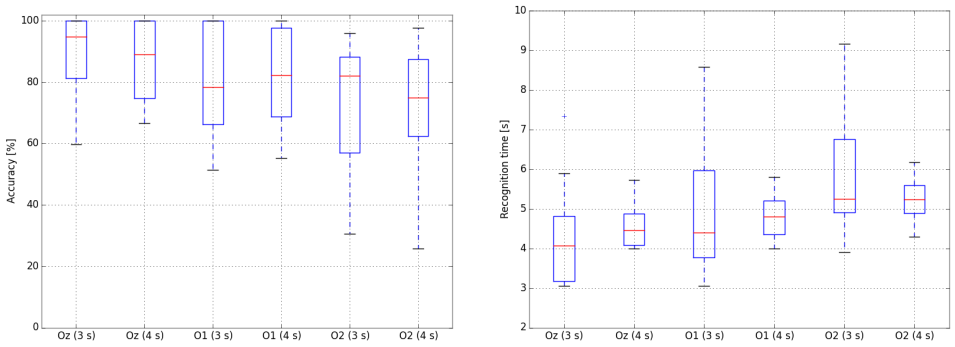
Note that, if some commands are interpreted erroneously by the BST, the user should perform the additional commands required to recover the error. For example, if a device is involuntarily turned on while navigating, the user shall turn it off again and continue with the sequence. The actual sequence performed by the two subjects, along with the performance metrics, are reported in Section 6.2.4.

6.2.2 Database results

The results obtained using the prerecorded EEG database play an essential role for validating the signal processing techniques. Inspecting the classification performance for the data of 17 subjects contained into the database, we identified the most effective feature extraction method and verified the feasibility of the pseudo-online protocol.

The first aspect under investigation is identifying the single most useful electrode for SSVEP. The aim is to confirm the results presented in [69]. In the paper, the authors state that occipital electrodes carry the most information about SSVEP; in particular, according to the paper Oz is the most relevant location, followed by O2 and O1. We used PSDA method to compare the classification accuracy and recognition time using only one EEG channel, since PSDA is the only univariate feature extraction method employed in this thesis. Hence, tests using singularly Oz, O2 and O1 were performed. Windows of length equal to 3 and 4 seconds are analyzed and their results averaged. Figure 6.4 shows summarizing diagrams of the results. Looking at the average accuracy, it is immediate to see that Oz offers more precise classification, both for 3 and 4 second windows. Furthermore, Oz appears also to allow, on average, faster recognition of the dominant SSVEP frequency. Therefore, we can conclude that Oz is the most useful channel for univariate SSVEP recognition by means of PSDA.

The investigation now turns on finding out the most efficient technique among PSDA, CCA and MSI. Being CCA and MSI multivariate methods, they can process multiple EEG channel simultaneously. Thus, the input for these methods are the occipital channels Oz, O1 and O2 together, or a subset of the three. In order to obtain a single value for accuracy and recognition time, we averaged the values corresponding to 10 seconds trials with the ones corresponding to 5 seconds trials. Note than several window lengths and different channel combination have been tried, so there are three "degrees of freedom": the length of windows, the employed channels and the feature ex-



(a) (b)

FIGURE 6.4: Preliminary univariate analysis on the three occipital electrodes in order to identify the single most useful channel. The data used comes from the prerecorded EEG database. The boxplots in (a) summarize the distribution of classification accuracy among the 17 subjects, while the diagram in (b) is about recognition time.

traction method. Firstly we focus on the processing algorithm and channel selection. Therefore, just 4-second windows are initially considered.

Figure 6.5 and 6.6 show respectively accuracy and recognition time results obtained by processing database information. It comes immediately from the figures that PSDA method is outperformed by the other two, both with respect to recognition precision and speed. Instead, the differences between CCA and MSI are more subtle. Excluding the (Oz,O1) configuration, the median values for accuracy are both 100%. The boxplots show a similar distribution for both (Oz,O2) and (Oz,O1,O2) configurations. Nonetheless, MSI method appears to guarantee an higher Q1 value, i.e., the middle number between the smallest accuracy and the median. Furthermore, for the subject with poorest performance (the outlier), MSI proves better accuracy and faster recognition time. For the reasons explained, although there is no room to affirm that MSI definely outperforms CCA, these results indicate MSI as the method to prefer for our online SSVEP recognition. The channel selection remains an open problem, since Emotiv EPOC does not provide the Oz location.

Obviously, increasing the window length provokes an increment in the mean accuracy, but also slows down the system. The recognition time, indeed, plays

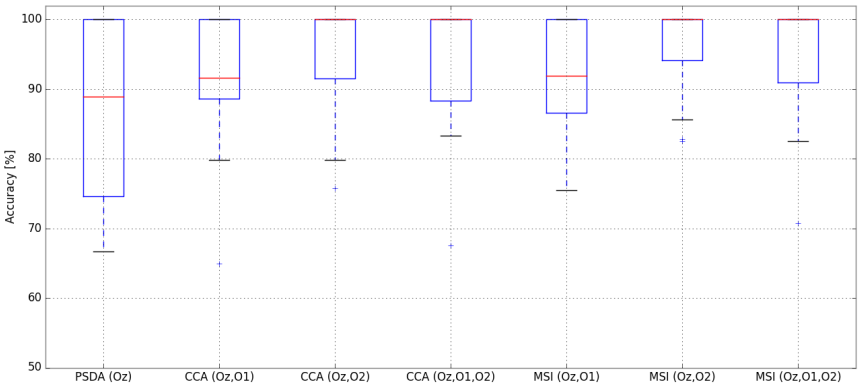


Figure 6.5: Classification accuracy distribution for 17 subjects. The different methods and channels considered are indicated on the x axis. Window length is fixed at 4 seconds. The data used comes from the prerecorded EEG database.

