1 2 9 0

UNIVERSIDADE Đ COIMBRA

Maria Francisca Coelho Machado

Hybrid BCI based on visual and auditory stimuli

Dissertation supervised by Professor Doctor Urbano José Carreira Nunes and Professor Doctor Gabriel Pereira Pires and submitted to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra, in partial fulfillment of the requirements for the Degree of Master in Electrical and Computer Engineering, branch of Automation.

Supervisors:
Prof. Dr. Urbano José Carreira Nunes
Prof. Dr. Gabriel Pereira Pires

Jury:
Prof. Dr. Fernando Manuel dos Santos Perdigão
Prof. Dr. Paulo José Monteiro Peixoto
Prof. Dr. Urbano José Carreira Nunes

Coimbra, September of 2019

Acknowledgments

Firstly I would like to thank my supervisors, Professor Doctor Urbano Nunes and Professor Doctor Gabriel Pires, for giving me the opportunity to learn and work in such an interesting and complex area as the brain-computer interface, for the time they devoted to completing and solidify my academic degree, and the guidance and advice they provided me.

To Aniana Brito I would like to thank the constant availability and help.

I would also like to express a special thanks to my colleagues, friends, and family for the hours made available to me to test my work.

I thank to ISR for the excellent resources and conditions that allowed me to accomplish this final stage of my master degree. This dissertation has been financially supported by the Project B-RELIABLE: PTDC/EEI-AUT/30935/2017, with FEDER funding through programs CENTRO2020 and Portuguese foundation for science and technology (FCT).

A big and sincere thanks to my close family for providing me with all the conditions to complete my master degree and for all the support.

Abstract

A Brain-Computer Interface or BCI is a device that can extract brain activity, where brain signals are analyzed and processed to enable the machine to accomplish certain purposes, such as communicating, playing video games, controlling prosthesis, wheelchairs, smart homes, etc. In the past years, the study of human-computer interaction has been focused on how to improve the people's quality of life, with special attention on individuals with additional needs.

People with severe motor disabilities face many challenges in their daily life. During the last decades, the science community has been concerned about developing BCIs to provide means of communication and functional rehabilitation for these individuals. The efficiency of a BCI on a real-world application to help severely paralyzed individuals is still very limited, but the constant research and improvement on signal processing and machine learning techniques are bringing the possibility of efficient BCI use in real-world applications one step closer to reality.

The implementation of an electroencephalographic (EEG) based BCI, signal acquisition technique used in this work to record the electrical activity, has several aspects that need to be taken into consideration: usability, user comfort, methods, paradigms, etc. To improve the performance of the BCI system for a specific application, a particular protocol and paradigm have to be chosen for the experiment. It is essential to choose the right paradigm to enhance the neurophysiological signal, improving the classification process and therefore providing better performances.

The goal of this dissertation was to develop a standalone framework to implement visual and auditory (hybrid) paradigms in applications outside the Matlab/Simulink system (framework implementing signal acquisition and signal processing). The framework developed is not only a framework for the rapid development of feedback and stimulus applications but also a platform to run neuroscientific tests independent from BCI systems. Three conditions were proposed and developed in the context of BCIs based on a neurophysiologic signal designated by 'P300 event-related potential'. The paradigms were developed and designed in Python 3.7. The Matlab/Simulink and the Python framework run on the same computer and are synchronized through a TCP/IP communication. The Python module receives the information provided by the BCI and generates and presents the stimuli on a visual and auditory interface. The proposed paradigms use an auditory stimulus based on natural meaningful spoken words and a visual image-based stimulus (flashing words with overlapped pictures of well-known faces) simultaneously, aiming to enhance the evoked potentials, increasing stimuli discrimination, and reducing user's mental effort. Five stimulation paradigms were experimentally tested by 10 healthy participants: word flashing, word flashing with auditory, famous face flashing, famous face flashing with auditory, and flashing a relative's faces. The performance of the proposed BCIs was significantly improved in comparison to the control P300-based BCIs (word flashing and word flashing with auditory). The best online accuracy, effective SPM, and ITR was achieved by the relative's face paradigm and compares favorably with the state-of-the-art performances with 95.56%, 3.07 symbols per minute, and 8.08 bpm, respectively. Followed by the famous faces paradigms, which had similar performances with an accuracy of 93.33%. However, the control paradigms had a lower performance with an accuracy of only 82%. The relative's face paradigm showed to recruit additional face selectivity mechanisms in addition to those for non-relative's face, eliciting the most discriminative ERP features.

Convolutional Neural Networks (CNNs) were researched and implemented offline to classify P300 ERPs, to compare their performance with the current classification approach. Although there are not many studies regarding CNN for EEG classification it is a promising technique in this field as feature extraction and classification are one single architecture and optimized automatically with minimal preprocessing. This was an exploratory work with a limited number of datasets, where the CNN was trained with a small amount of data. The approach currently being used at ISR-UC showed better performances than the CNN, nevertheless, the CNN was capable of adequately classifying the data. Although the CNN performed worse than the current approach, the aforementioned CNN properties are very relevant and give the motivation to further explore deep learning approaches.

Keywords : Brain-Computer Interfaces (BCI), electroencephalogram (EEG), communication, event-related potentials, P300, locked-in state (LIS), hybrid visual-auditory, faces paradigm, Convolutional Neural Network (CNN).

Resumo

A interface cérebro-computador ou ICC é um dispositivo extrai a atividade cerebral, onde os sinais cerebrais são analisados e processados que permite que a máquina atinja certos funções como comunicação, jogar vídeo jogos, controlar próteses, cadeiras de rodas, casas inteligentes, etc. Nos últimos anos, o estudo da interação homem-computador está-se a concentrar em melhorar a qualidade de vida das pessoas, com atenção especial a indivíduos com necessidades especiais.

As pessoas com deficiências motoras graves enfrentam vários desafios no dia-a-dia. Nas últimas décadas, a comunidade científica tem-se preocupado em desenvolver ICCs que forneçam meios de comunicação e reabilitação funcional para esses indivíduos. A eficiência de uma ICC para uma aplicação no mundo real que ajude indivíduos gravemente paralisados ainda é muito limitada, mas a constante pesquisa e melhoramento em técnicas de processamento de sinais e machine learning estão a aumentar a possibilidade do uso eficiente de ICC em aplicações no mundo real.

A implementação de uma ICC eletroencefalográfico (EEG), técnica de aquisição de sinal usada neste trabalho para registrar a atividade elétrica, possui vários aspectos que necessitam consideração: usabilidade, conforto do utilizador, métodos, paradigmas etc. Para melhorar o desempenho do sistema ICC para uma aplicação específica, um protocolo e paradigma específico devem ser escolhidos para os testes. É essencial escolher o paradigma certo para melhorar o sinal neurofisiológico, melhorando consequentemente o processo de classificação e, portanto, proporcionar melhores desempenhos.

Esta dissertação tem como objetivo contribuir com um paradigma híbrido no contexto de ICCs, com base em um sinal neurofisiológico designado pelo ’potencial relacionado a evento P300’. Um paradigma híbrido (estímulos visuais e auditivos) foi desenvolvido em Python 3.7. A framework desenvolvida não é apenas uma framework para o rápido desenvolvimento de aplicações de feedback e estímulo, mas também uma plataforma para executar testes neurocientíficos independentes dos sistemas ICC. O Matlab / Simulink (framework que implementa a aquisição e processamento de sinal) e a framework Python são executados no mesmo computador e são sincronizados através de uma comunicação TCP / IP. O módulo Python recebe as informações fornecidas pelo ICC e gera e apresenta os estímulos numa interface visual e auditiva. Os paradigmas propostos usam estímulos auditivos baseados em palavras pronunciadas com significado natural e estímulos visuais baseados em imagens (palavras intermitentes com imagens sobrepostas de faces conhecidas) simultaneamente, destinado a melhorar os potenciais evocados, aumentar a discriminação de estímulos e reduzir o esforço mental do utilizador. Cinco paradigmas de estimulação foram testados experimentalmente por 10 participantes saudáveis: palavras, palavras com audição, faces de famosos, faces de famosos com audição, e faces de um parente/amigo. O desempenho das ICCs propostos foi significativamente melhorado em comparação com as ICCs de controle (palavra e palavra com audição). A melhor precisão online, ITR e SPM efetivo foram alcançados pela condição de faces de um parente e comparam-se favoravelmente aos desempenhos de state-of-the-art com 95,56 %, 3,07 símbolos por minuto, e 8,08 bpm, respectivamente.

Rede neural convolucional (ConvNet) foi estudada e implementada offline para classificar os ERPs P300, para comparar seu desempenho com a atual abordagem de classificação. Embora não existam muitos estudos sobre ConvNet para classificação de EEG, é uma técnica promissora neste campo, pois a extração de features e classificação são uma arquitetura única e otimizada automaticamente com o mínimo de pré-processamento. Este foi um trabalho exploratório com um número limitado de datasets, onde a ConvNet foi treinada com uma pequena quantidade de dados. A abordagem atualmente utilizada no ISR-UC mostrou melhores desempenhos do que a ConvNet, no entanto, a ConvNet foi capaz de classificar adequadamente os dados. Embora a ConvNet tenha um desempenho pior do que a abordagem atual, as propriedades da ConvNet acima mencionadas são muito relevantes e motivam a explorar ainda mais as abordagens de deep learning.

$Palavras - chave$: Interfaces Cérebro-Computador (ICC), eletroencefalografia (EEG), comunicação, potenciais relacionados a eventos, P300, síndrome do encarceramento, híbrido visual-auditivo, paradigma de faces, Rede Neural Convolucional (ConvNet).

List of Figures

List of Tables

Contents

1

Introduction

This chapter presents an introduction to this dissertation. Some insights concerning the context and motivation of the developed work are presented, as well as the main goals and key contributions.

1.1 Context and motivation

The body's actions and functions are all controlled by neurons which communicate information from the brain to the rest of the nervous system using electrical signals. These electric pulses, that are sent from the brain to the body, control everything we do daily. The understanding of how the brain controls our body led to the development of devices and algorithms that can be implemented on a computer that recognizes the patterns of brain signals.

A Brain-computer interface (BCI) is a device that captures brain signals, and then processes and interprets the information. This human-computer interaction enables brain signals to control external applications, such as communication devices, motorized wheelchair (locomotion), prosthetic devices, smart devices, play video games, or as a tool for neurorehabilitation of motor disorders (e.g., resulting from a stroke) or for neurorehabilitation of neurodevelopmental disorders (e.g., autism). BCIs have endless applications and are used in several fields, such as medical, entertainment, neuroergonomics and smart environment, neuromarketing and advertisement, educational, etc. The generic representation of a BCI in Fig. 1.1 shows some of these applications.

Human communication and interaction with the surrounding environment are two of the most important things in everyday life. People with severe motor disabilities face many daily challenges in their basic needs. Some individuals have their motor control severely affected, having no capacity of movement, or presenting very low dexterity, affecting their head, eyes, limbs, and speech. In the past years, several efforts have been made to develop BCIs to provide a reliable communication channel and restore some functional abilities, giving them the opportunity to improve somehow their interaction with people and the surrounding. Locked-in state (LIS) is a condition where the patients are fully aware but are unable to move or communicate. Almost all the voluntary muscles are in complete paralysis except vertical eye movements and blinking. The patients with this condition tend to lose the physical abilities over time and some can enter in a complete locked-in

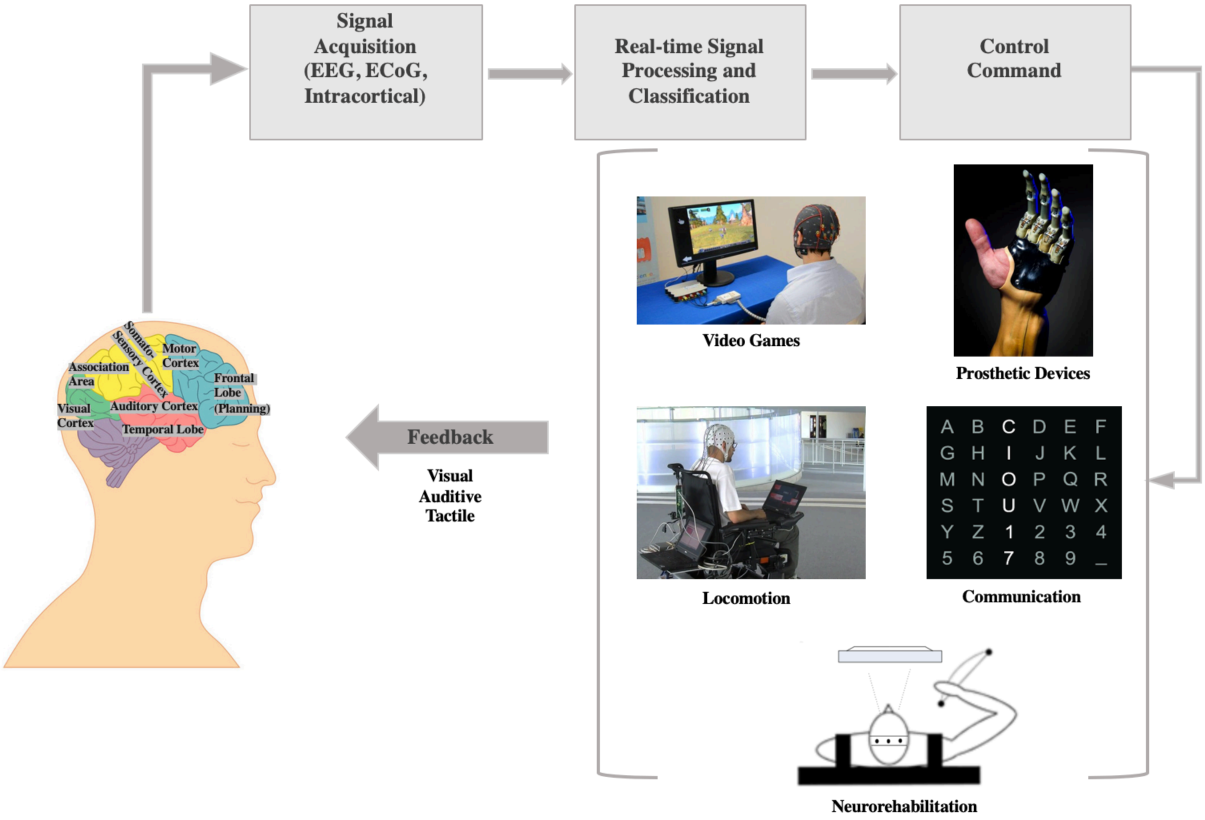


Figure 1.1: A generic representation of functional elements of a BCI system (figure inspired in [52]). The signals are recorded from the user's brain, the method used can be EEG, ECoG or intracortical; Brain signals are processed and classified in real-time and then adapted into control commands.

state (CLIS), losing the ability to control any muscle including their eyes but continue to have full awareness [4]. Since CLIS patients still produce and control brain signals it is possible to establish communication with them using BCIs. The efficiency of a BCI on a real-world application to help severely paralyzed individuals still has several limitations, but the constant research and improvement on signal processing techniques are bringing the possibility of an efficient BCI use on real-world applications closer to reality.

BCIs can be based on different neural signals, of which the electroencephalogram (EEG) is the most common. These interfaces are categorized based on the EEG brain activity patterns into four different types: P300, SSVEP (Steady-State Visual Evoked Potential), ERD (Event-Related Desynchronization) and slow cortical potential (SCP). These BCIs have both advantages and disadvantages. The SCP and ERD require long periods of training and with no guarantee of success, the P300 only needs a few minutes to train and SSVEP does not need training. Yet, SSVEP-based BCIs are very unpleasant due to continuous flashing stimuli and are gaze-dependent [47], so they cannot be used by patients with eye movement limitations. Additionally, the SSVEP is not evoked by some people. The P300 can be gaze-independent (covert attention), although the P300 usually has better results with gaze-dependent mechanism (overt attention) [12]. Therefore, while SSVEP-BCI generally yields the highest transfer rates, and while ERD-BCI is the only one that does not require any stimulation, P300-based BCI has so far shown the most suitable features for clinical use, and therefore is the most commonly used in clinical case studies with severely paralyzed patients [52]. Fig. 1.2 shows a comparison between the four BCIs regarding the training time and information transfer rate. The BCI developed in this work is based on the P300 neural mechanism, as the main goal of this dissertation is to design a BCI suitable to patients in CLIS states.

The Institute of Systems and Robotics - University of Coimbra (ISR-UC) has been researching methods to improve several aspects of BCI, regarding its information transfer, reliability, and general usability. To improve the signal-to-noise ratio of the P300 signal, a signal processing technique, the Statistical Spatial Filter, was designed in [58]. Also, a novel P300 layout paradigm, he Lateral Single character Speller (LSC), was developed to increase the evoked P300 potential amplitude to raise the discrimination to non-target events, by decreasing the target probability and the effect of adjacent distractors [59]. To allow patients to use the interface without ocular deviation (gaze independence), a new arrangement of P300 stimuli and a new paradigm strategy was proposed (GIBS speller) [57]. To decrease the user effort and improve usability of BCI to drive a robotic wheelchair, a human-machine shared controller was developed, supported by a motion planner that allows robust indoor navigation [50][51]. To reduce user effort, a method for detecting non-control states was implemented, which allowed the design of a BCI in which the user conveys commands only when desired [60]. More recently, an approach based on error related potentials (ErrPs) was implemented for BCI error detection and correction [14]. Currently, a technique for elimination of calibration or reduction of calibration time is in the process of being published, which will improve user usability.

In this thesis work, a hybrid (visual and auditory) P300 paradigm is proposed aiming to increase the discrimination of the evoked potentials. Also, the stimulation framework was developed as a standalone module so that it can be flexibly integrated with different BCIs through TCP/IP communication.

1.2 Goals

Several aspects need to be taken into consideration when developing an EEG-based BCI system such as the user needs, the paradigm, the methods, protocols, usability, etc. To improve the performance of the BCI for a specific application and make it user-friendly, a particular protocol and paradigm have to be chosen. Paradigm design is of great importance, as several features can be modeled to enhance brain waves, thus facilitating the classification process. Increase user's attention and engagement, decrease unpleasant and tiring effects, are factors that can contribute to a more effective BCI control.