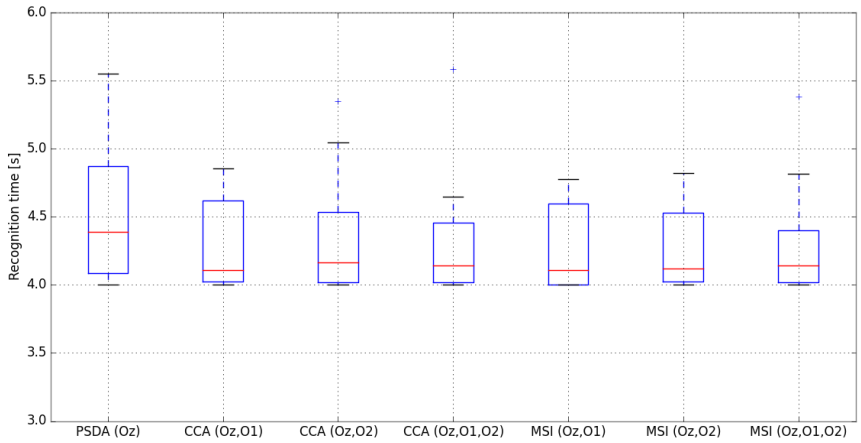


Figure 6.6: Recognition time distribution for 17 subjects. The shorter the time, the most efficient the system. Recognition time can not be smaller than the window length, which is fixed at 4 seconds. The data used comes from the prerecorded EEG database.

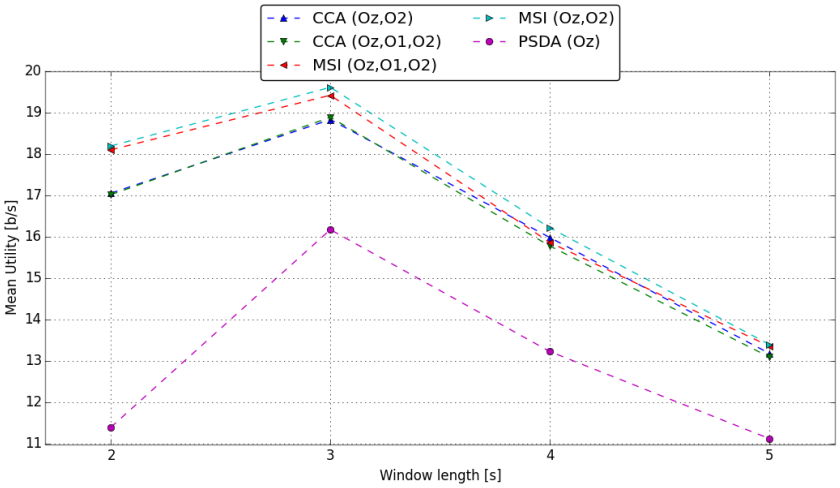


FIGURE 6.7: Impact of window length on Mean Utility for the different processing techniques. The data used comes from the prerecorded EEG database. The Mean Utility has been averaged among all subjects and it is expressed in bits per second.

an important role when computing overall performance metrics such as Utility or ITR. Therefore, we tried to identify the best choice for the window size according to Mean Utility value. To compute Utility, the formula requires a single value of accuracy; we do this by averaging the accuracy values previously computed. Thus, we obtain a single value of Mean Utility for each method and each window length. The comparison is plotted in Figure 6.7. All methods tested presents concordant results that indicate 3 seconds as the optimal choice for the window length. The diagram is particularly interesting because the reader can intuitively understand that smaller window length is not a guarantee of higher Utility. In fact, using 2 second windows it is more probable that the system is not able to provide any output at all, ending up with a lower Utility.

6.2.3 Offline results

Offline sessions recorded using Emotiv EPOC have been analyzed employing the knowledge acquired from EEG database results and with the final aim of making online BCI practicable. The main aspects to investigate are:

- What is the most effective feature extractor technique among the three considered?
- How to properly set the protocol parameters to ensure satisfying performance for most subjects?
- Is the system feasible for online signal processing and SSVEP recognition?

Following the same path of the previously described database analysis, we compared PSDA, CCA and MSI for different values of window length. The analysis is focused toward both obtaining optimal average performance and ensuring that all participants are able to use the system, even with reduced efficiency. With respect to PSDA, since the optimal channel Oz is not available with EPOC, we employed O1, which still exhibited good results in Figure 6.4. Instead, both O1 and O2 were employed with CCA and MSI, since they are the only occipital electrodes offered by EPOC and we want to exploit the feature of these algorithms of taking advantage from multivariate data.

Figure 6.8 depicts the average accuracy with respect to the window length. The diagram shows a behavior similar to the one found while analyzing the database: CCA and MSI outperform PSDA, but they present comparable accuracy with respect to each other. The average accuracy increases with larger windows, as expected. A considerable increase of average accuracy happens when passing from 5 second to 6 second windows. Analogous conclusion may be drawn observing the numerical results listed in Table 6.5, employing MSI. The table includes only 4 and 6 second windows to keep the presentation light. Since each subject performed two runs, the different metrics values have been averaged between the two runs. Observing the table, the reader is given another confirmation that performance varies among subjects. However, only subject #4 shows a substantial increase of accuracy with larger window length. On the contrary, the remaining subjects present a lower value of Utility with 6 second windows. We find again a greater ITR than Utility for high values

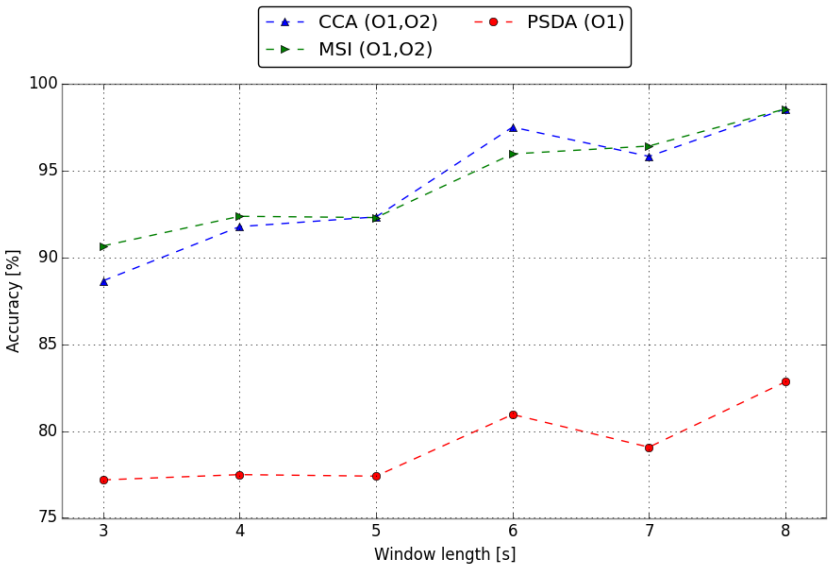


FIGURE 6.8: Average accuracy using Emotiv EPOC with respect to the window length considered. Different processing techniques are represented with different colors. The accuracy has been averaged among all trials performed by the four subjects.

of accuracy; recall how the ITR value is an unrealistic estimation of the real system capability, because it only takes into account the BST behavior.

Although accuracy value is essential to evaluate BST performance, the metric chosen in SMAD to express the overall system performance is Utility. Figure 6.9 summarizes how the Mean Utility varies against the window length. We can observe a monotonic decrease of Utility for larger windows. The reason for this behavior is in the relation between recognition time and Utility: the longer the time needed to produce an output, the less benefit the user obtains by using the system. Even if larger windows ensure better accuracy, the corresponding increase of recognition time affects the overall Utility negatively. This suggests that the optimal choice for the window length is 3 seconds. Albeit this value ensures the maximum Utility, we have followed a more conservative approach setting the window length for online sessions to 6 seconds. This decision is based mainly on the average accuracy (see Figure

Subject	Window length	SSVEP-based BCI				
		Accuracy [%]	Kappa	Duration [s]	ITR [b/m]	Mean Utility [b/m]
#18	4 sec	91.77	0.88	4.6	19.1	13.73
	6 sec	91.82	0.88	6.31	14.17	10.04
#19	4 sec	100	1.0	4.05	29.63	18.62
	6 sec	100	1.0	6	20	12.57
#20	4 sec	94.68	0.92	4.68	20.8	14.41
	6 sec	94	0.91	6.3	15.08	10.54
#21	4 sec	83.06	0.75	4.32	15.05	11.52
	6 sec	98.08	0.97	6.14	18.12	11.82
Average	4 sec	92.38	0.89	4.41	21.14	14.57
	6 sec	95.97	0.94	6.19	16.84	11.24

Table 6.5: Performance metrics for each subject performing offline SSVEP sessions with Emotiv EPOC. Data obtained using MSI as feature extraction technique.

6.8), since one of the basic assumption of SMAD is to minimize the number of erroneously performed actions. Furthermore, offline evaluation does not take into account user fatigue, gaze switching and other factors that may worsen performance with prolonged use of SSVEP-based BST. Nevertheless, online experiments with reduced windows would be an interesting extension of this work.

In conclusion, for online recognition it is necessary to verify that the computational effort required to process the signal is not excessive. In other words, we need to ensure that an entire EEG window takes less than one second to be processed. In fact, the system has one second to decide whether to collect or not additional data. If the current window produces no output, the system has to consider an additional second of signal. Therefore, the processing procedure that checks whether the output can be provided for the current window needs to take less than one second, otherwise it would lag EEG data, which comes in a continuous stream. To examine this aspect, we simply measure the time needed by the processing algorithms to treat a window. The worst case happens for 8 seconds windows. However, the techniques employed requires a very reduced computational effort: on a PC with a i5 CPU @ 1.60 GHz,