P300 is a positive component of the event-related potential (ERP) evoked by a rare and relevant stimulus, the target event, in an oddball paradigm. After the target event appears, a positive peak around 300ms occurs [20]. It was first reported by Sutton et al. in 1965 [73]. P300-based BCIs require selective attention and working memory, hence depending on the complexity of the

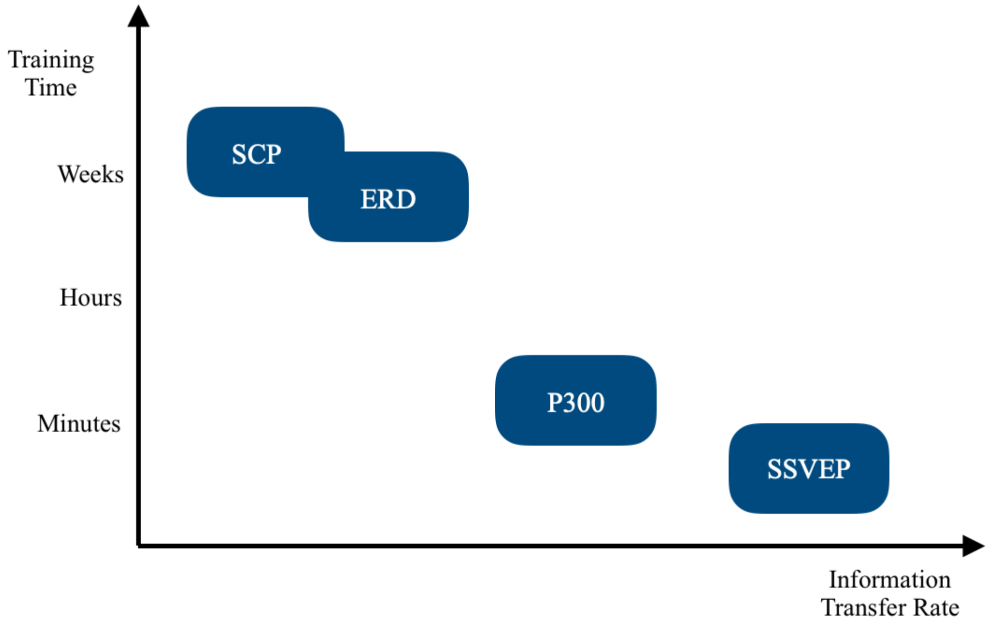


Figure 1.2: Comparison between SCP, ERD, P300, and SSVEP regarding the training time and information transfer rate [70].

paradigm [39]. CLIS patients are not capable of gazing peripheral stimuli, outside the foveal vision due to their visual impairment [4], and may experience reduced concentration spans and cognitive impairment, restricting them to successfully control highly difficult tasks [43] [47] [6] [67]. Consequently, P300-based BCIs paradigms need to take into consideration CLIS patients limitations, for example studies [68][26][17][23][61][28][54][7] show that using simpler and less choices for the paradigms improve the overall perform of the communication performance. Moreover, the scientific community has been researching strategies/paradigms to enhance the P300 signal and therefore increase the discrimination of the evoked potentials. Regarding the design of paradigms, a recent approach based on famous faces stimuli has been proposed as a promising technique to enhance ERP detection. Several studies in the field of human face perception show that visualizing a famous face elicits particularly strong ERPs (N170, P300, N400) [40]. Since CLIS patients usually have an intact hearing system, combining famous faces images and auditory stimuli, simultaneously, on a reduced lexicon paradigm may improve the overall BCI performance, which is the approach followed in this dissertation.

This dissertation aims to contribute with a hybrid paradigm in the context of BCIs based on a neurophysiologic signal designated by ’P300 event-related potential’. A hybrid paradigm, which combines visual and auditory stimuli, was developed and designed to try to overcome BCI inefficiency in patients with neurodegenerative disease. In this work the visual stimuli are famous faces (flashing words with overlapped pictures of well-known faces). The previous visual-auditory BCI framework developed at ISR-UC [3] was based on Matlab/Simulink (Highspeed Simulink, gtec), for visual stimulation (text-based) and signal processing and classification, and on Presentation software for auditory stimulation. Presentation software requires a commercial license and its connection to Matlab/Simulink requires a complex setup which limits the flexibility and usability of the overall BCI system. To overcome these limitations a new standalone framework was developed to present visual and auditory stimuli. The framework is fully independent of the rest of the Matlab/Simulink system, and it was designed with the Python 3.7 programming language. It is not only a framework for the rapid development of feedback and stimulus applications but also a platform to run neuroscientific experiments independent from BCI systems. The Matlab/Simulink and the Python framework run on the same computer and are synchronized through TCP/IP communication. This allows more flexibility and less complexity and also the possibility of connecting with different BCI applications. The Python module receives the information provided by the BCI and generates and presents the stimuli on a visual and auditory interface. The paradigm uses auditory stimuli based on natural meaningful spoken words and visual image-based stimuli concurrently, aiming to enhance the evoked potentials and increase the stimuli discrimination to improve the classification process. The suggested bimodal interface includes a reduced vocabulary comprised of the following words (same as in [3]): yes, no, hunger, thirst, urinate, air and position. The paradigm goal is to enable a basic ”yes or no” conversation, but also be fitted to provide the opportunity of some basic-needs/necessities options. The use of famous faces and natural meaningful spoken words is expected to help to capture the user’s attention and reduce user’s mental effort. On the other hand, the interface is not complex nor tiring, minimizing memory and cognitive demand, and maximizing the stimuli discrimination and intuitiveness. Although a BCI based on a reduced vocabulary can be quite limited when compared with other communication paradigms, it may represent a feasible BCI to be controlled by CLIS patients, taking into account their limitations. Figure 1.3 shows in red the BCI system blocks that were developed or enhanced in this work.

Another issue was addressed in this work, which was to research the possibility of improving the overall performance of the BCI system. In the past years, Convolutional Neural Networks (CNNs) have been proposed for EEG classification [13]. Nevertheless, there are not many studies regarding CNN application for EEG classification, but it is a promising technique in this field [13]. An advantage of CNNs is it performs automatic feature extraction. The feature extraction and classification parts are one single architecture and optimized automatically. CNNs also require minimal pre-processing techniques. In this dissertation, the performance of a CNN classifier and the approach currently being used at ISR-UC, based on a statistical spatial filter and a Bayes classifier, were compared.

1.3 Implementations and key contributions

The following implementations and contributions have been achieved:

- Standalone framework implementing visual and auditory paradigms in applications outside Matlab/Simulink, but keeping the acquisition and signal processing in Matlab/Simulink environment. Specifically, a data communication interface between Matlab/Simulink and external application, and a visual and auditory paradigm with a high-level programming language were developed;

- BCI laboratory experiments conducted with volunteers to test and validate the paradigms devel-

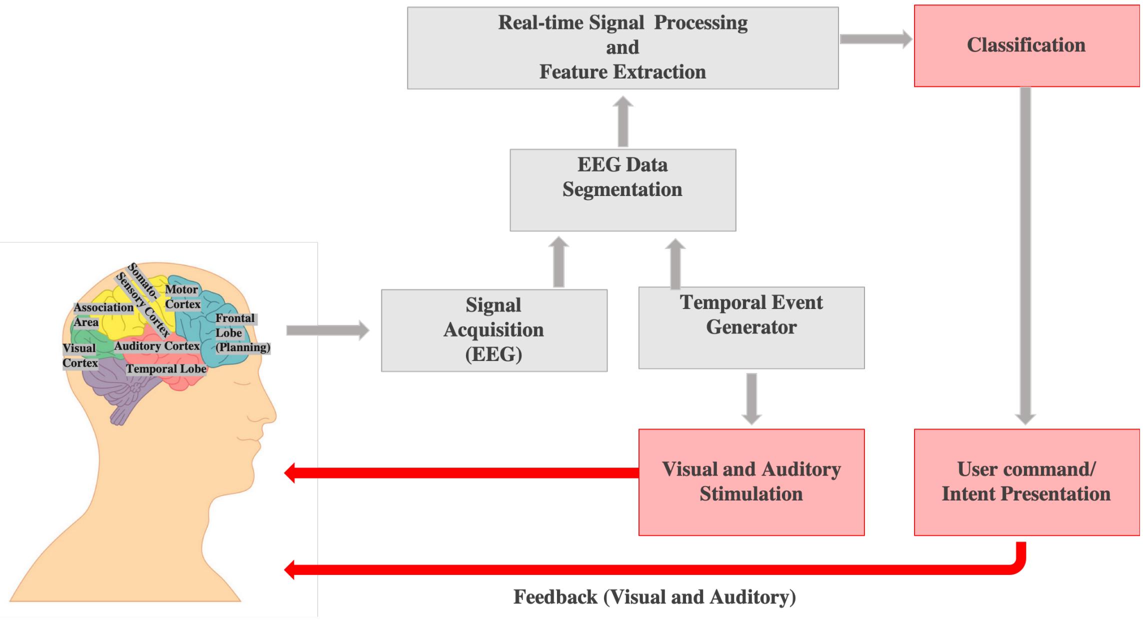


Figure 1.3: Representation of the BCI system used in this work. The red modules are the ones that contain the developed work made in this dissertation. The first module is the acquisition of the brain signals of the user which are used to extract the subject's intentions and also the responses to the stimuli. The next module of a BCI system does the signal pre-processing and classification. The stimulation module is where the stimuli are displayed on an interface that is presented to the user. The classification of the EEG signals is made to determinate the intention of the subject, the control command module provides the predicted results made by the classifier for each trial, as known as feedback, in the same interface where the stimuli are presented.

oped. Five stimulation paradigms were experimentally tested by 10 healthy participants: word flashing, word flashing with auditory, famous face flashing, famous face flashing with auditory, and flashing a relative's faces. A detailed neurophysiological study of the ERPs evoked by the five different paradigms was made;

- Application of classifiers to EEG neurophysiological data collected from experiments with the proposed paradigms;

- Offline implementation of a CNN classifier and performance analysis compared to the approach currently being used in ISR-UC, which combines a statistical spatial filter and a Bayes classifier.

The course flow of this dissertation and content of each chapter are the following:

- In Chapter 2 an explanation of some neurophysiologic concepts covered in this dissertation is made, also, the background material on the classification methods studied in this work is presented.
- Chapter 3 reviews the state of art on P300 paradigms, BCIs frameworks, and P300 detection using a convolutional neural network classifier.
- Chapter 4 describes the entire hybrid paradigm development environment, the functional elements performed by the P300-based BCI system, and the classification methods studied

in this work.

- In Chapter 5 a validation and discussion of the paradigms developed in this work are presented. Furthermore, the performance of the CNN classifier and the statistical spatial filter and Bayes classifier are discussed.
- Some final conclusions are made in Chapter 6.

2

Background material

This chapter is divided into two parts, the first section explains some fundamental neurophysiologic concepts to facilitate the comprehension of some topics covered in this dissertation. The second section describes the important background theory of two classification methods applied in this work.

2.1 Neurophysiological concepts

Brain signals can be recorded by different methods, from non-invasive to invasive approaches. Intracorticography is an invasive technique that requires a medical procedure, where the signal acquisition devices are microelectrodes arrays implanted under the skull within the grey matter. This is a highly invasive procedure with risks of infection involved. Electrocorticographic (ECoG) is a invasive technique, where the signals are acquired under the skull on the surface of the brain (exposed surface of the cortex), therefore an intracranial invasive medical procedure is needed. Electroencephalography (EEG) is a very popular non-invasive method that was first discovered by Hans Berger in 1929 [36]. Electrodes are placed on the surface of the scalp usually with gel to improve the conductivity. The electrodes capture the electric signals from the brain to the device that measures and records the signals. This a simpler, cheaper and safer technique that can be conducted easily compared to invasive techniques [34]. EEG provides millisecond-level temporal resolution, being one of the most direct measures of covert mental operations in the brain. EEG headsets with 10 to 20 electrodes are adequate for most academic or commercial needs. EEG caps have electrodes which are placed according to International 10-20 system [33]. BCIs based on EEG have some limitations, namely, they are very susceptible to neurophysiologic artifacts, they have a limited frequency range and the spatial resolution and signal-to-noise ratio (SNR) are low. The number of studies with EEG-based BCIs has been increasing over the past years, and on the contrary, the number of studies using invasive methods has been decreasing, according to [34].

Different neural mechanisms can be used on EEG-based BCIs depending on the application, research, medical and clinical purposes. The most important experimental paradigms used on EEG-BCIs systems are motor and mental imagery, P300 event related potentials (based on visual, auditory or hybrid stimulation), and steady-state visual evoked potential (SSVEP). Each paradigm

	PROS	CONS	USUAL PACE
NON-MOTOR MENTAL IMAGERY & MOTOR IMAGERY	- does not need stimuli; - less sensitive to timing than P300.	- needs intensive training; - can be tiring; - hard to detect more than two classes.	- one prediction after ∼4 seconds of single class imagery.
VISUAL P300	- user does not need skill; - good for multiple choice selection; - can be gaze independent.	- many repetitions in each trial; - very sensitive to timing.	- one letter after two to ten of flashes.
SSVEP	- user does not need skill; - less sensitive to timing than P300.	- the flicker may quickly fatigue the user; - It may be difficult to reliably select between more than 3 elements; - gaze dependent.	- one prediction each few seconds of signal.
AUDITORY	- user does not need skill; - good for multiple choice selection.	- many repetitions in each trial; - very sensitive to timing.	- one target sound after tens of sounds.
HYBRID	- accuracy can be improved; - classification methods can use more BCI outputs.	- may be more complicated than a single paradigm; - may be more difficult to develop.	- depends on the combination of the paradigms chosen.

Table 2.1: Comparison of the different types of paradigms used on EEG based BCIs systems.

elicits different kinds of brain signals and each one has its pros and cons. P300 is an event-related potential (ERP) paradigm in which the user needs to be aware of a specific stimulus/event, which is usually a visual symbol/image or a sound known as the target. This target must appear randomly and with a low probability among other events, called standard events (oddball paradigm). When the visual event is highlighted or the sound occurs, a positive deflection occurs around 300 ms after the onset of the event (delay between stimulus and response), as shown in Fig. 2.1. Given the low signal-to-noise ratio (SNR) of the P300 signal it is usually necessary that the target event is repeated several times in order to gather several responses, to select the target symbol the user is focused on. By repeating many targets and non-targets the BCI can detect which is the target that the user was focused on. The P300 waveform is evoked in almost every people, which is a good property for its general use on BCI. The P300 paradigm is usually applied for communication applications, like spellers. The classifier of the BCI system predicts for each signal block (one or more epochs - EEG segments associated to each event) whether it is a P300 or a non-P300 (standard). SSVEP is a paradigm where a periodic brain response is evoked in the visual cortex by a repetitive presentation of a flickering (visual stimulus) steadily with a certain frequency. The classifier module distinguishes between the flickering frequencies. Motor imagery is the paradigm where the user imagines kinesthetic limb movements of the left hand, right hand, foot, tongue, and so on. During the motor imagery, distinct mu/beta event-related (de)synchronization phenomena are generally shown around the motor cortex, which can be used for the classification of an individual's intention. In the non-motor mental imagery paradigms, the user has to perform mental tasks, such as mental calculation, internal singing or speech, 3D figure rotation, spatial navigation, letter composing, etc [34]. In auditory paradigms, the subject has to focus on a specific sound stimulus, which elicits brain signals. The stimuli can be presented to elicit a P300 ERP or auditory steady state response (ASSR). Hybrid paradigms combine two or more kinds of stimulation or two or more types of biosignals (e.g., EEG and EOG) simultaneously. The number of studies with visual P300, hybrid and SSVEP paradigms have been increasing over the years, unlike the motor imagery paradigms, as shown in Fig. 2.1. In Table 2.1, a comparison between the different paradigms is presented.

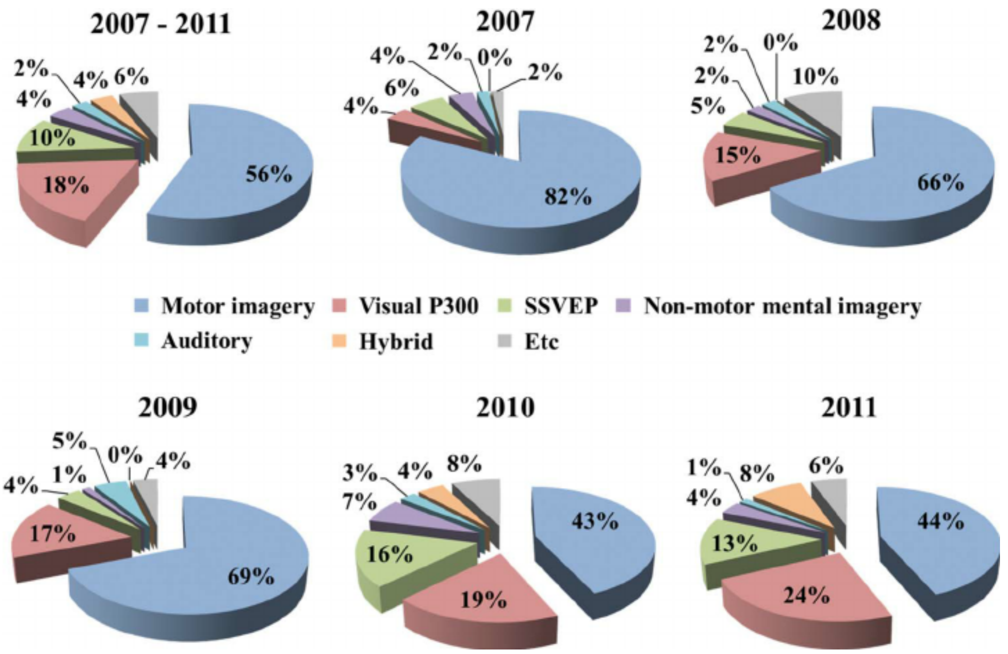


Figure 2.1: Percentage of each paradigm used in the published articles from 2007 to 2011 [34].

In this work, the developed BCI was based on a P300 paradigm. The oddball paradigm consists of having a rare target stimulus that is presented with other more frequent non-target stimuli in a serial input stream. The oddball paradigm elicits a positive deflection, P300, with a latency (delay between stimulus and response) of roughly 250 to 500 ms [20]. The P300 wave is an Event-Related Potential (ERP) component elicited in the process of a decision based on an evaluation or categorization of the stimulus. It is considered to be an endogenous potential since its occurrence links to a person's reaction to a relevant stimulus. The amplitude of the P300 wave is highly sensitive to the target stimulus probability and the level of attention of the subject. In the oddball paradigm, the number of repetitions of standards between two occurrences of a target is randomized, i.e., the target cannot be predicted by the occurrence of a given stimulus. Another ERP component elicited by the oddball paradigm is the mismatch negativity (MMN), which is evoked after an infrequent change in a repetitive sequence. The MMN is less dependent on the level of attention of the user than the P300.

Figure 2.2 shows the different ERP components produced in an oddball paradigm. The N100 or N1 is a negative evoked potential measured by electroencephalography and it peaks between 80 and 120 milliseconds after the target stimulus. This ERP is considered an exogenous response sensitive to the physical features (e.g., loudness or brightness) of an auditory, visual, or tactile stimulus, it has more recently been linked to word segmentation processes [64]. The N1 is often referred with the P200 (P2) evoked potential as the N100-P200 complex. The P200 component seems to be modulated by a vast and diverse number of cognitive tasks. It is known that the P200 is usually evoked as part of the natural response to visual stimuli. It has been investigated concerning visual search and attention, language context information, and memory and repetition effects. The N200 or N2 component is a negative wave that peaks at 200-350ms after the target stimulus, and it is typically evoked before the motor response, suggesting its link to the cognitive processes of stimulus identification and distinction. The N400 is a negative deflection that peaks around 250-500ms post target stimulus, it is part of the normal brain response to words/images and other meaningful stimuli like well-known faces, environmental sounds, smells, words, etc. The last ERP component is the P600, which is thought to be elicited by hearing or reading grammatical errors and other syntactic anomalies.