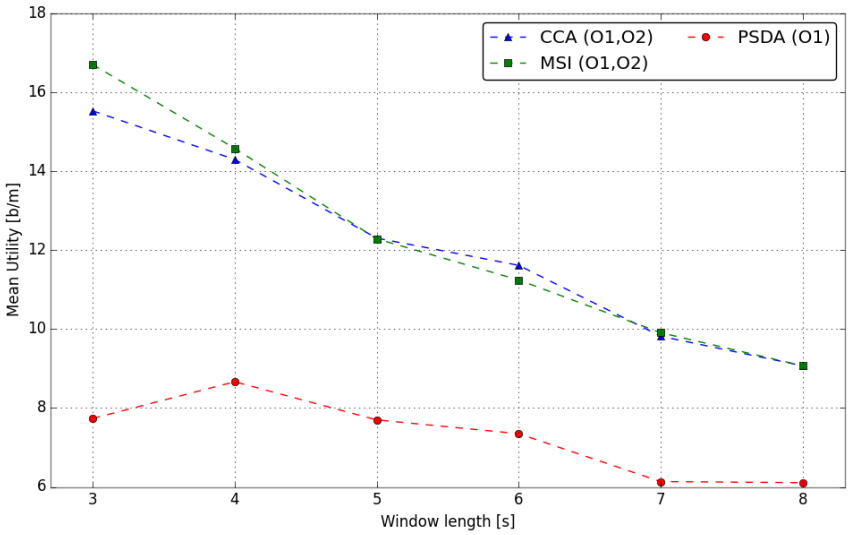


FIGURE 6.9: Relation between windows length and Mean Utility in offline recordings using Emotiv EPOC. Different processing techniques are represented with different colors. The Mean Utility has been average among all trials performed by the four subjects.

processing an 8 seconds window using CCA takes about 18 ms, whereas using MSI takes 19 ms and PSDA is even lighter (5 ms). Therefore, we can state that the system is feasible for online SSVEP recognition.

6.2.4 Online results

During online sessions, the subject is asked to perform a sequence of commands to simulate a common user scenario. In particular, the user should turn on lamp, fan, television and then act on TV channel and volume.

Clearly, not all trials have been interpreted correctly by the system. The actual command sequences triggered by the BST were:

Subject #19 : **N N D P P D N D D D N D D D N N D D N D N D N N P N N D**[1]

[1]Subject #19 involuntarily lose count of the *Volume up* commands and performed that command three times instead of twice. It has not been considered an error.

Subject	Sequence length	Commands performed	Correct trials	Accuracy [%]	Time	Actual bit rate [b/m]
#19	21	28	25	0.89	4'20"	5.77
#21	20	32	26	0.81	4'50"	5.21

Table 6.6: Experimental results of online SSVEP sessions for two subjects. The sequence of commands considered is a simulation of a common user scenario.

Subject #21 : **P N N N D D D P P D P N N P P D D D P N N D D N N D N N D P P D**

Table 6.6 summarizes the outcomes of the two online sessions. We can immediately observe that, in both cases, many more commands have been triggered than the ones comprising the optimal sequences. This is especially true for subject #22, who needs to perform 30% more commands than required. Such an issue is a natural consequence of the uncertainty related to biosignals pattern recognition: an error means not only contradicting the user intention, but also forcing the user to give additional commands to recover the error. This aspect is hardly captured by offline analysis; another reason why is it essential to involve online sessions while testing systems that involve biosignals interpretation. Furthermore, the accuracy has fallen considerably with respect to offline sessions. We can ascribe this worsening to the visual distraction the GUI entails and the gaze switching, which sometimes may involve a short period of imprecise SSVEP elicitation.

The single most important result in Table 6.6 is the bit rate value, computed using the Utility formula (4.1) with actual quantification of benefit and elapsed time. In other words, now we have the possibility of calculating b_d and T_d without any approximation: we know how much information has been expressed in the given period of time. Within the sequence, the number of operations expressing user intention are 8: three related to switching a device power state, three related to modifying a device state and two that affect the CI state. Recalling that the main menu contains $N = 5$ options while the television sub-menu contains $N = 6$ options, the total information expressed

is thus $b_d = 4 * \log_2 5 + 4 * \log_2 6 = 19.63$ bits. Concerning the time T_d, we ignore the resting period of 2 seconds between commands. Therefore, we have $T_d = 204$ seconds for subject #19 and $T_d = 226$ seconds for subject #21. The actual bit rate is finally obtained as $U = \frac{b_d}{T_d} * 60$.

The bit rate value is remarkably lower than the evaluation obtained through offline analysis (see Table 6.5). This situation can often be found in the literature. The lower accuracy with respect to offline estimation and the consequent increasing of commands required to fulfill the user intention are the main causes to the modest bit rate. Nevertheless, we consider these results as adequate to turn SMAD usable by means of cerebral activity only. In fact, despite other works – such as [47] or [45] – have presented much more performing online SSVEP-based BCIs, they rely on more sophisticated recording equipment. Furthermore, the almost totality of works related to BCIs reports the information bit rate computed using ITR formula; we have seen through this thesis that ITR is an optimistic overestimation of actual BCI performance. The work presented in [64] makes use of Emotiv EPOC and reports bit rates about four times larger than ours. However, the authors do not take into account CI behavior and employ ITR as performance evaluation metric.

Conclusions and future work 7

This thesis presented SMAD, a multimodal assistive system to control a smart environment by means of biological signals, or biosignals. The intended users of SMAD are individuals with severe motor impairments who desire to acquire more independence inside the domestic space. By using biosignals collected through a wireless headset, users can turn on and off various home devices such as lamp or television, and modify their property, for example changing tv channel.

The biosignals considered are EMG/EOG and EEG. This allows two kinds of interactions: one employing facial gestures and eye movements, the other exploiting cerebral activity (BCI). For the latter we use the SSVEP paradigm, which is based on the user gazing visual stimuli that keep repeating at a certain frequency. The same frequency is present in the subject's EEG signal.

The work started from a BCI-oriented point of view, with a literature review about general aspects and state of the art of SSVEP-based BCI. We found that many concepts proposed for BCI can be easily extended to other biosignals, including EMG and EOG. Thus, we defined the Biological Signals Transducer (BST), a component that allows HMI through biosignals independently of the specific type of biosignal.