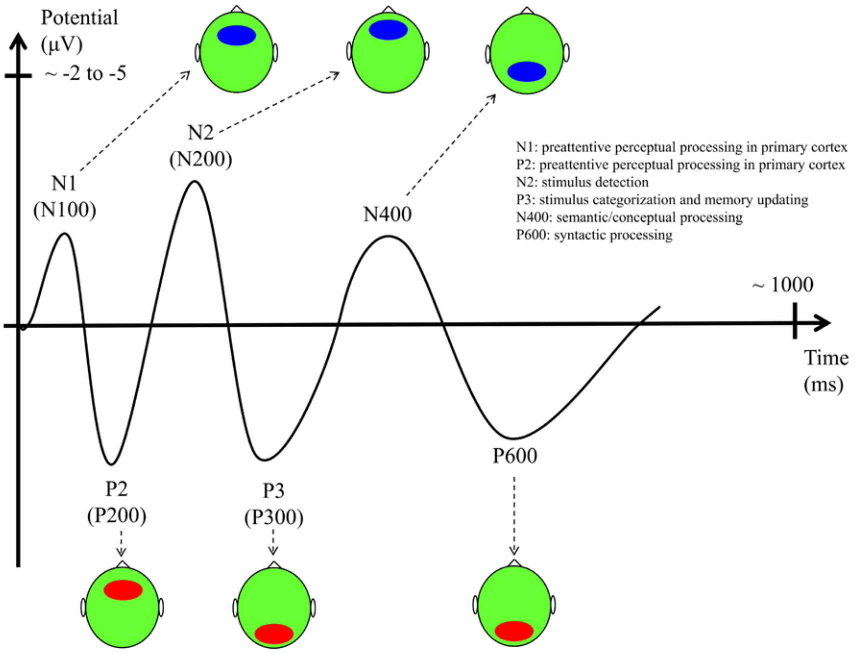

Potential
(µV)
~ -2 to -5
N1
(N100)
N2
(N200)
N400
P2
(P200)
P3
(P300)
P600
~ 1000
Time
(ms)
N1: preattentive perceptual processing in primary cortex
P2: preattentive perceptual processing in primary cortex
N2: stimulus detection
P3: stimulus categorization and memory updating
N400: semantic/conceptual processing
P600: syntactic processing

Figure 2.2: Waveform showing several ERP components, including the N400 and P300. Note: ERP is plotted with negative voltages upward [16].

2.2 Classification methods

This section describes two techniques used for the classification of event-related potentials that were tested offline or online in the context of the P300-based BCI system used in this dissertation. Concepts and methodological issues related to the methods are explained. The Bayes classifier is used after EEG signals of each channel are processed with a statistical spatial filter. After spatial filtering results a vector with the concatenation of the two most discriminative projections, which are then classified by the Bayes classifier. The CNN uses directly the EEG signals without this processing step.

2.2.1 Statistical spatial filter + Naïve Bayes

Linear classifiers used in P300-based BCIs, such as Naïve Bayes, require pre-processing and feature extraction before the classification. The pipeline combining spatial filtering and linear classification is shown in Fig. 2.3. The pre-processing and feature extraction steps are essential to increase SNR, hence they have a significant role in the performance of the subsequent classifier. Linear algorithms are sufficient for P300 classification, as long as appropriate feature extraction methods are applied [60]. The spatial filter used in this dissertation is the beamformer based on the Fisher Criterion (FC), proposed and developed in [60].

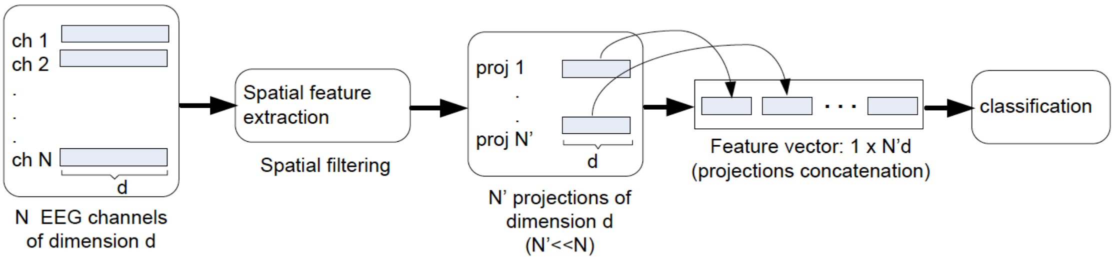

ch 1
ch 2
.
.
.
ch N
d
N EEG channels of dimension d
Spatial feature extraction
Spatial filtering
proj 1
.
.
proj N'
d
N' projections of dimension d (N'<<N)
. . .
Feature vector: 1 x N'd (projections concatenation)
classification

Figure 2.3: Pipeline of the statistical spatial filter + Naïve Bayes [60]

A spatial filter is a weighting vector, W, that combines the data of N channels at each time instant t. The spatial filter output is obtained, in the matrix notation, according to:

$$\mathbf{y} = W'\mathbf{X} \tag{2.1}$$

where $\mathbf{y}$ is the output projection obtained from input channels $\mathbf{X}$, and ' denotes the transpose operator. The FC is given by the Rayleigh quotient:

$$J(W) = \frac{W'\mathbf{S}_b W}{W'\mathbf{S}_w W} \tag{2.2}$$

where $\mathbf{S}_b$ is the spatial between-class matrix and $\mathbf{S}_w$ is the spatial within-class matrix. The optimum filter W is obtained determining the generalized eigenvalue problem:

$$\mathbf{S}_b W = \mathbf{S}_w W \Lambda. \tag{2.3}$$

The selected filter is the eigenvector associated with the largest eigenvalue. Taking the spatio-temporal matrix $\mathbf{X}_k$ (dimension $N \times T$) from each epoch k, the matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ are computed from

$$\mathbf{S}_b = \sum_i p_i(\overline{\mathbf{X}}_i - \overline{\mathbf{X}})(\overline{\mathbf{X}}_i - \overline{\mathbf{X}})' \tag{2.4}$$

and

$$\mathbf{S}_w = \sum_i \sum_{k \in C_i} (\mathbf{X}_{i,k} - \overline{\mathbf{X}}_i)(\mathbf{X}_{i,k} - \overline{\mathbf{X}}_i)' \tag{2.5}$$

where $i \in \{+, -\}$ and, C_+ and C_- represent respectively the target and non-target classes, and p_i is the class probability. The average of the epochs in class C_i and the average of all epochs are respectively denoted $\overline{\mathbf{X}}_i$ and $\overline{\mathbf{X}}$.

The Naïve Bayes (NB) algorithm is a probabilistic classifier based on the Bayes Theorem, which was created by Thomas Bayes [1701-1761]. NB is a particular case of the rule-based Bayes classifier, which considers independence between features. In the Bayes theorem the attribution of a given feature vector $\mathbf{y} = (y_1, y_2, ..., y_n)$ to a class variable C, is made according to [55]:

$$P(C|\mathbf{y}) = \frac{P(C)P(\mathbf{y}|C)}{P(\mathbf{y})} \tag{2.6}$$

The $P(w_k \mid X)$ is called the posterior conditional probability of class w_k, $P(X \mid w_k)$ the likelihood, $P(w_k)$ is the prior probability of class, and $P(X)$ the prior probability of predictor. The learning problem consists of estimating the posterior probability from the likelihood and prior probabilities. The likelihood $P(X \mid w_k)$ can be expressed using the chain rule:

$$\begin{aligned} P(\mathbf{y} \mid C) = P(\mathbf{y_1}, ..., \mathbf{y_n} \mid C) = \\ = P(\mathbf{y_1} \mid \mathbf{y_2}, ...\mathbf{y_n}, C)P(\mathbf{y_2} \mid \mathbf{y_3}, ...\mathbf{y_n}, C) \cdots P(\mathbf{y_{n-1}} \mid \mathbf{y_n}, C)P(\mathbf{y_n} \mid C) \end{aligned} \tag{2.7}$$

The previous sets of probabilities can be difficult and expensive to estimate. Fortunately, with the naïve conditional independence hypothesis, stated as:

$$P(\mathbf{y_i} \mid \mathbf{y_{i+1}}, ..., \mathbf{y_n} \mid C) = P(\mathbf{y_i} \mid C) \tag{2.8}$$

it is possible to express the likelihood as follows:

$$P(\mathbf{y} \mid C) = P(\mathbf{y_1}, ..., \mathbf{y_n} \mid C) = \prod_{i=1}^{n} P(\mathbf{y_i} \mid C) \tag{2.9}$$

Finally, the posterior probability can then be written as:

$$P(C|\mathbf{y}) = \frac{P(C)\prod_{i=1}^{n} P(\mathbf{y_i} \mid C)}{P(\mathbf{y})} \tag{2.10}$$

2.2.2 Convolutional Neural Network

To understand the main ideas, the key points, and the functionality of each layer within the convolution neural network (CNN) architecture, a detailed reading of [45] was made. CNNs are a group of deep neural networks that can utilize the spatial structure of data to learn about the data so that the algorithm can output something valuable. A neural network is based on a collection of connected units or nodes called artificial neurons, where the output of a neuron can be the input of another. A neuron is a placeholder for a mathematical function, and its mission is to provide an output by applying the function on the inputs provided. The function used in a neuron is called an activation function. The most common activation functions are step, sigmoid, tanh, and ReLU.

CNN trains and tests the data given to the network, the data passes through three distinct elements, the convolution layers with kernel filters, pooling layers, and fully-connected layers, as shown in Fig 2.4 [45]. The four main ideas behind CNNs are local receptive fields (locally connected layer), weight sharing across spatial positions using convolutional filters, spatial/temporal pooling, and the use of many layers. A typical CNN has two main parts. The first is responsible for feature extraction and consists of one or more pairs of convolution and pooling layers. The second part is just a classic fully-connected multilayer perceptron taking extracted features as input.

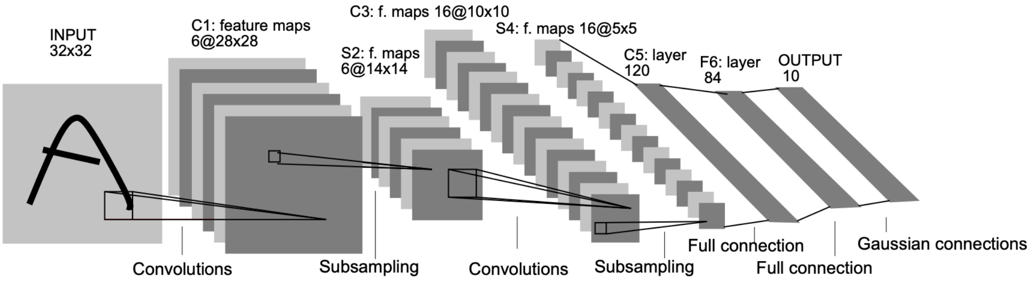


Figure 2.4: Example of a complete CNN architecture (LeNet-5) [45].

The convolutional layer is the first layer of a CNN. The convolution kernel operation in this layer outputs a high value for a given location if the convolution feature is present in that position, else outputs a low value. It is a process where a small matrix of numbers (called kernel or filter) passes over the image and transforms it based on the values from filter. The exact value is decided according to the following formula:

$$h_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} w_{k,l} x_{i+k-1,j+l-1} \tag{2.11}$$

where m is the kernel width and height, h is the convolution output, the row and column indices of the result matrix are marked with i and j respectively, x is the input data, and w is the convolution kernel. The kernels are convolved with the input data to obtain feature maps (convolution output). Feature maps indicate activated regions, i.e., regions where features specific to the kernel have been detected in the input. The kernel usually is a smaller-sized matrix in comparison to the input dimensions of the image, which consists of real-valued entries. For example, when we perform convolution over a 6x6 image with a 3x3 kernel, we get a 4x4 feature map. This is because there are only 16 (4x4) unique positions where we can place our kernel inside this picture. The convolution operation can be thought of as performing some transformation. If the input of the CNN is an image this transformation can result in various effects (e.g. extracting edges, blurring, etc.). The size and type of the kernel will depend on what the application of CNN is for since each filter captures different characteristics of the image. Typically the filter size is 3x3, but 5x5 or 7x7 are also used depending on the application.

The pooling layer, also called subsampling or downsampling layer, makes the CNN a little bit translation-invariant in terms of the convolution output while reducing the number of parameters. There are three subsampling methods used: max-pooling, sum-pooling, and average-pooling. The most common type of pooling is max-pooling. Max-pooling partitions the input into a set of non-overlapping rectangles and for each such sub-region, outputs the maximum value. In other words, this layer reduces the dimensionality, the number of parameters as well as the computational complexity but retains the important information. Besides, it helps to make the features robust against noise and distortion. So mathematically, for example for the max pooling:

$$h_{i,j} = max\{x_{i+k-1,j+l-1} \quad \forall \quad 1 \leq k \leq m \quad and \quad 1 \leq l \leq m\} \tag{2.12}$$

Contrary to the convolution operation, pooling has no parameters. It slides a window over its input, and simply takes the max value in the window. If the max-pooling is a 2x2 window and stride 2, this configuration halves the size of the feature map. This is the main use case of pooling, downsampling the feature map while keeping important information. Deciding the number of layers and other parameters of a CNN is a very subjective process, which usually involves a lot of trial and error. The easiest thing to do is pick a network that has been proven to work for a similar problem and train it for the task. Another approach is beginning with a very small model and gradually increasing the model size (adding layers and increasing the number of units per layer).

In the fully connected (FC) layer, the feature map matrix is converted into a vector, the features are combined to create a model. Then, an activation function classifies the outputs. The neurons in the FC layer will get activated based on whether multiple entities represented by convolution features are present in the inputs. The activated neurons will produce different activation patterns based on what features are present in the input data. This provides a compressed representation of what exists in the input data, to the output layer, that the output layer can easily use to accurately classify the data. In other words, the goal of the FC layer is to sum up the weights of the features coming from the prior layers and designate the probability of each class. For example, if there is a CNN for gender classification, and the output vector is a probability of [0.8, 0.2], it means there is an 80% probability of male gender and 20% for the female gender.

The CNN has to learn the convolution features given the input data. To do this, a cost function (e.g. root mean squared error, cross-entropy loss, etc.) is defined, which rewards the correctly identified data and penalize misclassified data. After the loss is defined, the weights of the features (neuron value of the features) can be optimized to reflect useful features that lead the CNN to accurately classify a specific object.

3

State of the art

This chapter is divided into three sections and reviews the state of the art that supports the developed work in this dissertation. The state of the art is focused on P300 paradigms, BCI platforms/frameworks, and P300 detection using a CNN.

3.1 Paradigms

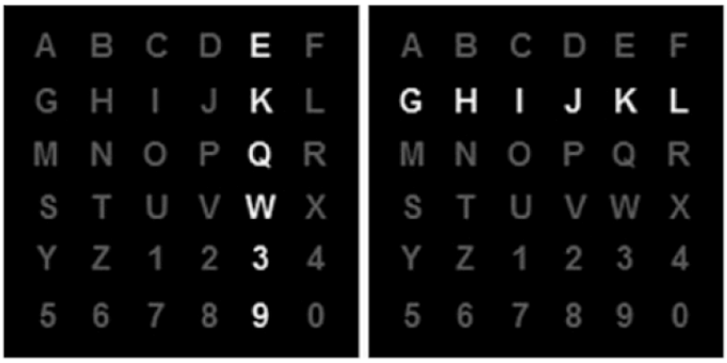
In the past few years, the number of publications related to P300-based BCIs has been increasing substantially. In this chapter, some of the most representative P300 paradigms currently available will be presented and analyzed.

3.1.1 Visual paradigms

Visual paradigms are usually displayed on a screen. The user observes the paradigm to interact with the BCI application by performing a given selective attention task, and receives visual feedback of the user's intention that was detected by the classification algorithm. In a P300 visual paradigm, the subject needs to focus on one specific object/image/character which is called the target stimulus among a set of other stimuli which are called the non-target stimuli. The P300 component appears in the EEG 300-600ms after the target stimulus is shown. Some important visual paradigms are here mentioned:

1. Row/column paradigm (RCP) [20]: RCP is based on a 6x6 matrix of characters (26 letters and 10 digits) where the rows and columns flash randomly. The RCP was the first important paradigm boosting the growth of the BCIs interfaces. Many studies were based in this paradigm, where the subject has to focus on a specific character and mentally count the number of occurrences of the target stimulus. The flashing row and column that contain the target character are expected to elicit a P300 component. Figure 3.1 shows a visual representation of the RCP.

2. Checkerboard paradigm (CBP) [75]: CBP uses an 8x9 matrix with 72 items, the matrix is placed over a checkerboard which the subject does not have a perception about (visual explanation in figure 3.2). The CB paradigm was created to overcome specific issues that

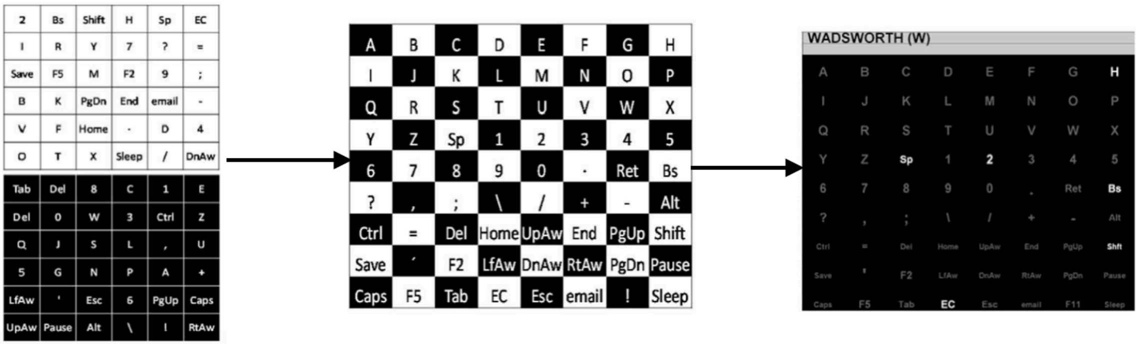


A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	0

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	0

Figure 3.1: RCP, the rows, and columns of the matrix flash in random order. The infrequent event (i.e., the row or column containing the item the BCI user wishes to select) has a 1/6 probability of appearing [25].

lead to errors in the RCP, like incorrect selections due to stimuli adjacent to the intended item. The items that are located in the white cells of the checkerboard are set into a white 6x6 matrix and the same goes for the items that are in the black cells. Before the group of items flash, the white or black matrix join random items. The result is random groups of six characters/items flashing. In [75], the main goal was to compare the CBP over the RCP, where the online results showed much higher accuracy in the CBP over the RCP. Not only the mean accuracy was higher in the CBP but also the number of sequences was lower. Regarding the waveform morphologies, the results differ in several aspects in which the CBP had higher amplitudes at 200ms and had a larger late negative peak.



2	Bs	Shift	H	Sp	EC
I	R	Y	7	?	=
Save	F5	M	F2	9	;
B	K	PgDn	End	email	-
V	F	Home	·	D	4
O	T	X	Sleep	/	DnAw

Tab	Del	8	C	1	E
Del	0	W	3	Ctrl	Z
Q	J	S	L	,	U
5	G	N	P	A	+
LfAw	'	Esc	6	PgUp	Caps
UpAw	Pause	Alt	\	!	RtAw

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	Sp	1	2	3	4	5
6	7	8	9	0	·	Ret	Bs
?	,	;	\	/	+	-	Alt
Ctrl	=	Del	Home	UpAw	End	PgUp	Shift
Save	'	F2	LfAw	DnAw	RtAw	PgDn	Pause
Caps	F5	Tab	EC	Esc	email	!	Sleep

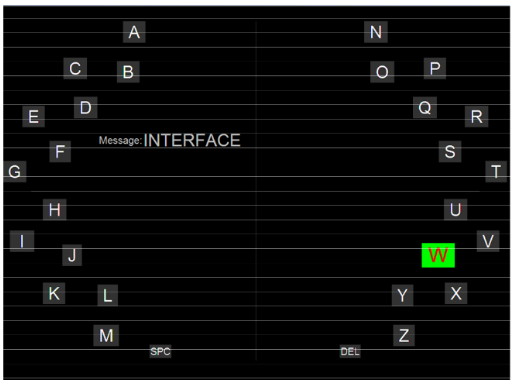
WADSWORTH (W)

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	Sp	1	2	3	4	5
6	7	8	9	0	.	Ret	Bs
?	,	;	\	/	+	-	Alt
Ctrl	=	Del	Home	UpAw	End	PgUp	Shft
Save	'	F2	LfAw	DnAw	RtAw	PgDn	Pause
Caps	F5	Tab	EC	Esc	email	F11	Sleep

Figure 3.2: CB paradigm, the left image is the checkerboard pattern. In the middle are the two virtual 6x6 matrices derived from the checkerboard. Finally, the right image is the matrix as presented to the subject with the top row of the white 6x6 virtual matrix flashing [74].

3. Lateral single-character Speller paradigm (LSC) developed at ISR-UC [59]: in the LSC paradigm all letters of the alphabet follow an event strategy that significantly reduces the time for symbol selection (eliminating the inter-symbol interval), reduces target probability (single character flash instead of a group) and reduces adjacent distractor effects which are very common on the RC speller. In [59] RC and LSC paradigms were tested with a healthy group and a clinical group with motor disabilities (people with amyotrophic lateral sclerosis

and cerebral palsy), where the LSC paradigm had significantly higher mean accuracy. The two paradigms produced different waveforms and the signal-to-noise ratio was also significantly higher for LSC. A screenshot of the paradigm is shown in Fig. 3.3.



Message: INTERFACE

Figure 3.3: Screenshot of the ISR-UC LSC speller [59].

4. Faces paradigms. Table 3.1 shows a brief summary of relevant studies that use the faces paradigm:

   - Familiar faces paradigm [41]: in this study the subjects were asked to send a personal known face, either family or friends, to compare the results with a famous face and an unknown face, using the same paradigm as in the Kaufmann et al. 2011 study [40]. This previous study consisted of using famous faces that the subjects could recognize as stimuli for flashing items in the character matrix of the P300-Speller. In [41] the results showed that no significant differences were found between the three conditions using faces, however, the classical character flashing (CF) and the face flashing (FF) showed some relevant differences. The amplitude of the P300 was much higher in the FF paradigm than the CF and an interesting occurrence was that the FF with the unknown face also evoked an N400f wave with high amplitude. This result is very interesting since the N400f wave is involved in face recognition and normally is observed for faces that are known (see Chapter 3.1: Neurophysiological concepts). In this study, experiments were conducted with 16 healthy subjects and 9 patients with neurodegenerative disease. As expected the patients had lower performances than the healthy subjects in the CF condition, however in the FF condition the differences were not so significant which means that the FF paradigm helps to overcome the difficulties and inefficiency in patients with neurodegenerative diseases.

   - Configural processing of human faces [78]: several experimental stimuli were studied with an oddball paradigm. The conditions proposed were: a face, an eyeless face and only eyes, where each condition was analyzed with the corresponding upright image and the inverted image. The inverted face condition obtained the best online classification accuracy. The P300 amplitude examination exhibited a notable difference among stimuli, larger P300 amplitudes were evoked by the face-related stimuli than by the highlight icon.

- Random set presentation and face familiarity [77]: three different spellers, all derived from the RCP speller, were analyzed. In these conditions a second modification was studied, self-face (image of the subject) and non-self-face images were used for stimulation. The paradigms using an image of a face show higher classification accuracies than the paradigms using flashing characters. Also, face stimuli show significantly intensified N400 waves when compared to flashing character paradigms. The self-face condition had a slightly (three percent) better classification accuracy than the non-self-face condition.

- Green Famous Faces Paradigm [48]: this study aims to examine if the performance of the P300 speller can be increased when the chromatic characteristic and the famous face paradigm are incorporated. Two conditions were studied: a picture of a famous face (FF) and a green picture of a famous face (GFF). The GFF paradigm showed higher online accuracy than the FF paradigm. Opposite to previous studies, the FF and GFF paradigms did not elicit the N400.

3.1.2 Auditory paradigms

Since CLIS patients have several limitations, like the inability to control their eye muscles to gaze visual targets (as mentioned in Chapter 1.1), an auditory stimuli is an alternative solution since it does not rely on gaze. Halder et al. (2013) [24] proposed a five-by-five matrix. The speller has 25 letters where different auditory stimuli are assigned to rows and columns. In [24], it can be noticed that in the auditory paradigm the ERPs have lower amplitudes and longer latencies. Höhne et al. (2012) [31] used three spatial directions for sound presentation. In the study three conditions were compared: artificially tones, spoken and sung syllables. From the experiments, it was verified that the spoken and singed syllables were easier to focus than the artificial tones. Schreuder et al. (2010) [66] used five speakers and five tones at the ear's subject level. The speakers covered 180º of the face area and were placed a meter from the subject. Halder et al. (2010) implemented a binary oddball paradigm by using three stimuli [23], with different variables such as pitch, amplitude, and direction. The best results were obtained with the pitch stimuli for most of the subjects. In [27] two binary paradigms were compared, one of the paradigms was based on spoken words and the other one on tones. Better performances were obtained using the spoken words, tested on healthy subjects. In [38] a comparison between an eye-tracking, an electrooculography system, and a binary auditory BCI in a LIS patient was made. It was demonstrated that a LIS patient can control the three studied cases with accuracy higher than 70%.

Paradigm	Participants	Stimuli properties	Performance (repetitions)	
			Offline classification	Online accuracy
Familiar faces [41]	10 healthy + 9 ND*	personally known face flashing speller (36-choice)	-	77.14% (2)
		famous face flashing speller (36-choice)	-	88.71% (2)
Configural processing of human faces [78]	7 healthy	face image upright flashing direction command (8-choice)	-	85.4% (2)
		face image inverted flashing direction command (8-choice)	-	97.9% (2)
Random set presentation and face familiarity [77]	15 healthy	self-face flashing on random set presentation speller (36-choice)	90.7% (2)	-
		non-self-face flashing on random set presentation speller (36-choice)	93.7% (2)	-
Green famous faces [48]	17 heathy	famous face flashing speller (36-choice)	-	75.6% (4)
		green famous face flashing speller (36-choice)	-	86.1% (4)
Familiar faces	9 healthy	personally known face flashing in specific lexicon (7-choice)	-	95.6% (mean: 4.2)
	10 healthy	famous face flashing speller in specific lexicon (7-choice)	-	93.3% (mean: 4.3)

Table 3.1: Relevant studies of BCIs based on visual faces: results of offline and online sessions. *ND: neurodegenerative disease. The red row corresponds to the paradigms developed in this dissertation.

3.1.3 Hybrid paradigms

Auditory paradigms work well when it is a binary system, however when the paradigm has a higher number of choices it has limited results, therefore the study and experiments that combine the auditory stimuli with other modalities (hybrid paradigms) on BCIs have been increasing and showed some promising results. Table 3.2 shows a brief summary of relevant studies of hybrid paradigms.

1. Gaze-dependent visual and auditory tasks [15]: in this study, the subjects had a target face image and simultaneously a target audio stimulus. The followed protocol was based on the P300 visually evoked potential. The P300 wave showed a higher amplitude on the hybrid stimuli over the visual and appeared between 200-250 ms after the target stimuli. This study showed good results with a 96% average accuracy. However, the paradigm does not suit for CLIS patients since it demands visual overt control.

2. Four different gaze-independent paradigms [2]: the applied paradigms were 1) visual speller (V); 2) auditory speller (A); 3) a simultaneous auditory and visual stimuli (AV); 4) an alternating auditory and visual cues (V*A). Paradigm V is a gaze-independent speller with 6 symbols. The A paradigm used only 6 stimuli, short spoken syllables (left 'ti', left 'to', middle 'it', middle 'ot', right 'ti', right 'to') were sung with different pitches by three speakers (bass, tenor and soprano human voices) as stimuli. Paradigm AV corresponds to visual and auditory stimuli presented simultaneously. In paradigm AV was noticed that ERP components (P1 and N1) have the biggest impact, between 250-450ms, the P300 component can be observed in the conditions V and AV with no big differences, but condition A has no significant P300 component. The visual speller had the best offline accuracies and the worse was the auditory speller.

3. P300-based gaze-independent BCI developed at ISR-UC, which combines auditory stimuli and visual stimuli detected covertly by S. Barbosa et al. (2016) [3]. The conditions were: visual stimuli (VO-Visual Overt and VC-Visual Covert), auditory stimuli (AU) and a hybrid with both (HVA). The visual and the auditory stimuli were made with seven words oddball tasks. In this experiment, the VO condition had a higher amplitude of the wave P300 (23.6V), then the AU condition, followed by the HVA and finally the VC condition with a 20.0V value. The VO evokes not only the higher amplitude of the wave P300 but also of the N200; the AU task showed almost none N200 component. It can be noticed that the waveforms of HVA and VC are very similar although the VC has a higher amplitude, suggesting that in the HVA condition the visual stimuli have a bigger impact. The HVA condition leads to the best online performances having an online accuracy of 85.3%, and it was considered less demanding.

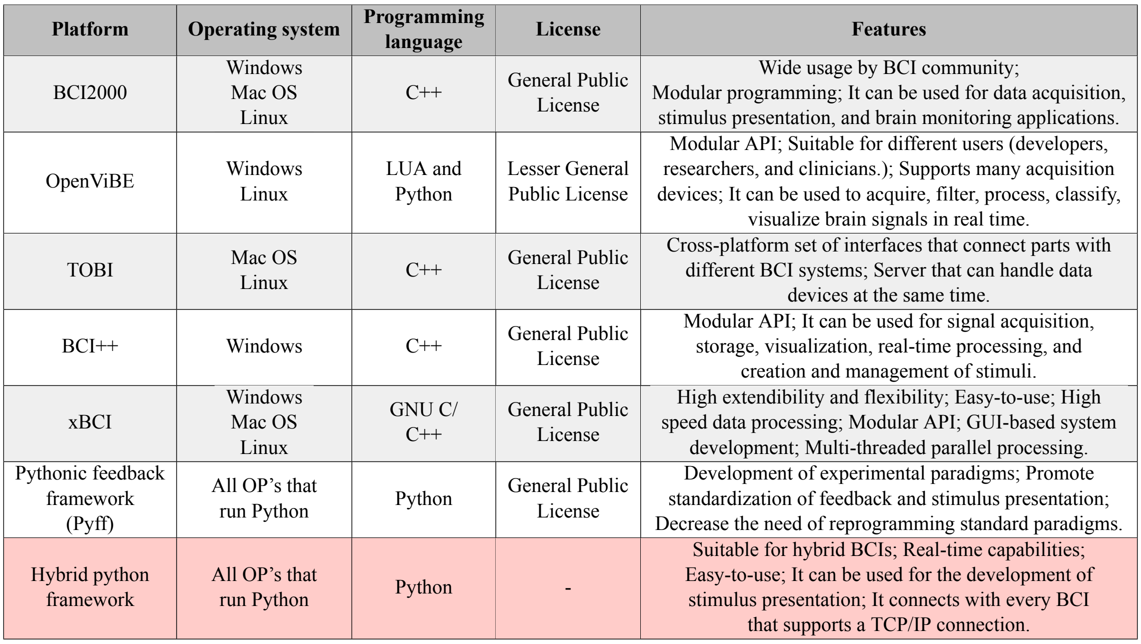
Paradigm	Participants	Gaze	Stimuli properties	Performance (repetitions) Online accuracy
Visual and auditory tasks [15]	3 healthy	Dependent	Face image + spoken syllables: Cued by number and arrow + “a”, “i”, “u”, ”e”, “o”, en”, “da” and “go” (8-choice)	96% (2)
Gaze-independent paradigms [2]	15 healthy	Independent	Visual symbols + spoken syllables: unique geometrical shape and color visual symbols + “ti” and “to” left, “ti” and “ot” middle, “ti” and “to” right, 3 pitches (6-choice/step, 2-step) Alternated visual symbols and spoken syllables: unique geometrical shape and color visual symbols and “ti” and “to” left, “ti” and “ot” middle, “ti” and “to” right , 3 pitches (6-choice/step, 2-step)	92% (6) 87.7% (10)
Auditory and visual stimuli detected covertly (HVA-BCI) [3]	10 healthy	Independent	Word image + spoken words: Words within a specific lexicon + Natural spoken words within a specific lexicon (7-choice)	85.3% (6)
P300-based gaze-independent auditory and visual face stimuli	10 healthy	Independent	Face image + spoken words: face image flashing within a specific lexicon + Natural spoken words within a specific lexicon (7-choice)	93.3% (mean: 4.3)

Table 3.2: Summary of relevant studies of hybrid (auditory and visual) based BCIs: results of online sessions. The red row represents the paradigm developed in this work.

3.2 Platforms for BCIs

Stimuli generation and presentation involves the development of a framework that has to be integrated on the BCI system. In this chapter we present several available frameworks that can be helpful in the design of the stimulation. An overview of publicly available software platforms for BCIs is done, identifying the advantages and disadvantages of each platform available on the market. One of the most popular commercial general-purpose platforms is the MATLAB/SIMULINK (The Mathworks, Inc.), from which it is possible to develop a variety of applications for scientific purposes. Software platforms specifically designed for the development of BCIs consist of the following blocks: data acquisition, processing, and feature extraction, classification and feedback presentation. Many BCI research labs use either existing modules from open generic BCI platforms or create new frameworks/modules fitted to their particular needs. In the past few years several BCI platforms were developed with the purpose to help people to build a BCIs, and to increase the BCI research community. Note that in Chapters 2.1 and 2.2, some background knowledge can be found to facilitate the comprehension of some topics. Table 3.3 contains a summary of open BCI platforms currently available.

In the Appendix A, BCI platforms are presented with more detail. The BCI2000 was considered, however, the programming language is C++. The syntax of C++ is complex and the standard library is small, making C++ difficult to learn for someone with little programming experience. Our goal is to have a framework of rapid development of feedback and stimulus applications and easy to use for everyone. Being this a goal, the platforms with programming languages with C++ were excluded. The Python language was considered to be the best option. Python is very easy to code as compared to other popular languages like Java and C++. It is a high-level language, meaning that



Platform	Operating system	Programming language	License	Features
BCI2000	Windows Mac OS Linux	C++	General Public License	Wide usage by BCI community; Modular programming; It can be used for data acquisition, stimulus presentation, and brain monitoring applications.
OpenViBE	Windows Linux	LUA and Python	Lesser General Public License	Modular API; Suitable for different users (developers, researchers, and clinicians.); Supports many acquisition devices; It can be used to acquire, filter, process, classify, visualize brain signals in real time.
TOBI	Mac OS Linux	C++	General Public License	Cross-platform set of interfaces that connect parts with different BCI systems; Server that can handle data devices at the same time.
BCI++	Windows	C++	General Public License	Modular API; It can be used for signal acquisition, storage, visualization, real-time processing, and creation and management of stimuli.
xBCI	Windows Mac OS Linux	GNU C/ C++	General Public License	High extendibility and flexibility; Easy-to-use; High speed data processing; Modular API; GUI-based system development; Multi-threaded parallel processing.
Pythonic feedback framework (Pyff)	All OP's that run Python	Python	General Public License	Development of experimental paradigms; Promote standardization of feedback and stimulus presentation; Decrease the need of reprogramming standard paradigms.
Hybrid python framework	All OP's that run Python	Python	-	Suitable for hybrid BCIs; Real-time capabilities; Easy-to-use; It can be used for the development of stimulus presentation; It connects with every BCI that supports a TCP/IP connection.

Table 3.3: Summary of relevant BCIs platforms. The red row is the platform designed and tested in this dissertation.

writing in Python does not require to remember the system architecture, nor to manage the memory, which makes it more programmer friendly. OpenViBE only works in Windows or Linux operating systems, which is an disadvantage. Also, OpenViBE is not capable of developing stimulation and presentation applications. Pyff framework was considered, however, the interface was considered not to be practical nor intuitive. For these reasons, a Python framework that suits all of our demands was developed from scratch.

3.3 P300 detection using Convolutional Neural Network

CNN's have the benefit of automatically learning feature representations from raw signals data, which may represent a good approach to learn features invariant/independent of subjects. Consequently, CNN's can increase the full potential of recognizing P300 waves. However, despite CNNs have been successfully used on image processing and speech recognition, their use for P300 detection is still incipient [11] [53] [49]. A CNN usual performs a spatial convolution on the raw data and then performs temporal convolution on the abstract data that came from the spatial convolution.

1. CCNN [11] has three stages and uses as input a matrix N x C, where N indicates the number of temporal signal samples and C represents the number of channels used for EEG signal recording. In the initial stage, it performs convolution along space to learn spatial features. In the following stage, a convolution along time of the abstract data that came from the spatial convolution is completed, to learn temporal features. The final stage uses fully-connected layers to obtain an accurate correspondence within learned features and a particular class. Table 3.4 shows the specifications of the CCNN.

Layer	Operation in a layer	Kernel Size	Feature Maps/ Neurons
1	Convolution	(1,C)	10
2	Convolution	(13,1)	50
3	Fully-Connected	-	100
Output	Fully-Connected	-	2

Table 3.4: CCNN architecture [11], the first column specifies the sequence of layers. The second column represents the operation in a layer. The third column describes the kernel size in the convolution layers. The last column specifies the number of feature maps/neurons in a layer.

Layer	Operation in a layer	Kernel Size	Feature Maps/ Neurons
1	Convolution Dropout	(N/15,C) -	16 -
Output	Fully-Connected	-	2

Table 3.5: OCLNN architecture [71], the first column specifies the sequence of layers. The second column represents the operation in a layer. The third column describes the kernel size in the convolution layers. The last column specifies the number of feature maps/neurons in a layer.

2. OCLNN [71] performs both spatial and temporal convolution in the primary layer rather than performing only spatial convolution as in CNN's quoted above and the input is an N x C matrix. This classifier can learn feature representations from raw temporal data and simultaneously can learn spatial features with only one convolution layer and without fully-connected layers before the output layer, which decreases the network complexity significantly. Table 3.7 shows the specifications of the OCLNN.

3. BN3 [49] combines Batch Normalization and Dropout techniques and the input is an N x C matrix. It uses Batch Normalization in Layer 1 and Layer 3. Batch normalization is a method we can use to normalize the inputs of each layer, to improve the speed, performance, and stability of a neural network. [49] also applies dropout in the fully-connected layers to decrease overfitting. Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. Before the output layer, BN3 employs two fully-connected layers rather than one for better generalization and accumulation of features. Table 3.5 shows the specifications of the BN3.

4. CNN-R [53] uses a deeper and wider network structure. Since it is a complex network, this classifier uses pooling and dropout techniques to reduce overfitting. The input is an N x C matrix. It uses smaller kernel sizes but more extra layers for temporal convolution. It also uses two fully-connected layers before the output layer. Plus, CNN-R uses more feature maps for the convolution layers and more neurons for the fully-connected layers than the CCNN. Table 3.6 shows the specifications of the CNN-R.

Layer	Operation in a layer	Kernel Size	Feature Maps/ Neurons
1	Batch Normalization Convolution	- (1,C)	- 16
2	Convolution	(20,1)	16
3	Batch Normalization Fully-Connected Dropout	- - -	16 128 128
4	Fully-Connected Dropout	- -	128 128
Output	Fully-Connected	-	2

Table 3.6: BN3 architecture [49], the first column specifies the sequence of layers. The second column represents the operation in a layer. The third column describes the kernel size in the convolution layers. The last column specifies the number of feature maps/neurons in a layer.

Layer	Operation in a layer	Kernel Size	Feature Maps/ Neurons
1	Convolution Pooling	(1,C) (3,1)	96 96
2	Convolution Pooling	(6,1) (3,1)	96 96
3	Convolution	(6,1)	96
4	Fully-Connected Dropout	- -	2048 2048
5	Fully-Connected Dropout	- -	4096 4096
Output	Fully-Connected	-	2

Table 3.7: CNN-R architecture [53], the first column specifies the sequence of layers. The second column represents the operation in a layer. The third column describes the kernel size in the convolution layers. The last column specifies the number of feature maps/neurons in a layer.