An important advantage of SMAD is that biosignals are acquired using a consumer device, namely Emotiv EPOC. This device is categorized as *wearable device*, being relatively cheap, easy to use and fast to setup. With respect to very expensive traditionally used recording equipment, EPOC represents one step ahead for biosignals-based HMI to make its way out of the research laboratories.

Both interaction modalities have been designed to be reliable and have been tested during experimental sessions performed at the LAI of the Universidade Federal do Espirito Santo. Control using facial gestures and eye movements has been tested with eight healthy subjects. The results have been fulfilling, especially for three of them who reached optimal performance. The amount of information expressible through the system range from 14 bits/minute to 53 bits/minute.

For what concerns the BCI, EEG signal processing has been firstly developed and validated using a database recorded in previous works. The database contains SSVEP sessions of several subjects including some with disabilities. Then, other experiments have been carried out at LAI using Emotiv EPOC with few subjects, showing satisfying results. The offline analysis indicates that the users may express up to 15 bits/minute. The online experiments provided a more limited result, with about 5.5 bits per minute. Nevertheless, such outcomes have been obtained following a fully realistic reasoning, which is not always the case in BCI research.

With respect to future research, there is certainly room for further tests and improvements. Indeed, a research project is starting at LAI that intend to perform extensive experimentation with SMAD, initially using the EMG/ EOG-based BST. Several individuals with disabilities will be involved. The researches will bring the robotic wheelchair and the acrylic box to which connect the devices will be brought to the subject's house. There, under the supervision on an occupational therapist, the individuals will learn how to use the system and start to use it in everyday life. After a month, the researchers will come back to the subject's house and assess user impressions and system

performance.

Another promising idea to improve SMAD consists in expanding the scope of the system, from one room to the entire house. After implementing some kind of wheelchair motion control, several rooms may be equipped with sensors detecting when the wheelchair enters the room. Hence, SMAD would adapt the menu, i.e., the available devices, to the new ones available in the room.

Dealing with the EEG-based BST, several aspects may be evaluated to improve performance and usability. Firstly, the technique explained in Section 5.2 should be implemented to turn on and off the GUI and the visual stimuli. This would be of great help for the usability of the BST. Adjustments to frequency selection, spatial filtering and feature extraction methods are the most promising ones. More online tests are needed, since we have realized that results coming from offline analysis are likely going to be proved overoptimistic. Furthermore, BCI control needs to be extensively tested on individuals with disabilities, too.

A feature of SMAD is that it leaves open the possibility of developing new BSTs without affecting the CI module, as long as the BST respects the three-commands paradigm of the CI. Therefore, we strongly encourage the development of alternative BSTs, based on different BCI paradigms or other biosignals. Finally, since CI behavior is fundamental for the overall system performance, strategies can be evaluated to minimize the overhead introduced by the option scanning approach, i.e., the menu.

Bibliography

[1] Albert M Cook and Janice Miller Polgar. *Cook and Hussey's assistive technologies: principles and practice*. Elsevier Health Sciences, 2013.

[2] Claudia Zickler, Angela Riccio, Francesco Leotta, Sandra Hillian-Tress, Sebastian Halder, Elisa Holz, Pit Staiger-Sälzer, Evert-Jan Hoogerwerf, Lorenzo Desideri, Donatella Mattia, et al. A brain-computer interface as input channel for a standard assistive technology software. *Clinical EEG and Neuroscience*, 42(4):236–244, 2011.

[3] Jacob Rosen, Moshe Brand, Moshe B Fuchs, and Mircea Arcan. A myosignal-based powered exoskeleton system. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(3): 210–222, 2001.

[4] Changmok Choi, Silvestro Micera, Jacopo Carpaneto, and Jung Kim. Development and quantitative performance evaluation of a noninvasive emg computer interface. *Biomedical Engineering, IEEE Transactions on*, 56(1):188–191, 2009.

[5] Craig Chin, Armando Barreto, and Miguel Alonso Jr. Electromyogram-based cursor control system for users with motor disabilities. In *Computers Helping People with Special Needs*, pages 905–912. Springer, 2006.

[6] Richard S Snell and Michael A Lemp. *Clinical Anatomy of the Eye*. Wiley-Blackwell, 2nd edition, 1997.

[7] Zhiwei Zhu and Qiang Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1):124–154, 2005.

[8] A. Ubeda, E. Iañez, and J.M. Azorin. Wireless and portable eog-based interface for assisting disabled people. *Mechatronics, IEEE/ASME Transactions on*, 16(5):870–873, Oct 2011.

[9] Rafael Barea, Luciano Boquete, Manuel Mazo, and Elena López. System for assisted mobility using eye movements based on electrooculography. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 10(4):209–218, 2002.

[10] Chun Sing Louis Tsui, Pei Jia, John Q Gan, Huosheng Hu, and Kui Yuan. Emg-based hands-free wheelchair control with eog attention shift detection. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 1266–1271. IEEE, 2007.

[11] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[12] Xiaorong Gao, Dingfeng Xu, Ming Cheng, and Shangkai Gao. A bci-based environmental controller for the motion-disabled. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(2):137–140, 2003.

[13] Jd R Millan, Frédéric Renkens, Josep Mouriño, and Wulfram Gerstner. Noninvasive brain-actuated control of a mobile robot by human eeg. *Biomedical Engineering, IEEE Transactions on*, 51(6):1026–1033, 2004.

[14] José del R Millán, Frédéric Renkens, Josep Mouriño, and Wulfram Gerstner. Brain-actuated interaction. *Artificial Intelligence*, 159(1):241–259, 2004.

[15] Tiziano D'albis, Rossella Blatt, Roberto Tedesco, Licia Sbattella, and Matteo Matteucci. A predictive speller controlled by a brain-computer interface based on motor imagery. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 19(3):20, 2012.

[16] Ferran Galán, Marnix Nuttin, Eileen Lew, Pierre W Ferrez, Gerolf Vanacker, Johan Philips, and J del R Millán. A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9):2159–2169, 2008.

[17] Rossella Blatt, Simone Ceriani, Bernardo Dal Seno, Giulio Fontana, Matteo Matteucci, and Davide Migliore. Brain control of a smart wheelchair. In *10th International Conference on Intelligent Autonomous Systems*, 2008.

[18] Robert Leeb, Hesam Sagha, Ricardo Chavarriaga, and José del R Millán. A hybrid brain–computer interface based on the fusion of electroencephalographic and electromyographic activities. *Journal of neural engineering*, 8(2):025011, 2011.

[19] J d R Millán, Rüdiger Rupp, Gernot R Müller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kübler, Robert Leeb, et al. Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in neuroscience*, 4, 2010.

[20] T Freire Bastos-Filho, Fernando Auat Cheein, SM Torres Muller, W Cardoso Celeste, Celso de la Cruz, Daniel Cruz Cavalieri, Mário Sarcinelli-Filho, P Faria Santos Amaral, Elisa Perez, Carlos Miguel Soria, et al. Towards a new modality-independent interface for a robotic wheelchair. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(3):567–584, 2014.

[21] Sandra Mara Torres Müller, Wanderley Cardoso Celeste, Teodiano Freire Bastos-Filho, and Mário Sarcinelli-Filho. Brain-computer interface based

on visual evoked potentials to command autonomous robotic wheelchair. *Journal of Medical and Biological Engineering*, 30(6):407–415, 2010.

[22] J. Castillo, S. Muller, E. Caicedo, A. Ferreira de Souza, and T. Bastos. Proposal of a brain computer interface to command an autonomous car. In *Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC), 5th ISSNIP-IEEE*, pages 1–6, May 2014.

[23] Eric R Kandel, James H Schwartz, Thomas M Jessell, et al. *Principles of neural science*, volume 4. McGraw-Hill New York, 2000.

[24] Jane E Huggins, Patricia A Wren, and Kirsten L Gruis. What would brain-computer interface users want? opinions and priorities of potential users with amyotrophic lateral sclerosis. *Amyotrophic Lateral Sclerosis*, 12(5):318–324, 2011.

[25] Christoph Guger, Gunther Krausz, Brendan Z Allison, and Guenter Edlinger. Comparison of dry and gel based electrodes for p300 brain–computer interfaces. *Frontiers in neuroscience*, 6, 2012.

[26] Pascal Laferriere, Edward D Lemaire, and Adrian DC Chan. Surface electromyographic signals using dry electrodes. *Instrumentation and Measurement, IEEE Transactions on*, 60(10):3259–3268, 2011.

[27] Alexander J Casson, D Yates, S Smith, John S Duncan, and Esther Rodriguez-Villegas. Wearable electroencephalography. *Engineering in Medicine and Biology Magazine, IEEE*, 29(3):44–56, 2010.

[28] Eleanor Criswell. *Cram's introduction to surface electromyography*. Jones & Bartlett Publishers, 2010.

[29] MBI Reaz, MS Hussain, and Faisal Mohd-Yasin. Techniques of emg signal analysis: detection, processing, classification and applications. *Biological procedures online*, 8(1):11–35, 2006.

[30] HH Jasper. The ten-twenty electrode system of the international federation. the international federation of clinical neurophysiology. *Electroenceph. Clin. Neurophysiol.*, 10:371–375, 1958.

[31] George H Klem, HO Lüders, HH Jasper, and C Elger. The ten-twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology.*, Supplement 52:3–6, 1999.

[32] Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems. *Neuroimage*, 34(4):1600–1611, 2007.

[33] Jonathan R Wolpaw, Dennis J McFarland, Gregory W Neat, and Catherine A Forneris. An eeg-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78(3):252–259, 1991.

[34] Steven G Mason and Gary E Birch. A general framework for brain-computer interface design. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 11(1):70–85, 2003.

[35] B Blankertz, G Dornhege, M Krauledat, J Williamson, R Murray-Smith, KR Muller, et al. The berlin brain-computer interface presents the novel mental typewriter hex-o-spell. *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course*, pages 108–109, 2006.

[36] François-Benoît Vialatte, Monique Maurice, Justin Dauwels, and Andrzej Cichocki. Steady-state visually evoked potentials: focus on essential paradigms and future perspectives. *Progress in neurobiology*, 90(4):418–438, 2010.

[37] Christoph S Herrmann. Human eeg responses to 1–100 hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental brain research*, 137(3-4):346–353, 2001.

[38] Yijun Wang, Ruiping Wang, Xiaorong Gao, Bo Hong, and Shangkai Gao. A practical vep-based brain-computer interface. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 14(2):234–240, 2006.