4

P300-based face and auditory hybrid paradigm

In this chapter, all functional elements comprising the BCI system are described, namely the classification architecture, experimental setup, software framework, protocols, and terminology. The experimental setup is shown in Figure 4.1. Signal processing and classification were developed with the Simulink/Matlab framework [Mathworks 2012b], using the 'Highspeed' Simulink g.USBamp driver for signal acquisition. The new modules developed in this dissertation were integrated on a hybrid-BCI based on visual (text) and auditory stimuli previously developed at ISR-UC [3].

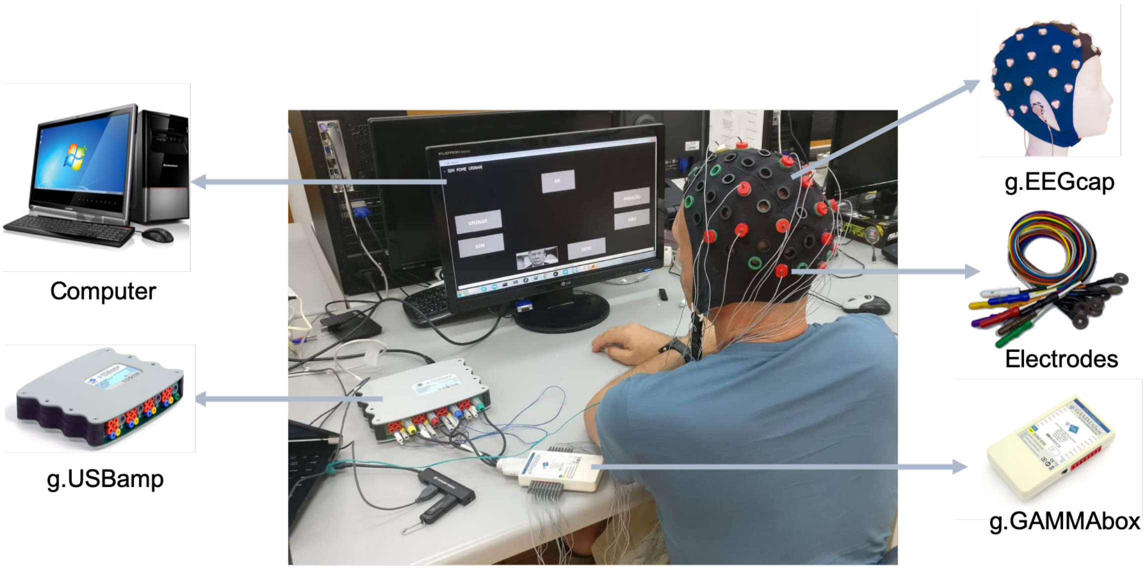


Figure 4.1: Picture of the experimental setup.

The synchronization between the signals recorded and events in a P300-based BCI system is crucial for the success of the experiments. The new stimulation module developed in this dissertation was implemented with Python 3.7 and is synchronized with the Matlab/Simulink framework through a TCP/IP communication. Simulink/Matlab and the Python framework run on the same computer. The data processing and storage were performed on a TOSHIBA laptop (Intel Core i7 2.30 GHz, 8.00 GB RAM, Microsoft Windows 7 Ultimate). The delay between event generation and stimuli presentation has to be constant and minimal. All algorithms in the framework are executed in real-time, triggered by the acquisition sampling rate of 256 Hz.

The functional modules of the BCI used in this work are represented in Figure 4.2. The developed work was a stimulation and presentation module. A new classifier based on a CNN was also implemented but it was only tested offline.

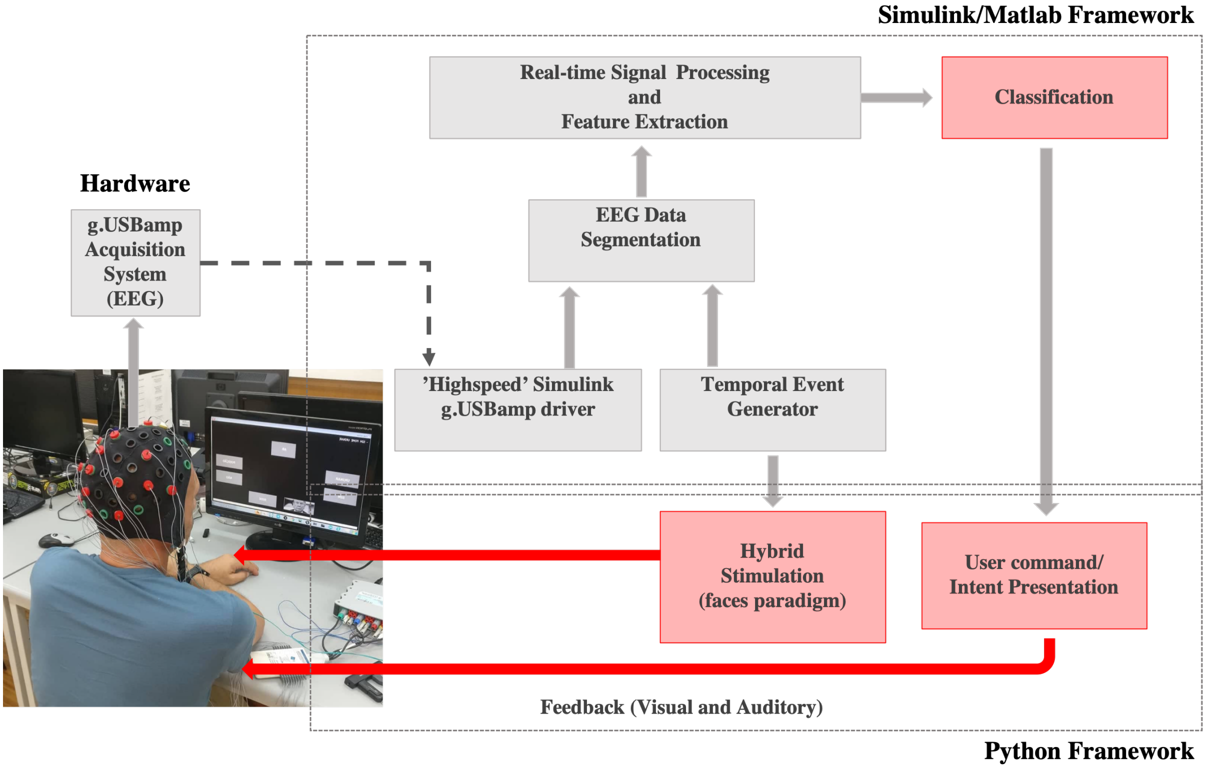

Simulink/Matlab Framework
Real-time Signal Processing and Feature Extraction
Classification
Hardware
g.USBamp Acquisition System (EEG)
EEG Data Segmentation
'Highspeed' Simulink g.USBamp driver
Temporal Event Generator
Hybrid Stimulation (faces paradigm)
User command/ Intent Presentation
Feedback (Visual and Auditory)
Python Framework

Figure 4.2: BCI system representation of functional elements and information flow of the P300 BCI. The red modules represent the modules that were developed in this dissertation.

BCI experiments are divided into two sessions: calibration and online operation. In the calibration session, the user selects a pre-defined set of symbols (target events), in order to gather labeled data (ground truth) to train the classifier. The obtained classification models are then applied in the online session. Main blocks of the BCI system are shown in Fig. 4.3.

4.1 Signal acquisition

The BCI system records the EEG brain signals using active Ag/AgCl electrodes electrodes placed on a cap (g.EEG cap). Active electrodes pre-amplify the EEG signal and are sent to the g.USBamp acquisition system. EEG signals were recorded monopolarly utilizing 16 electrodes, placed accordingly to the extended international 10-20 standard, as shown in Fig. 4.4. The right or left earlobe was selected for reference and the AFz channel for ground. Sixteen channels were

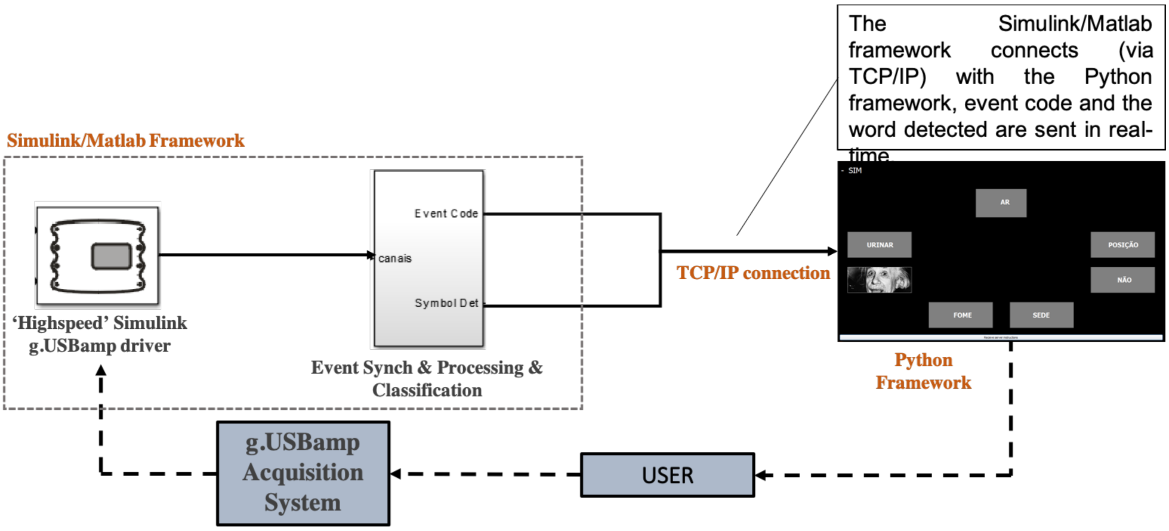

The Simulink/Matlab framework connects (via TCP/IP) with the Python framework, event code and the word detected are sent in real-time
Simulink/Matlab Framework
Event Code
canais
Symbol Det
'Highspeed' Simulink g.USBamp driver
Event Synch & Processing & Classification
TCP/IP connection
SIM
AR
URINAR
POSIÇÃO
NÃO
FOME
SEDE
Python Framework
g.USBamp Acquisition System
USER

Figure 4.3: Main blocks of the BCI system, and representation of User in the loop.

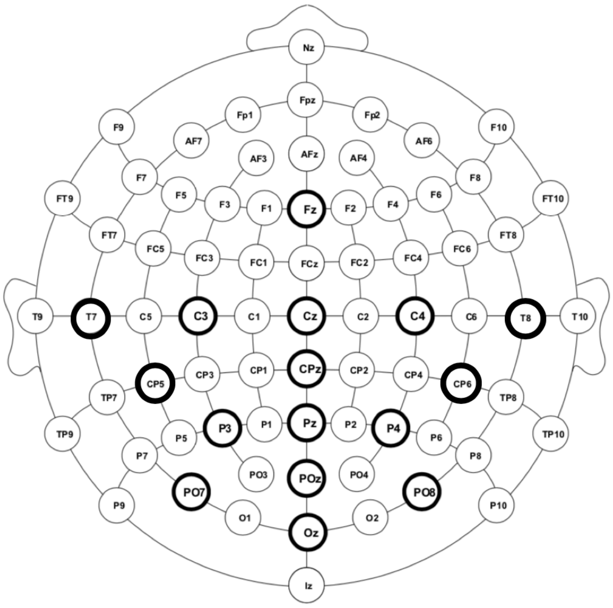


Figure 4.4: Location of electrodes, according to the international 10-20 extended system, of the 65 channel cap used in the experiments (g.EEGcap). Bold circles illustrate the channels used in the P300 experiments in this work.

selected: Fz, Cz, C3, C4, CPz, Pz, P3, P4, PO7, PO8, POz, Oz, T7, CP5, T8, and CP6. The first 12 channels were selected respecting the visual task [59] and the last four channels were selected considering the auditory cortex location [21]. EEG signals are sample at a 256Hz rate with a 16-channel g.USBamp acquisition system (g.tec medical engineering GmbH, Austria) [22]. The data were epoched (segmented) between the stimulus onset and 1s after. The 'Highspeed' Simulink g.USBamp driver provides a hard real-time clock through a hardware interrupt that drives the whole Simulink model, which ensures a real-time operation. Since the EEG signals are very sensitive to noise, the acquisition system performs directly a preprocessing step. EEG signals are filtered by a band-pass filter between 0.5 and 30 Hz to remove high frequency noise and a notch filter at 50 Hz, which rejects the power line noise source. The g.USBamp Simulink block allows some hardware configurations: sampling rate, monopolar and bipolar montages, pre-set filters, etc.

4.2 Event synchronization, processing, and classification

The 'Event & synchronization & processing & classification' block performs the event generation and synchronization, data buffering, epoch segmentation, and implements the algorithms for EEG signal processing and classification. This block is implemented in a Simulink S-function, which allows buffering and real-time operation [Mathworks 2012b]. As already described, the BCI system is based on the oddball paradigm. To create this paradigm, random events have to be created and synchronized with the recorded EEG data. The events generated are synchronized with the signal samples, whose sampling period is 1/256 seconds. Temporal events generated in this block are fully parameterized, stimulus time (T_{ON}) = $550ms$, inter-stimulus interval (ISI) = $100ms$, inter-trial interval (ITI) = $8s$, as shown in Fig. 4.5. Each event is associated with an epoch of 1 second long, as shown in Fig. 4.6. Each event has an associated code, which indicates the word/symbol that will be ON (highlighted) or OFF. At any given time there can be only one ON event. No significant changes have to be made between the offline and online implementations since the classification algorithms are directly embedded in this module after they have been trained offline in Matlab.

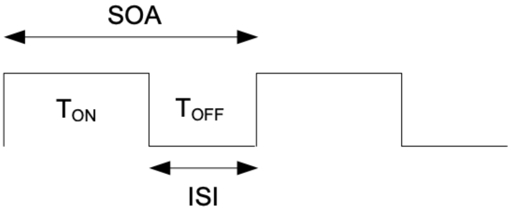


Figure 4.5: Temporal diagram of a set of stimulus events. SOA: stimulus-onset asynchrony, the time between the onset of one stimulus and the onset of the next stimulus. ISI: inter-stimulus interval, the time between the offset of one stimulus to the onset of another [60].

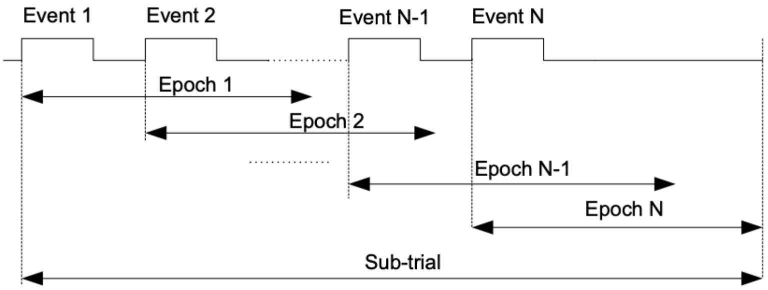


Figure 4.6: Temporal diagram relating epochs extracted from the continuous EEG data stream and the events [60].

4.2.1 Statistical spatial filter and Naïve Bayes

The BCI uses the same classification method as the one presented in [58]. It uses a statistical spatial filter based on a Fisher criterion beamformer (SF-FCB), represented in Fig. 4.7 by the

"Spatial Filter W" block. After spatial filtering, the feature space is an unidimensional vector that results from the concatenation of the two most discriminative projections of the spatial filter. This correspond to a vector with 512 features (amplitudes of the time samples). To further reduce the dimension of the feature vector, extracted features are selected according to their discriminative power using a feature selection method. From the 512 size vector, 200 features are selected using the R-square correlation method, $\mathbf{y} = [y(t_1)y(t_2)...y(t_T)]$. The r-square coefficient (square of the Pearson's correlation coefficient) obtains the discriminative relevance of the features between target and non-target epochs, returning a value ranging from 0 to 1. High r^2 values denote large discrimination.

For the calibration session, the selected features are then used to train a Bayesian classifier that is presented in its naïve form, which implies that the features are conditionally independent:

$$p(\mathbf{y}|C_i) = \prod_{j=1}^{N_f} p(\mathbf{y}(j) \mid C_i) = \prod_{j=1}^{N_f} \frac{1}{\sqrt{2\pi}\sigma_i(j)} exp\bigg(- \frac{(\,y(j) - \mu_i(j))^{\,2}}{2\sigma_i^2(j)}\bigg) \tag{4.1}$$

where each feature j is assumed to have a normal distribution $\mathcal{N}(\mu_i(j), \sigma_i^2(j))$. N_f represents the number of features, and the target and non-target classes are defined by $C_i(i \in \{+, -\})$. The posterior conditional probability $p(C_i|\mathbf{y})$ is calculated from the conditional probabilities using the Bayes theorem:

$$P(C_i|\mathbf{y}) = \frac{P(C_i)p(\mathbf{y}|Ci)}{p(\mathbf{y})} \tag{4.2}$$

the prior probability ($P(C_i)$) for target is $1/14$ and for non-target is $13/14$. The class is detected using the maximum a posteriori decision rule:

$$\hat{c} = argmax\{P(C_+|\mathbf{y}), P(C_-|\mathbf{y})\}. \tag{4.3}$$

From the offline session, a classification model is saved to be used in the online session, which includes: 200 indices of the selected features, the classification, and spatial filter models. These models are then used to classify online each stimulus event as target or non-target (binary classification). Then, a C-class classification is applied (where C is number of classes, i.e., the number of possible choices (symbols) in the paradigm), which combines the posterior probabilities of the binary classifiers associated to each event. The word/symbol associated with the highest probability is chosen as being the word mentally selected by the user. Fig. 4.7 shows the overall classification process.

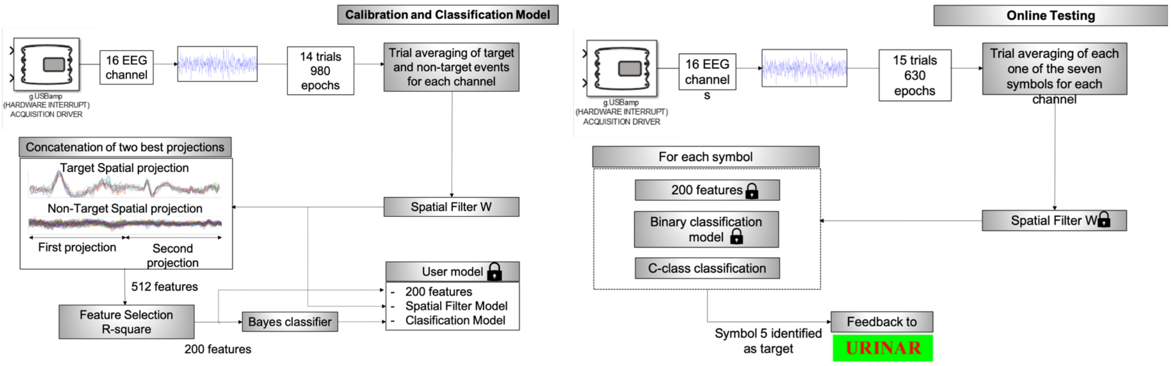


Figure 4.7: Classification architecture of the P300-based BCI system. Left: offline training to obtain feature extractor models, selected features, and classifier models. Right: for the online recognition, the binary classifiers are applied to each event, and then their output scores are combined to identify the mentally selected symbol. Figure adapted from [3].

4.2.2 Convolutional Neural Network

The Bayesian classifier was implemented in combination with a statistical spatial filter. This spatial filter significantly increases the signal-to-noise ratio of the EEG signals at the same time as it reduces the feature vector dimension. These properties make the subsequent choice of the classifier almost irrelevant as it was shown in [60], where several linear and non-linear classifiers were tested. Here, we want to compare the current approach with a CNN-based classifier providing as input the raw EEG channels, to analyze whether the CNN can learn effective spatial and temporal features. CNN's can learn highly complex patterns due to its non-linear architecture and have been considered for use with EEG since the network can handle large variability in the data.

The CNN studied and implemented offline in this dissertation was based on the CNN proposed in [11]. It is implemented in Matlab 2012b. Figure 4.8 shows the proposed CNN architecture applied. The CNN input is the matrix $(C \times N)$ where C expresses the number of electrodes used for EEG signal recording, and N represents the number of recorded temporal samples in one epoch (1 second segment associated to each event). In this work, $C = 16$ and $N = 256$. The temporal signal samples were previously band-pass filtered between 0.5Hz and 30Hz to remove high-frequency noise and notch filtered at 50Hz, to reject the power line frequency noise source, the same way as in the ”spatial filter + Naive Bayes” approach. The CNN classifier was trained based on subject-specific datasets.

The CNN is composed of five layers (L_0,L_1,...,L_4), each layer is composed of one or several maps (M). The map(s) in each layer are the result of the convolution of the map(s) of the previous layer with a kernel mask. A CNN kernel mask is simply a set of weights shared all over the input space (different sets of weights describe different convolution kernels). The first hidden layer, L_1, performs a spatial convolution operation with a kernel with size (1, 16). This convolution operation converts each receptive field of the signal samples into an abstract datum in a feature map. The signal samples in each receptive field are from all C channels in the space domain and sampled at one-time point in the time domain. The kernel size (1, 16) is used to make this layer to

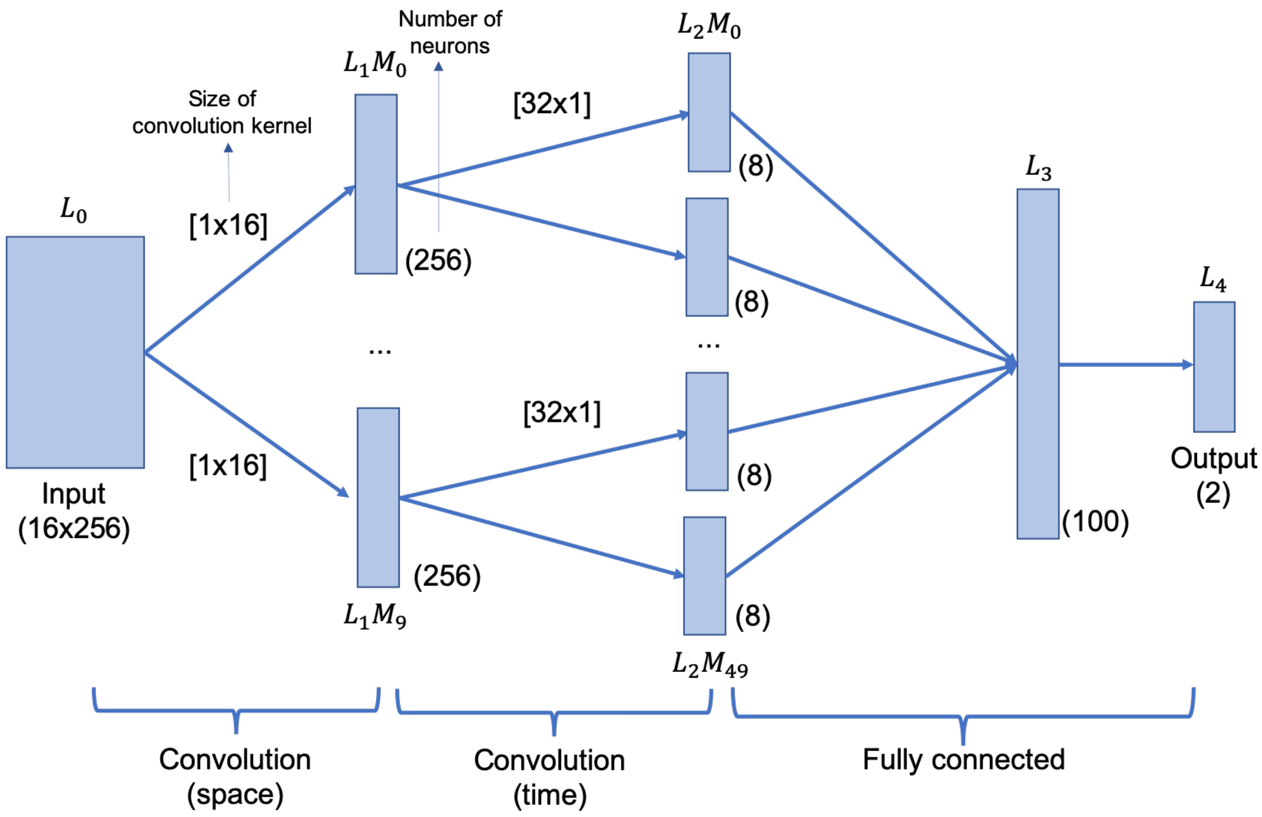


Figure 4.8: Convolution neural network architecture [11].

learn the spatial features from EEG signals acquired using all channels. L_1 generates ten feature maps, which are the input to the second hidden layer. The choice of 10 feature maps follows the suggestion in [11]. These feature maps are abstract temporal signals instead of raw signals since the spatial convolution layer converts each receptive field of raw signals into an abstract datum. The second hidden layer, L_2, performs the temporal convolution operation with the kernel size (32, 1). The temporal convolution operation converts each receptive field of the abstract signal samples into a feature map. The CNN architecture is based on what is traditionally done in BCI. Firstly an optimal spatial filter is performed, then the signal is processed in the time domain. The convolution kernels are all vectors and not matrices. The third hidden layer, L_3, is a fully-connected layer of size 100. L_4 is the output layer of the network with a size of two neurons, which represent the probabilities of the absence or the presence of a P300 wave. The layers are defined as:

- Input layer L_0: (16 x 256) matrix, $I_{i,j}$, $0 \leq i < C, 0 \leq j < N$.
- First hidden layer L_1: has ten maps ($L_1M_{0..9}$), each map has a size of 256. The convolution kernel size is [1 x 16].
- Second hidden layer L_2: is composed of fifty maps ($L_2M_{0..49}$), where each map has 8 neurons. The convolution kernel size is [32 x 1].
- Third hidden layer L_3: is composed of one map with 100 neurons. This map is fully con-

nected to the different maps of L_2.

- Output layer L_4: has one map with two neurons, one neuron represents the class ”P300” and the other neuron represents the class ”non-P300”. This layer is fully connected to L_3.

The current value of a neuron, $n(l, m, j)$, in the CNN is $x^l_m(j)$, where m denotes the map, l the layer, and j the position in the map:

$$x^l_m(j) = f(\sigma^l_m(j)). \tag{4.4}$$

The activation function, f, varies from layer to layer. For the first and second hidden layer the activation function used is a scaled sigmoid function, which represents convolution of the input signal:

$$f(\sigma) = 1.7159 \tanh(\frac{2}{3}\sigma) \tag{4.5}$$

the constants are set according to [46]. For the last two layers the activation function is the classical sigmoid function:

$$f(\sigma) = \frac{1}{1 + exp^{-\sigma}} \tag{4.6}$$

$\sigma^l_m(j)$ is defined for each layer and represents the scalar product between a set of input neurons and the weight connections between these neurons and the neuron $n(l, m, j)$. The first hidden layer is a space convolutional layer and the second hidden layer is a time convolutional layer. In L_1 and L_2 each neuron is related to a subset of neurons from the prior layer. For the first hidden layer:

$$\sigma^1_m(j) = w(1, m, 0) + \sum_{i=0}^{i<16} I_{i,j} w(1, m, i) \tag{4.7}$$

a set of weights $w(1, m, i)$ with m fixed, and $0 \leq i < C$, corresponds to a spatial filter. The goal of this layer is to find the best channel combination for the classification. The convolution represents spatial filters, and the kernel size is [1x16].

For the second hidden layer:

$$\sigma^2_m(j) = w(2, m, 0) + \sum_{i=0}^{i<32} x^1_m(j * 32 + i) w(2, m, i) \tag{4.8}$$

this layer transforms the signal of 256 values into 8 new values in L2. The convolution represents temporal filters, and the kernel size is [32x1].

For the third hidden layer:

$$\sigma^3(j) = w(3,0,j) + \sum_{i=0}^{i<50} \sum_{k=0}^{k<8} x_i^2(k) w(4,i,k) \tag{4.9}$$

the second and the third hidden layers are fully connected. For the output layer:

$$\sigma^4(j) = w(4,0,j) + \sum_{i=0}^{i<100} x^3(i) w(5,i) \tag{4.10}$$

the third hidden layer and the output layer are fully connected.

The threshold ($w(1,0,j)$, $w(2,0,j)$, $w(3,0,j)$, and $w(4,0,j)$) and the input weights, for each neuron, are initialized with a standard distribution around $\pm 1/n(l,m,i)_{N_{input}}$. The $n(l,m,i)_{N_{input}}$ is the number of inputs of $n(l,m,i)$. The learning rate, γ, for L_1 and L_2 is:

$$\gamma = \frac{2\lambda}{n(l,m,0)_{N_{shared}} \sqrt{n(l,m,i)_{N_{input}}}} \tag{4.11}$$

$n(l,m,0)_{N_{shared}}$ represents the number of neurons that share the same set of weights and λ is a constant, which was set to 0.2. The learning rate, γ, for L_3 and L_4 is:

$$\gamma = \frac{\lambda}{\sqrt{n(l,m,i)_{N_{input}}}}. \tag{4.12}$$

The training of the CNN is carried out by a classical backpropagation learning algorithm to tune the filter values/weights of the CNN, where the weights are corrected due to a gradient descendent to minimize the output error. The training stops once the least mean square error is minimized. The output layer is composed of two neurons, $x^4(0)$ and $x^4(1)$, where $x^4(1)$ represents the presence of a P300 wave and the $x^4(0)$ the absence of a P300 wave. During the test, a P300 wave is detected if $x^4(1) > x^4(0)$, otherwise there is no P300 wave.

4.3 Stimulation and presentation

A P300 BCI system requires a strict synchronization between recorded data and stimuli events, where the delay between event generation and stimuli presentation has to be very small and also constant. The stimulation module provides information about the presented stimuli using trigger signals and events, which are used to synchronously process the input signals and extract corresponding features. At the end of each trial, the classifier determines the intention of the subject and sends the control output (symbol detected) to the 'Hybrid paradigm' framework where the presentation module shows the feedback to the user, which enhances the user's response to the presented stimuli and helps assess the system’s efficacy.

4.3.1 S-Functions

The goal of the ’Hybrid paradigm’ block in Fig. 4.3 is to be an interface that is capable of receiving the codes from the Matlab framework, and present visual images and auditory stimuli simultaneously, in real-time. The visual paradigms previously designed in the laboratory were implemented with S-Functions in Matlab script using simple graphic structures. S-Functions provide a powerful mechanism for extending the capabilities of the Simulink environment. An S-Function is a computer language description of a Simulink block written in Matlab, C, C ++, or Fortran, and that can be compiled as MEX files. S-Functions are dynamically linked sub-routines that the Matlab interpreter can automatically load and execute. After studying the implementation of stimuli based on images in the paradigm through S-Functions, it was concluded that their implementation would be difficult due to poor documentation, lack of flexibility and no warranty of image rendering in real-time. On the other hand, first tests with auditory stimuli showed the impossibility of loading sounds in real-time, without affecting the 256 Hz sampling rate acquisition.

In order to include in the paradigms an improved graphic component, with the possibility of including audio, 2D stylized graphics, images, and interaction, it was decided to develop the stimulation module outside Matlab/Simulink as a standalone application module. On the previous version, the visual stimuli (based on text) were implemented with an S-function and the auditory stimuli were produced outside Matlab using the commercial software ”Presentation”, which required an expensive license and and a complex set-up. Since all the rest of the system is still implemented in Matlab/Simulink it was necessary to create a communication channel between Matlab/Simulink and the external application.

4.3.2 Python stimulation framework

To increase flexibility in the development of paradigms, the feasibility of various development environments outside of Matlab/Simulink was explored. Some platforms were considered as mention in Chapter 3.2, however, they were excluded since the interfaces were considered not to be practical nor intuitive, they depended on many package dependencies, and proved to be not flexible enough.

Although some platforms implemented in Python were discarded, Python language was still the most appealing approach as it is a high level, interpreted and versatile dynamic programming language that emphasizes code readability. So it was decided to move on with and implementation from scratch based on Python. The Python framework is completely written in Python 3.7 and consequently not attached to a special operating system. This framework allows for implementing feedback and stimuli without having to worry about the underlying BCI system. Hence, it is not only a framework for the rapid development of feedback and stimulus applications but also a platform to run neuroscientific experiments independent from BCI systems. Some graphical user interface (GUI) toolkits that Python supports were searched. At an early stage the visual part of the paradigm was developed in Tk/Tcl, which is part of Python, and the auditory stimuli were developed with the Winsound module. Firstly, the framework was tested without real-time restrictions and seemed to provide a good performance. However, tests in real-time exhibited high processing delays. Although Tk/Tcl and Winsound are popular for their simplicity and effectiveness, they are slow to update, generating delays in real-time applications. To solve this problem the framework was re-written in PyQt and PyGame (a cross-platform set of Python modules designed for writing video games).

The data communication interface between Matlab/Simulink and the external application is based on a TCP/IP connection, providing a reliable, connection orientated, with no packet loss. The data shared between Matlab/Simulink and the external application are the event codes. However, the interface can receive any data, for example it can be used to share EEG data. To send the event codes from the Matlab to the Python framework, a TCP/IP server object was issued on the 'Event synchronization & processing & classification' block (Fig. 4.3), and thereby Matlab was acting as server which provided the flexibility to have different clients connected to it. An excerpt of the Matlab code is given as an example:

```
    ...

disp('Esperando cliente...');
LIGACAO = tcpip('localhost', 2000, 'NetworkRole', 'server');
fopen(LIGACAO);
disp('Conectado!')

    ...

if(block.Dwork(25).Data(IndexEv) != 0)
    %Mensagem a enviar
    caracteres = [num2str(block.Dwork(25).Data(IndexEv)) 10];
    fwrite(LIGACAO, caracteres, 'char');
end

    ...

det = sf_output_class(class,val)
if(det != 0)
    d = det;
    caracteres = [num2str(d) 10];
    fwrite(LIGACAO, caracteres, 'char');
end

    ...
```

The variable block block.Dwork(25).Data(IndexEv) stores the random sequence of event generation, i.e., event code number, and sf_output_class(class,val) is the code for the symbol/event detected by the classifier. Online and calibration modes use the same process, except that in calibration mode there is no detection.

The external Python framework was implemented using modular programming, i.e., breaking the larger code into separate, smaller, more manageable subtasks or modules. Also, it uses asynchronous programming, which is a form of parallel programming that allows a unit of work to run separately from the primary application thread. A TCP/IP client socket was implemented in a module to connect with the Matlab server and receive the codes in real-time. By default, the client connects to the hostname 127.0.0.1 (localhost) and the port 2000. Another module was implemented to load the .wav files and to issue the auditory stimuli. Each .wav file has a natural meaningful spoken word. To load and play the sounds the PyGame library was used.

To develop the the visual stimuli, PyQt libraries were considered. PyQt is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is free software developed by the British firm Riverbank Computing. PyQt implements around 440 classes and over 6,000 functions and methods including a substantial set of GUI widgets. GUI programming with Qt is built around the concept of signals and slots for communication between objects. A signal is emitted when an event occurs (e.g. a button is clicked), and slots are callable functions that handle the event (e.g. show a pop-up when a button is clicked). This allows more flexibility when handling GUI events which results in a cleaner codebase. A disadvantage is that it can take a while to get familiar with PyQt. It is a huge framework and there are many ways to implement different things, some of them are conflicting and might be confusing.

PyQt5 has several Python modules, from which were here used the *QtCore*, *QtGui*, and *QtWidgets*. The *QtCore* module includes non-GUI functionality. All other PyQt modules depend on this module. The *QtGui* module comprises the majority of the GUI classes. These include several table, tree, and list classes based on the model–view–controller design pattern. It also presents an advanced 2D canvas widget able to save thousands of items. The *QtWidgets* module implements a set of UI elements to create classic desktop-style user interfaces. Widgets are the basic elements for designing user interfaces in Qt. Widgets can display data and status information, receive user input, and render a container for other widgets that should be grouped.

To create the GUI a module, which implements all the specifications of the visual interface, was developed. The GUI is shown to the user and has a button, *Receive server instructions*, to connect to the Matlab/Simulink server and start the hybrid paradigm test. Once the connection is successfully established, the method *async def get_one_code()* is called, and several threads are initialized. An excerpt of this method, for the online test, is presented:

```
    async def get_one_code(self, tcp_client):
        serv_code = tcp_client.read_only_one_code_from_server()
        t_sound = PlaySoundThread(thread_id=serv_code,
                                  name=serv_code,
                                    counter=int(serv_code),
                                    index_vec=int(serv_code))
        t_sound.start()
        if tcp_client.count in [c   1, c * 2, c * 3 + 1, c * 4 + 2, c * 5 ...
            + 3, c * 6 + 4, c * 7 + 5, c * 8 + 6, c * 9 + 7, c * 10 + 8, ...
            c * 11 + 9, c * 12 + 10, c * 13 + 11, c * 14 + 12, c * 15 + 13]:
            self.update_code_in_label(serv_code)
            await asyncio.sleep(1)

        else:
            self.decode_code(serv_code, True)
            await asyncio.sleep(2)
            t_sound.join()
        return serv_code
```

This asynchronous method waits for the Matlab/Simulink server to send the event codes. The codes are received by *tcp_client.read_only_one_code_from_server()*. The event codes are sent as an argument to the *PlaySoundThread* class and the corresponding sound to is played. The received codes are counted so the GUI can provide feedback to the user after each trial (*c* is the number of repetitions times the number of events used in the paradigm (one trial), which correspond to 10 x 7 for the calibration test and for the online test the number of repetitions varies from participant to participant and has to be updated at the beginning of each online test), online tests have 15 trials. Otherwise, the GUI presents the visual stimuli regularly, where the interface flashes the corresponding word of the event code received. When all the event codes are received, all the threads terminate its execution and the program closes.

4.3.3 Paradigms description

Five P300-based gaze-independent conditions were tested, one with visual stimuli (flashing words), one with visual image-based stimuli (flashing words with overlapped pictures of well-known faces), one with visual image-based stimuli (flashing words with overlapped pictures of relative's faces), one with auditory and visual stimuli simultaneously (flashing words + spoken words), and one with auditory and visual image-based stimuli (flashing words with overlapped pictures of well-known faces + spoken words). A set of seven Portuguese words comprising a small communication lexicon was used in the five oddball tasks. The words chosen were SIM (Yes), NÃO (No), FOME (Hunger), SEDE (Thirst), AR (Air), POSIÇÃO (Position), and URINAR (Urinate) according to [3]. For the auditory stimuli the same Portuguese words were used and were recorded and rectified using the Audacity (Audacity Team, version 2.0.5) software, provided from [3]. Fig. 4.9 shows the layout of the visual paradigm during a stimulus flashing associated with word 'SIM'.