[39] Cheng Ming and Gao Shangkai. An eeg-based cursor control system. In *[Engineering in Medicine and Biology, 1999. 21st Annual Conference and the 1999 Annual Fall Meetring of the Biomedical Engineering Society] BMES/EMBS Conference, 1999. Proceedings of the First Joint*. IEEE, 1999.

[40] Matthew Middendorf, Grant McMillan, Gloria Calhoun, Keith S Jones, et al. Brain-computer interfaces based on the steady-state visual-evoked response. *IEEE Transactions on Rehabilitation Engineering*, 8(2):211–214, 2000.

[41] Ming Cheng, Xiaorong Gao, Shangkai Gao, and Dingfeng Xu. Design and implementation of a brain-computer interface with high transfer rates. *Biomedical Engineering, IEEE Transactions on*, 49(10):1181–1186, 2002.

[42] Dan Zhang, Alexander Maye, Xiaorong Gao, Bo Hong, Andreas K Engel, and Shangkai Gao. An independent brain–computer interface using covert non-spatial visual selective attention. *Journal of Neural Engineering*, 7 (1):016010, 2010.

[43] A Cotrina, T Bastos, A Ferreira, A Benevides, J Castillo-Garcia, D Rojas, and A Benevides. Towards a ssvep-bci based on depth of field. In *Proceedings of the 6th International Brain-Computer Interface Conference, Graz*, pages 097–1–097–4. Graz University of Technology Publishing House, 2014.

[44] Zhonglin Lin, Changshui Zhang, Wei Wu, and Xiaorong Gao. Frequency recognition based on canonical correlation analysis for ssvep-based bcis. *Biomedical Engineering, IEEE Transactions on*, 53(12):2610–2614, 2006.

[45] Guangyu Bin, Xiaorong Gao, Zheng Yan, Bo Hong, and Shangkai Gao. An online multi-channel ssvep-based brain–computer interface using a

canonical correlation analysis method. *Journal of neural engineering*, 6 (4):046002, 2009.

[46] Ola Friman, Ivan Volosyak, and A Graser. Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 54(4):742–750, 2007.

[47] Ivan Volosyak, Hubert Cecotti, Diana Valbuena, and A Graser. Evaluation of the bremen ssvep based bci in real world conditions. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, pages 322–331. IEEE, 2009.

[48] Danhua Zhu, Jordi Bieger, Gary Garcia Molina, and Ronald M Aarts. A survey of stimulation methods used in ssvep-based bcis. *Computational intelligence and neuroscience*, 2010:1, 2010.

[49] Rosanne Zerafa, Tracey Camilleri, Owen Falzon, and Kenneth P Camilleri. Comparison of plain and checkerboard stimuli for brain computer interfaces based on steady state visual evoked potentials. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 33–36. IEEE, 2013.

[50] Jordi Bieger, Gary Garcia Molina, and Danhua Zhu. Effects of stimulation properties in steady state visual evoked potential based brain-computer interfaces. In *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2010.

[51] Patrick Drew, Rory Sayres, Katsumi Watanabe, and Shinsuke Shimojo. Pupillary response to chromatic flicker. *Experimental brain research*, 136 (2):256–262, 2001.

[52] Yijun Wang, Y-T Wang, and T-P Jung. Visual stimulus design for high-rate ssvep bci. *Electronics letters*, 46(15):1057–1058, 2010.

[53] Zhenghua Wu, Yongxiu Lai, Yang Xia, Dan Wu, and Dezhong Yao. Stimulator selection in ssvep-based bci. *Medical engineering & physics*, 30(8): 1079–1088, 2008.

[54] Sergio Parini, Luca Maggi, Anna C Turconi, and Giuseppe Andreoni. A robust and self-paced bci system based on a four class ssvep paradigm: algorithms and protocols for a high-transfer-rate direct brain communication. *Computational Intelligence and Neuroscience*, 2009, 2009.

[55] Rafał Kuś, Anna Duszyk, Piotr Milanowski, Maciej Łabęcki, Maria Bierzyńska, Zofia Radzikowska, Magdalena Michalska, Jarosław Żygierewicz, Piotr Suffczyński, and Piotr Jerzy Durka. On the quantification of ssvep frequency responses in human eeg in realistic bci conditions. *PloS one*, 8(10):e77536, 2013.

[56] Brice Rebsamen, Chee Leong Teo, Qiang Zeng, VMH Ang, Etienne Burdet, Cuntai Guan, Haihong Zhang, and Christian Laugier. Controlling a wheelchair indoors using thought. *Intelligent Systems, IEEE*, 22(2):18–24, 2007.

[57] L Kirkup, A Searle, A Craig, P McIsaac, and P Moses. Eeg-based system for rapid on-off switching without prior learning. *Medical and Biological Engineering and Computing*, 35(5):504–509, 1997.

[58] David E Thompson, Lucia R Quitadamo, Luca Mainardi, Shangkai Gao, Pieter-Jan Kindermans, John D Simeral, Reza Fazel-Rezai, Matteo Matteucci, Tiago H Falk, Luigi Bianchi, et al. Performance measurement for brain–computer or brain–machine interfaces: a tutorial. *Journal of neural engineering*, 11(3):035001, 2014.

[59] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[60] Bernardo Dal Seno, Matteo Matteucci, and Luca T Mainardi. The utility metric: a novel method to assess the overall performance of discrete

brain–computer interfaces. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(1):20–28, 2010.

[61] Saroj KL Lal, Ashley Craig, Peter Boord, Les Kirkup, and Hung Nguyen. Development of an algorithm for an eeg-based driver fatigue countermeasure. *Journal of Safety Research*, 34(3):321–328, 2003.