The seven words were chosen regarding some of the basic needs of CLIS patients, with the possibility of a "YES/NO" conversation. The word AR is regarding the breathlessness sensation, POSIÇÃO concerns the need for position changing due to pain or discomfort, the rest of the words

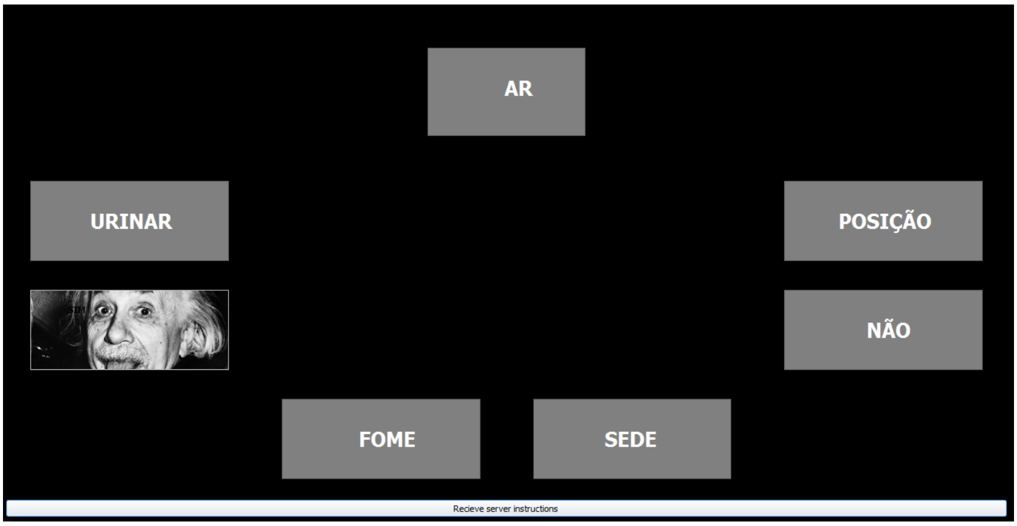


Figure 4.9: Layout of the seven visual stimuli on the flashing words with overlapped pictures of well-known faces task. The highlighted event is ”SIM”, the background is changed from white/gray to an image of a famous face.

are intuitive. The suggested bimodal interface is not complex and includes a reduced set of vocabulary due to the intention of increasing the involvement of patients with the task, improve their selective attention, and reduce their working memory demands. Each of the five tasks consisted of a 7-stimuli oddball paradigm and were divided into a calibration session and an online session:

- The calibration session was composed of 14 trials, each trial corresponds to a specific word that was pre-established before the test. The 14 trials that constitute the calibration session are composed with the following word targets: SIM, FOME, URINAR, POSIÇÃO, NÃO, SEDE, AR, SIM, FOME, URINAR, POSIÇÃO, NÃO, SEDE, and AR. Each trial consisted of 10 sequences (7 random stimuli per sequence). In each sequence, there was one target stimulus and nine non-target (standard) stimuli. For each trial, the participant has to focus on the target stimuli and to mentally count them ignoring all other standard events. The time between each selection (inter-trial interval—ITI) was 8 s. The 8 s value was selected to allow the participants to rest and perform ocular movements between trials. This session records the data labeled with class information, associated with true labels (ground truth). At the end of each calibration session the information is saved (140 target epochs and 840 standard epochs) and used to train the classification models and then applied in the online sessions.

- For the online session, the sets per trial were optimized for each participant using the calibration data. Once chosen the best number of repetitions for each participant (chosen on the first task), it remained the same for the rest of the tasks, to compare the performance between them. The system returned the selected symbol after one trial (see Fig. 4.10). Participants had to complete a set of 15 trials (selection of 15 words): SIM, FOME, URINAR, POSIÇÃO, NÃO, SEDE, AR, SIM, FOME, URINAR, POSIÇÃO, NÃO, SEDE, AR, and SIM. The set of words was the same for all participants and for the 5 conditions. Before each trial, participants were informed of the target word to focus. Feedback (selected word) was visually provided at the end of each trial for the five conditions.

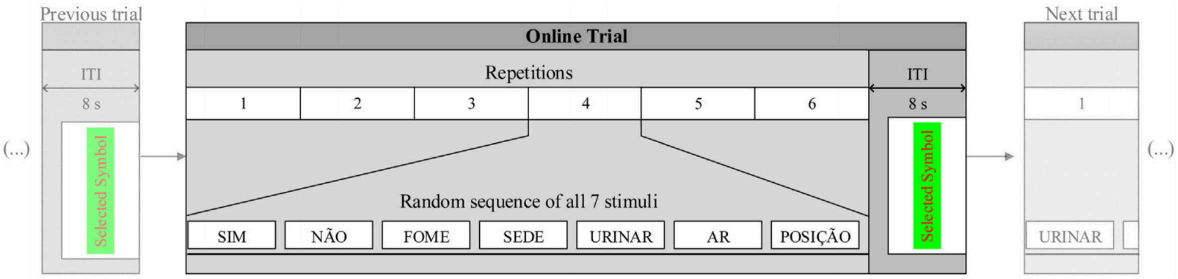


Figure 4.10: Illustration of one online trial and inter-trial interval (in which it is returned the selected symbol), Fig. reproduced from [3]. The number of repetitions, for this participant, is six.

The temporal diagram of the stimuli of one sequence for the hybrid modality is presented in Fig. 4.11. All stimuli have a duration of 550 ms and an interstimulus interval of 100 ms (stimuli onset asynchrony of 650 ms). The T_{ON} was set has 550 ms based on the minimum requirements for the perception of the auditory stimuli. The T_{ISI} of 100 ms was set as the minimum interval required to distinguish two consecutive words. The temporal parameters were maintained for the five conditions, even though for the tasks with no auditory stimuli the temporal parameters could considerably be shortened. The time between trials, on the online session, was 27.3 s (6 × 7 × 0.650s) plus the ITI of 8 s, for a participant with the number of sequences equal to 6. For a participant with a number of sequences equal to 3, the time for one-word selection was 13.7 s (3 × 7 × 0.650s). Participants performed four conditions in a single experiment, which took about 2-3 h to complete, depending on the number of sequences chosen for the online sessions. The period of the experiment also included the EEG montage and a period for familiarization with the stimuli and interface arrangement. The subjects performed the flashing words with overlapped pictures of relative's faces task on another day, which took about 30 minutes.

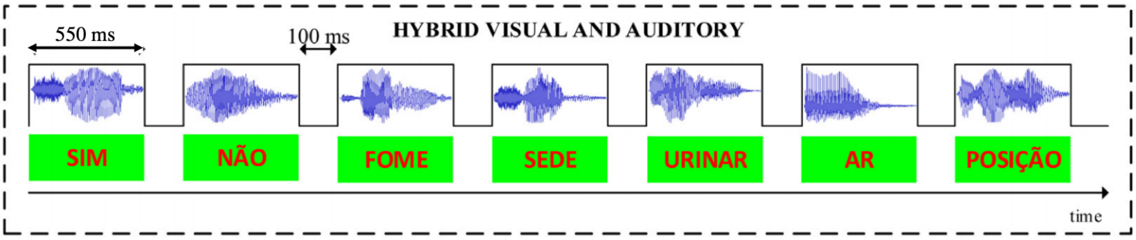


Figure 4.11: Temporal diagram of stimuli presentation for the hybrid modality with word flashing + spoken words (Fig. reproduced from [3]). All stimuli have a duration of 550 ms and an interstimulus interval of 100 ms (stimuli onset asynchrony of 650 ms).

5

Analysis and results

This chapter presents all the tests, results, and analyses that have been done to verify and validate the developed work. The first section concerns the stimulation and presentation modules, reporting the experiments with the five P300 paradigms conditions. The second part of this chapter concerns the classification module, comparing the performance of the two classification approaches.

5.1 P300 Paradigms

Five stimulation paradigms were experimentally tested by 10 healthy participants: word flashing (W), word flashing with auditory (WA), famous face flashing (F), famous face flashing with auditory (FA), and flashing a relative's faces (FF). Each test consisted of a calibration session and a subsequent online session. The order of the paradigms for each participant was randomized. The conditions W, WA, F, and FA were all conducted in the same day. Participants had a brief interval between each paradigm that allowed subjects to rest between the different tests. Condition FF was conducted after a week of conducting the other four conditions.

5.1.1 Participants

The five tests were conducted with ten healthy participants, except participant five (P5) that was not available at the time of the FF experiment to perform the FF task. Eight of the participants were male, and two were female. The mean age of the participants is 28.5 years, with a range from 21 to 50. None of the participants has a hearing impairment, and all had a normal or corrected vision. Only two participants (P1 and P10) had previous experience with BCIs, the other participants had never controlled a BCI before (naïve users). Table 5.1 shows their identification codes, ages, gender, BCI experience, first condition conducted, and number of repetitions.

5.1.2 Offline classification results

The offline classification results were obtained for the five condition tests. The BCI was operated using several repetitions per trial (N_{rep}), adjusted individually to each participant. Since the signal

Participant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Age	22	23	50	25	23	23	23	50	21	25
Gender	F	M	M	M	M	M	M	M	M	F
BCI experience	yes	naïve	naïve	naïve	naïve	naïve	naïve	naïve	naïve	yes
First condition conducted	WA	WA	W	FA	F	F	FA	WA	F	W
N_{rep}	3	6	5	4	5	5	6	3	3	3

Table 5.1: Gender, age, BCI experience, first condition conducted, and number of repetitions for online operation. (F—Female, M—Male, N_{rep}—Number of repetitions).

has a very low SNR, it is usually not possible to detect the P300 with only one sequence. Figure 5.1 depicts the offline classification accuracy (y-axis) for each subject as well as averaged accuracies for all subjects. The number of sequences considered was from one to six (x-axis) for all five different paradigms.

The analysis of the average of the 10 subjects indicated that accuracy increases with the sequence number in all five paradigms. The average classification accuracy of the P300-paradigm was greater in the F paradigm, followed by the WA paradigm, then the FA paradigm and the FF and W paradigm have a close accuracy. However, it must be taken into consideration that participant 9 (P9) had a 50% accuracy for every number of sequence in the FF condition, which indicates that some type of error occurred during the calibration session. By eliminating the P9 of the mean accuracy the average classification accuracy of FF condition is approximately the same as the FA condition. Besides, we counted the number of sequences needed for subjects to achieve a ≥70% accuracy level in the five paradigms. A level of ≥70 is regarded as a minimum level of communication [44][42].

A paired t-test was calculated between the W and F conditions, the WA and FA conditions, the F and FF conditions, and between the W and FF conditions (see Table 5.2). The results of the paired t-test indicate that the number of sequences was not significantly different between any of the paradigms pair (to determine statistical significance p has to be $p<0.05$). The participants required 1.9 ± 1.52 (mean ± standard deviation) sequences to achieve the goal of ≥70% classification accuracy in the W paradigm, whereas 1 ± 0 sequences were needed to achieve the same goal in the F paradigm. For the WA paradigm, the subjects required 1.3 ± 0.67 to achieve the goal of ≥70% classification accuracy, whereas 1.8 ± 1.68 sequences were needed to achieve the same goal in the FA paradigm. Offline selection accuracies for selecting one symbol out of 7 by using single sequence data were 86.98% ± 16.41 for FF (without P9), 85.29% ± 7.36 for F, 81.12% ± 11.51 for WA, and 80.93% ± 17.81 for FA, and 75.59% ± 16.04 for W.

As said previously, the order of the five conditions was randomized for each participant. In the first condition tested, the number of sequences was decided and maintained for the rest of the four online sessions (see Table 5.1).

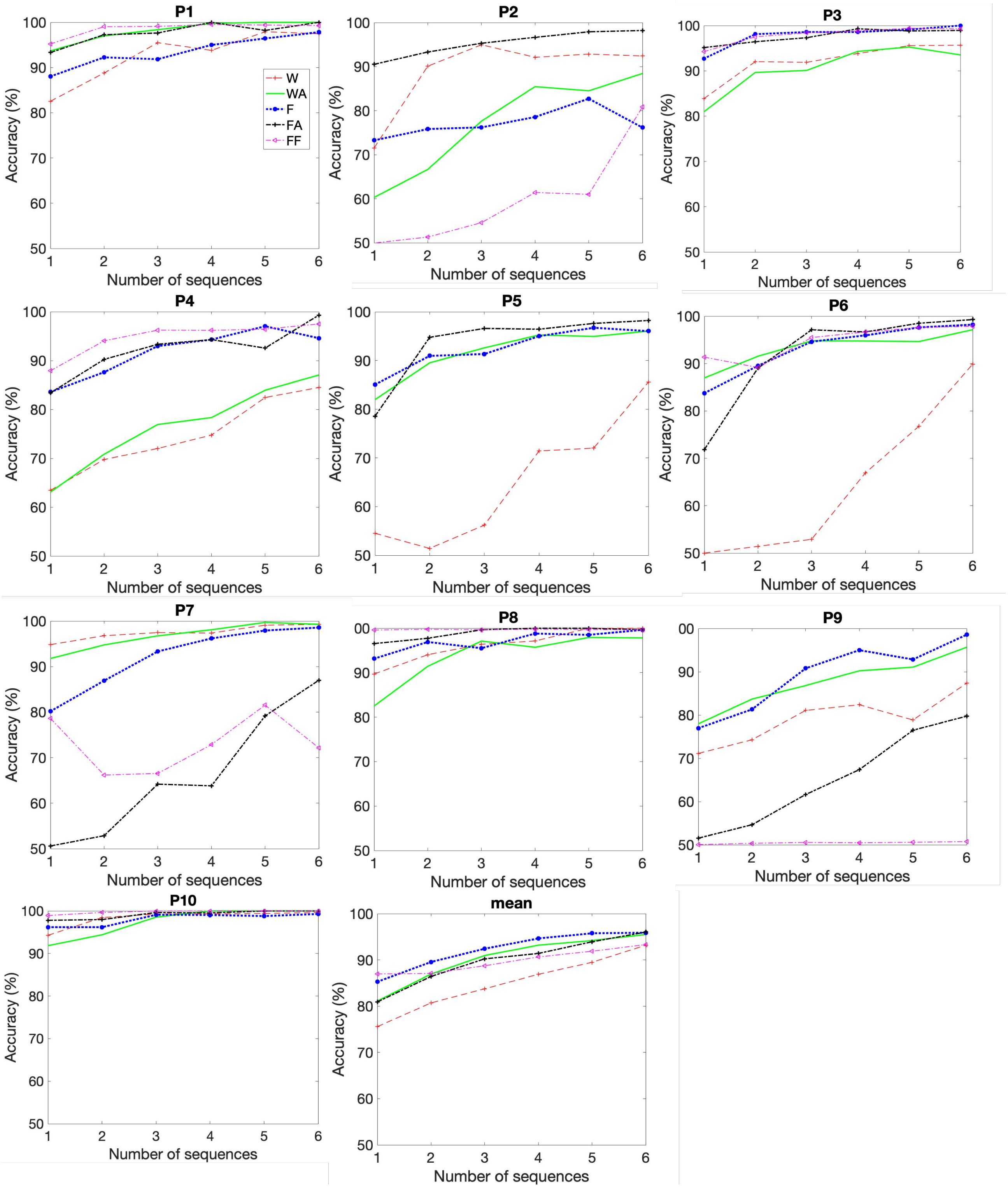


Figure 5.1: Individual and average offline classification accuracies of the P300-paradigms for 10 participants in the five conditions.

5.1.3 Online performance metrics

To compare the performance of the five tasks and these with the state of art results, some online metrics were calculated: online classification accuracy, the information transfer rate (ITR), and the

	W & F	W & FF	F & FF	WA & FA
p	0.0782	0.6865	0.1520	0.3955

Table 5.2: Results of the paired t-test between the five paradigms for the offline sessions taking into account the number of sequences needed for subjects to achieve a ≥70% accuracy (to determine statistical significance p has to be p<0.05).

		W			**WA**			**F**			**FA**			**FF**		
		Pac	$eSPM$	ITR	Pac	$eSPM$	ITR	Pac	$eSPM$	ITR	Pac	$eSPM$	ITR	Pac	$eSPM$	ITR
Participants	P1	93.33	3.82	9.34	93.33	3.82	9.34	100	4.10	11.50	100	4.10	11.50	100	4.10	11.50
	P2	93.33	1.98	4.84	80	1.70	3.33	80	1.70	3.33	100	2.12	5.95	86.67	1.84	4.02
	P3	86.67	2.19	4.79	80	2.02	3.96	100	2.53	7.10	100	2.53	7.10	100	2.53	7.10
	P4	46.67	1.46	1.35	53.33	1.67	1.89	86.67	2.71	5.93	93.33	2.92	7.13	100	3.13	8.77
	P5	73.33	1.85	3.24	73.33	1.85	3.24	86.67	2.19	4.79	86.67	2.19	4.79	-	-	-
	P6	93.33	2.36	5.76	66.67	1.68	2.60	100	2.53	7.10	93.33	2.36	5.76	100	2.53	7.10
	P7	100	2.12	5.95	100	2.12	5.95	100	2.12	5.95	86.67	1.84	4.02	93.33	1.98	4.84
	P8	86.67	3.55	7.77	93.33	3.82	9.34	100	4.10	11.50	100	4.10	11.50	100	4.10	11.50
	P9	60	2.46	3.29	80	3.28	6.42	80	3.28	6.42	73.33	3	5.25	80	3.28	6.42
	P10	86.67	3.55	7.77	100	4.10	11.50	100	4.10	11.50	100	4.10	11.50	100	4.10	11.50
mean		82	2.53	5.41	82	2.61	5.76	93.33	2.94	7.51	93.33	2.93	7.45	**95.56**	**3.07**	**8.08**
STD		16.07	0.77	2.31	14.31	0.97	3.15	8.43	0.86	2.81	8.43	0.84	2.80	7.03	0.85	2.73

Table 5.3: Online session accuracy Pac, ITR, and $eSPM$ for the five conditions: W, WA, F, FA, and FF (STD—Standard Deviations).

effective symbols per minute (eSPM). Word accuracy of online testing is the most important metric since it defines the feasibility of the interface, and it is calculated as the percentage of correctly selected words for each online test. The interface speed depends on the number of repetitions chosen for each participant and was established as a trade-off between accuracy and speed. For N_{rep}=6 the interface speed was fixed to 2.2 SPM, N_{rep}=5 the interface speed is 2.6 SPM, N_{rep}=4 the interface speed is 3.3 SPM, N_{rep}=3 the interface speed is 4.1 SPM (discarding the ITI time). The effective SPM (eSPM) is also calculated. The bit rate or information transfer rate (ITR) embeds concurrently the number of encoded symbols, the online accuracy, and the SPM:

$$SPM = \frac{60}{N_{rep} \times (N_{ev} \times SOA) + 1} \tag{5.1}$$

$$eSPM = SPM \times P_{ac} \tag{5.2}$$

$$ITR = SPM\Big[\log_2(N_s) + P_{ac}\log_2(P_{ac}) + (1 - P_{ac})\log_2\frac{(1 - P_{ac})}{(N_s - 1)}\Big] \tag{5.3}$$

where N_{rep} is the number of repetitions selected according to Table 5.1, N_{ev} is the number of events on the paradigms (N_{ev} = 7), SOA is the stimulus onset asynchrony ($T_{ON} + T_{OFF}$ = 0.650 s), the value 1 represents the time needed to record the epoch of the last event of the trial, P_{ac} is the online accuracy, and N_s is the number of symbols/words (N_s = N_{ev}).

For the online session, each subject had to select a sequence of 15 words for each of the five conditions. Table 5.1 shows the online accuracy, P_{ac}, the bit rate, ITR, and the $eSPM$ for the five conditions.