[62] Alan T Pope, Edward H Bogart, and Debbie S Bartolome. Biocybernetic system evaluates indices of operator engagement in automated task. *Biological psychology*, 40(1):187–195, 1995.

[63] Anton Nijholt, Danny Plass-Oude Bos, and Boris Reuderink. Turning shortcomings into challenges: Brain–computer interfaces for games. *Entertainment Computing*, 1(2):85–94, 2009.

[64] Yue Liu, Xiao Jiang, Teng Cao, Feng Wan, Peng Un Mak, Pui-In Mak, and Mang I Vai. Implementation of ssvep based bci with emotiv epoc. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*, pages 34–37. IEEE, 2012.

[65] Dennis J McFarland, Lynn M McCane, Stephen V David, and Jonathan R Wolpaw. Spatial filter selection for eeg-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394, 1997.

[66] Ola Friman, Ivan Volosyak, and A Graser. Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces. *Biomedical Engineering, IEEE Transactions on*, 54(4):742–750, 2007.

[67] S Orfanidis. Svd, pca, klt, cca, and all that. In *Optimum Signal Processing*, pages 1–77. 2007.

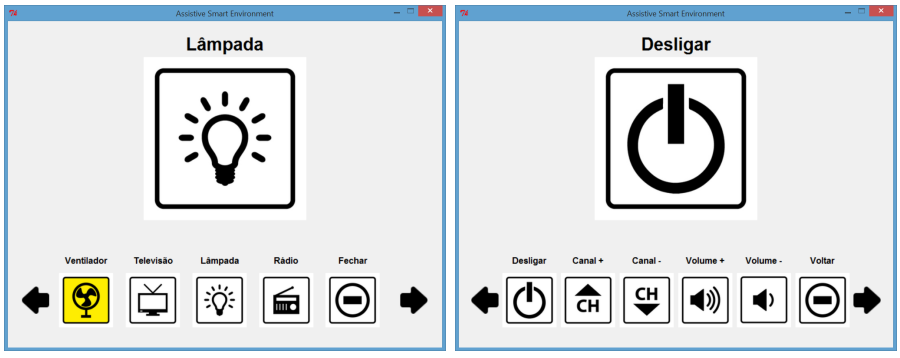
[68] Yangsong Zhang, Peng Xu, Kaiwen Cheng, and Dezhong Yao. Multivariate synchronization index for frequency recognition of ssvep-based brain–computer interface. *Journal of neuroscience methods*, 221:32–40, 2014.

[69] Sandra Mara Torres Müller, Teodiano Freire Bastos-Filho, and Mário Sarcinelli-Filho. Monopolar and bipolar electrode settings in ssvep-based brain-computer interface. *Journal of Medical and Biological Engineering*, in press.

Appendices

Menu panel file - an example A

This appendix provides the JSON code that realizes the main menu panel (a) and the television submenu (b) showed in the figure below.



(a) (b)

```
{
  "panels": [{
      "id" : 1,
      "items" : [
          { "id" : 10,
            "name" : "Fan",
             "imgOff" : "ventilador.jpg",
             "imgOn" : "ventilador - amarelo.jpg",
             "deviceOperation": "switch",
             "message" : "vnt"
             },
          { "id" : 11,
            "name" : "Television",
             "imgOff" : "tv.jpg",
             "imgOn" : "tv - amarelo.jpg",
             "deviceOperation": "device_on",
             "message" : "tlv",
             "panelId" : 2
             },
          { "id" : 12,
            "name" : "Lamp",
             "imgOff" : "luz.jpg",
             "imgOn" : "luz - amarelo.jpg",
             "deviceOperation": "switch",
             "message" : "lmp"
             },
          { "id" : 13,
            "name" : "Radio",
             "imgOff" : "radio.jpg",
             "imgOn" : "radio - amarelo.jpg",
             "deviceOperation": "switch",
```

```
            "message" : "rdo"
            },
          { "id" : 14,
            "name" : "Close",
            "imgOff" : "stop.jpg",
            "turnOff" : 1
            }
        ]
},{
  "id" : 2,
    "items" : [
          { "id" : 20,
            "name" : "Turn off",
            "imgOff" : "power.jpg",
            "deviceOperation": "device_off",
            "message" : "tlv",
            "panelId" : 1,
            "objectId" : 11
            },
          { "id" : 21,
            "name" : "Channel +",
            "imgOff" : "canal mais.jpg",
            "imgOn" :  "canal mais - amarelo.jpg",
            "deviceOperation": "device_modify",
            "message" : "chu"
            },
          { "id" : 22,
            "name" : "Channel -",
            "imgOff" : "canal menos.jpg",
            "imgOn" :  "canal menos - amarelo.jpg",
            "deviceOperation": "device_modify",
```

```
                "message" : "chd"
                },
            { "id" : 23,
              "name" : "Volume +",
                "imgOff" : "volume mais.jpg",
                "imgOn" :  "volume mais - amarelo.jpg",
                "deviceOperation": "device_modify",
                "message" : "vlu"
                },
            { "id" : 24,
              "name" : "Volume -",
                "imgOff" : "volume menos.jpg",
                "imgOn" :  "volume menos - amarelo.jpg",
                "deviceOperation": "device_modify",
                "message" : "vlu"
                },
            { "id" : 25,
              "name" : "Go back",
                "imgOff" : "stop.jpg",
                "panelId" : 1
                }
        ]
  }]
}
```