The best performance, with an accuracy of 95.6%, a $eSPM$ of 3.07 symbols per minute, and

	W& F			W & FF			F & FF			WA &FA		
	Pac	eSPM	ITR	Pac	eSPM	ITR	Pac	eSPM	ITR	Pac	eSPM	ITR
p-value	0.0773	0.3093	0.1003	0.0410	0.1948	0.0432	0.5653	0.7584	0.6770	0.0556	0.4622	0.2439
test decision	-	-	-	*	-	*	-	-	-	-	-	-

Table 5.4: Results of the paired t-test between the five paradigms for the online sessions (for statistical significance p<0.05, test decision—*).

an ITR of 8.08 bpm, was yielded by the FF paradigm, followed by the F and FA conditions, which had similar performances with an accuracy of 93.33%, a $eSPM$ of 2.93 for FA and 2.94 for F condition symbols per minute, and an ITR of 7.45 bpm for FA condition and 7.51 bpm for F condition. The W and WA paradigms had a lower performance with an accuracy of 82%. The results of the paired t-tests showed that the accuracy ($p < 0.05$) and ITR ($p < 0.05$) were significantly different between the FF and W paradigms. The mean classification accuracy in the FF paradigm was 13.56% higher than that of the W and WA paradigm, while the mean ITR in the FF paradigm was 2.67 bpm higher than that of the W paradigm. For the W and F conditions, we observe that the mean accuracy is 11.33% higher for the F case, however, the paired t-test shows that this difference is not statistically significant (paired t-test, p = 0.0773). A paired t-test, regarding the accuracy, was performed between the conditions WA and FA but no significant difference between them was found (paired t-test, p = 0.0556). Table 5.4 shows all the p-paired tests performed between the conditions. Results show that the approach based on famous and relative's faces is effective in comparison to text stimuli, and that the combination with auditory stimuli did not introduce any improvements.

5.1.4 ERP analysis

The grand averages of target and non-target EEG signals are plotted for the five conditions in Fig. 5.2 A). According to [37], the fusiform face area is a part of the human visual system used in facial recognition. Hence, channel PO7 was selected to represent the fusiform face area, and channels Cz and Pz were also selected as they are channels of reference for the analysis of P300 ERPs [41] [48] [77].

The amplitude and latency of the P300 and N400 waves were analyzed. The amplitude measures the value of the wave peak and the latency is the instant of the wave peak. These two parameters were analyzed from the epochs evoked by the target events gathered during the calibration sessions of the 10 participants taking the 16 channels, making a total of 140x16=2240 epochs per condition. Each of the five stimuli produces distinct waveforms. The specified time windows to examine the P300 amplitudes and latencies were set to 300–450 ms. The specified time windows to examine the N400 amplitudes and latencies were set to 450–750 ms for. The time windows were determined by examining the grand mean waveforms shown in Fig. 5.2.

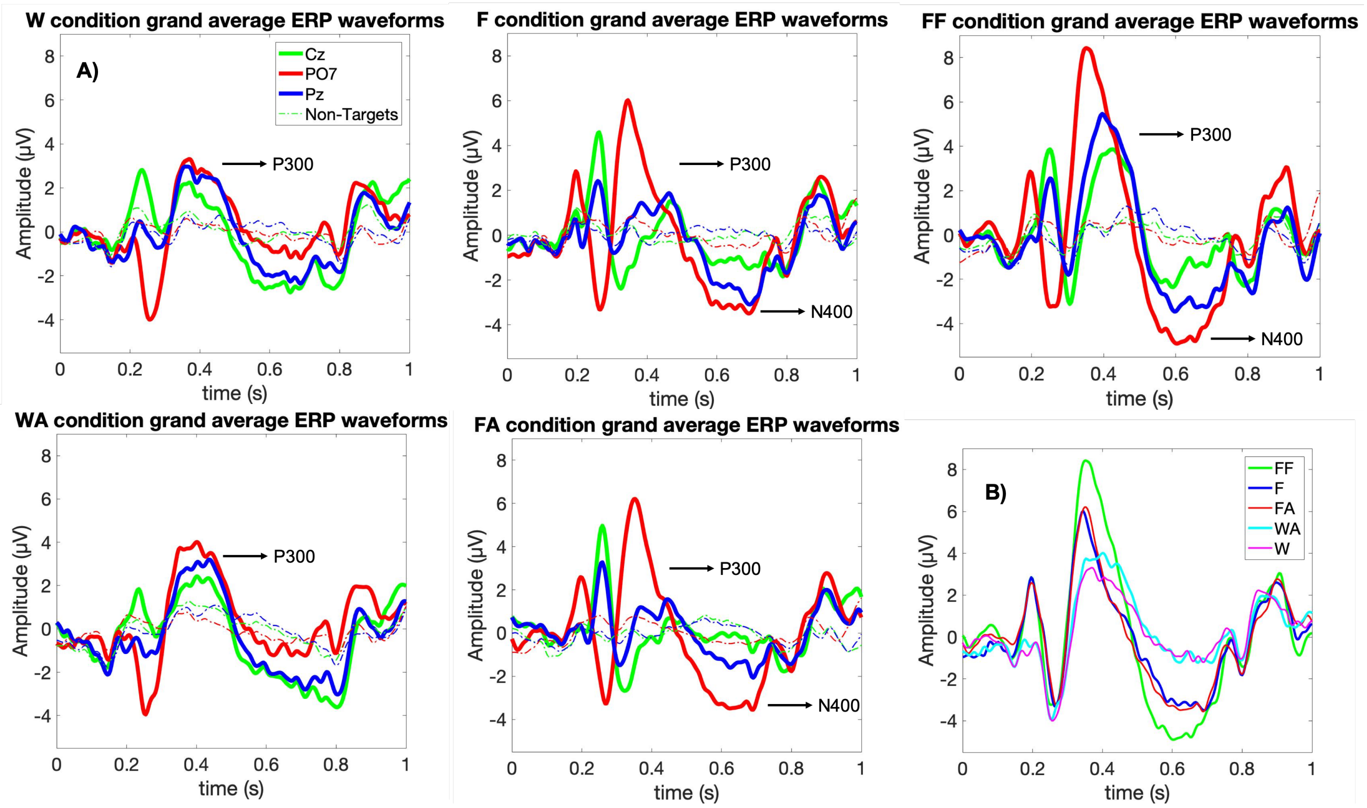
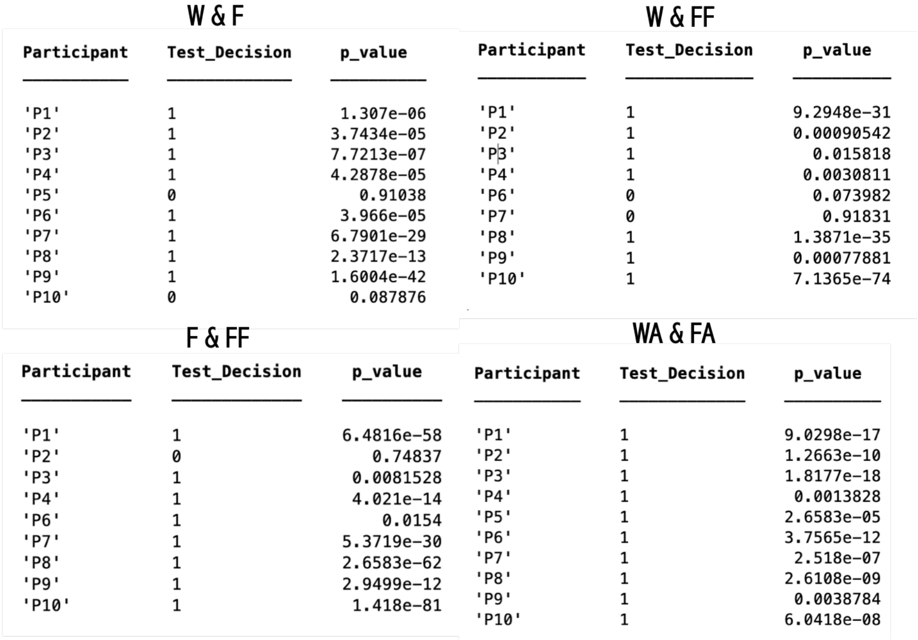


Figure 5.2: A) Grand average of target and non-target EEG signals for the channels Cz, PO7, and Pz for the five conditions (obtained from the 10 participants, except P5 for FF condition). B) Target grand average of EEG signals for PO7 channel for the five conditions.

5.1.4.1 P300 ERP analysis

A paired t-test was calculated for the maximum peak found on the time window 300–450 ms for the 2240 epochs for the five conditions for each participant. The paired t-tests results, in Fig. 5.3, show that for the majority of the participants the face paradigms (F, FF, and FA) amplitudes are significantly higher than the non-face paradigms (W and WA) amplitudes. Between the F and the FF amplitudes, there is also a significantly statistical difference. The average of the maximum peak found on the time window 300–450 ms for the 2240 epochs of each participant was calculated. The mean amplitude of the P300 wave for the condition FF is the highest (18.69 μV), followed by the conditions F (18.01 μV), W (17 μV), WA (16.16 μV), and FA (15.91 μV). The P300 wave has an average latency of 368.8 ms, 372.9 ms, 381.7 ms, 383.3 ms, and 384.6 ms for the F, FF, W, WA, and FA conditions, respectively. The latency for FF and F conditions are statistically lower than the latency for the condition W (paired t-test, p = 0.03, p = 0.01) and when comparing the FF and F conditions no significant differences were found (paired t-test, p = 0.32). Moreover, the FA and WA conditions do not present significant differences (paired t-test, p = 0.75). The higher latency for the conditions FA and WA may be due to the increased complexity in stimuli perception, as more time is needed for the subject to perceive the auditory stimulus [32].

The ERP waveforms obtained for F, FF and FA conditions are very different from the W and WA



W & F

```
Participant    Test_Decision    p_value
___________    _____________    __________

'P1'           1                  1.307e-06
'P2'           1                 3.7434e-05
'P3'           1                 7.7213e-07
'P4'           1                 4.2878e-05
'P5'           0                    0.91038
'P6'           1                  3.966e-05
'P7'           1                 6.7901e-29
'P8'           1                 2.3717e-13
'P9'           1                 1.6004e-42
'P10'          0                   0.087876
```

W & FF

```
Participant    Test_Decision    p_value
___________    _____________    __________

'P1'           1                 9.2948e-31
'P2'           1                 0.00090542
'P3'           1                   0.015818
'P4'           1                  0.0030811
'P6'           0                   0.073982
'P7'           0                    0.91831
'P8'           1                 1.3871e-35
'P9'           1                 0.00077881
'P10'          1                 7.1365e-74
```

F & FF

```
Participant    Test_Decision    p_value
___________    _____________    __________

'P1'           1                 6.4816e-58
'P2'           0                    0.74837
'P3'           1                  0.0081528
'P4'           1                  4.021e-14
'P6'           1                     0.0154
'P7'           1                 5.3719e-30
'P8'           1                 2.6583e-62
'P9'           1                 2.9499e-12
'P10'          1                  1.418e-81
```

WA & FA

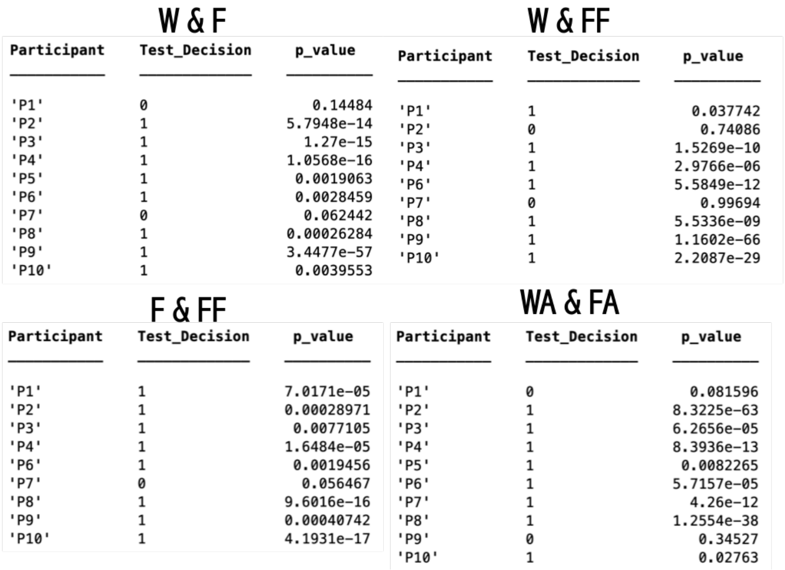
```
Participant    Test_Decision    p_value
___________    _____________    __________

'P1'           1                 9.0298e-17
'P2'           1                 1.2663e-10
'P3'           1                 1.8177e-18
'P4'           1                  0.0013828
'P5'           1                 2.6583e-05
'P6'           1                 3.7565e-12
'P7'           1                  2.518e-07
'P8'           1                 2.6108e-09
'P9'           1                  0.0038784
'P10'          1                 6.0418e-08
```

Figure 5.3: Results of the paired t-test between the five paradigms for the P300 wave amplitudes (for statistical significance p<0.05, test decision—1)

conditions as shown in Fig. 5.2. The FF condition evokes the most pronounced P300 ERPs and the W and WA conditions the least pronounced. The FF waveform is similar to the F and FA waveforms but with a higher amplitude. The conditions W and WA have identical waveforms. This suggests that the evoked potentials are mostly driven by the image visual stimuli. Comparing the waveforms of the conditions with auditory and no auditory (F with FA and W with WA) we can conclude that the waveforms are very similar. This is emphasized in Fig. 5.2 B), comparing the grand average of the P300 epochs for the five conditions in the P07 channel. Most interestingly, the most pronounced P300 waveforms are from the face paradigm conditions (F, FF, and FA), which are more pronounced in the channels placed in the fusiform face region than in other channels. Another interesting aspect is that the P300 waveforms of the channels Pz and Cz are different from the waveform from the channel PO7 in the F and FA conditions. However, in condition FF the Pz and Cz channels have more pronounced waveforms and are similar to the PO7 channel. This suggests that different perceptual/cognitive processes are triggered in visualizing a relative's face.

5.1.4.2 N400 ERP analysis

The N400 component is related to face processing, it is accentuated by face structural congruence in well-known faces and associated person information [56]. An identical analysis performed for the P300 ERP was completed for the N400 ERP. A paired t-test was calculated for the minimum peak found on the time window 450–750 ms for the 2240 epochs for the five conditions for each participant (see Figure 5.4). The paired t-tests show that for the majority of the participants the face paradigms (F, FF, and FA) negative amplitudes are significantly higher than the non-face paradigms (W and WA) amplitudes. Between the F and the FF amplitudes, there is also a significantly statically difference for all the participants except P7. The mean amplitude of the N400 minimum peaks is -16.61 µV for the condition FF, -16.33 µV for F condition, -15.19 µV for W, -14.13 µV for FA paradigm, and -13.83 µV for WA. The N400 wave has an average latency of 607.3 ms, 609.6 ms, 614.5 ms, 615.4 ms, and 616.1 ms for the W, FF, WA, F, and FA conditions, respectively.



W & F

```
Participant    Test_Decision      p_value
___________    _____________    __________

'P1'           0                   0.14484
'P2'           1                5.7948e-14
'P3'           1                  1.27e-15
'P4'           1                1.0568e-16
'P5'           1                 0.0019063
'P6'           1                 0.0028459
'P7'           0                  0.062442
'P8'           1                0.00026284
'P9'           1                3.4477e-57
'P10'          1                 0.0039553
```

W & FF

```
Participant    Test_Decision      p_value
___________    _____________    __________

'P1'           1                  0.037742
'P2'           0                   0.74086
'P3'           1                1.5269e-10
'P4'           1                2.9766e-06
'P6'           1                5.5849e-12
'P7'           0                   0.99694
'P8'           1                5.5336e-09
'P9'           1                1.1602e-66
'P10'          1                2.2087e-29
```

F & FF

```
Participant    Test_Decision      p_value
___________    _____________    __________

'P1'           1                7.0171e-05
'P2'           1                0.00028971
'P3'           1                 0.0077105
'P4'           1                1.6484e-05
'P6'           1                 0.0019456
'P7'           0                  0.056467
'P8'           1                9.6016e-16
'P9'           1                0.00040742
'P10'          1                4.1931e-17
```

WA & FA

```
Participant    Test_Decision      p_value
___________    _____________    __________

'P1'           0                  0.081596
'P2'           1                8.3225e-63
'P3'           1                6.2656e-05
'P4'           1                8.3936e-13
'P5'           1                 0.0082265
'P6'           1                5.7157e-05
'P7'           1                   4.26e-12
'P8'           1                1.2554e-38
'P9'           0                   0.34527
'P10'          1                   0.02763
```

Figure 5.4: Results of the paired t-test between the five paradigms for the N400 wave amplitudes (for statistical significance p<0.05, test decision—1).

Similar to the P300 ERPs, Fig. 5.2 shows that the N400 ERPs are most pronounced in the FF condition, and the conditions W and WA have the least pronounced N400 waveforms. The PO7, channel positioned on the fusiform face area, has a higher negative peak amplitude, on the conditions with the face paradigm (F, FF, and FA), than the Cz and Pz channels. However, in the W and WA conditions, PO7 channel presents a slight pronounced N400 peak practically nonexistent, but the Pz and Cz channels present some type of negative wave around 450–750 ms. In Fig. 5.2 B) it is clear that the N400 waveforms are similar between the WA & W conditions and between the FA & F conditions. The FF paradigm presents similar waveforms to the FA & F conditions but with higher negative peak amplitudes. This suggests that visualizing a relative's face evokes more pronounced N400 ERPs than visualizing a famous face, and even more with just viewing the word flashing.

5.1.5 Discriminative feature analysis

The R-square values, between the target and non-target epochs of the 10 participants for the 16 channels, were computed to evaluate the discriminative information derived from the five types of stimuli. Fig. 5.6 shows the temporal distributions of the most discriminative information for the five paradigms. The discriminative components match the P300 (300–450 ms) and N400 (450–750 ms) location. It is also possible to observe that before 300 ms there are discriminative components. For channels 1-5 and 13-16, there is a high discrimination for the P200 component (see also Fig. 5.2A)). P200 is typically elicited as part of the normal response to visual stimuli, the amplitude of the peak may be modulated by many different aspects of visual stimuli, such as color, orientation, shape, etc. Fig. 5.2 A) shows that the face stimuli present a higher peak than the word flashing stimuli. For the PO7 channel, a pronounced positive peak followed by a negative wave occurs before 300 ms. Since the positive peak is pronounced in the face paradigms and does not appear in the word flashing paradigms, leads us to assume that it is a vertex positive potential (VPP) ERP. VPP is a wave that peaks between 140 and 200 ms following the onset of a face stimulus [8] [35]. A negative peak is elicited by the five paradigms which was interpreted as the N200 ERP component, typically evoked 180 to 300 ms. The N200 is evoked by oddball paradigms typically suggesting a link to the cognitive processes of stimulus identification and distinction [29]. We analyze the brain areas and the different ERPs evoked by a non-face condition and a face condition, Fig. 5.5 shows the analysis for a representative participant P6. The channels were divided into two groups taking into thought the brain areas, we observe that these two areas evoke different ERPs. P6 evokes what was assumed a P200 ERP on Fz to CP6 channels, which is in agreement to previous studies that state that the P200 wave displays maximal amplitudes at the frontal electrodes [63][69]. For channels that are positioned on the visual cortex (Pz to OZ), a positive peak around 200 ms followed by a negative wave occurs. For these channels, the positive peak considered the VPP ERP, is higher on the F condition than the W condition, which is in agreement with [62][18], that state that the VPP is sensitive to face processing.

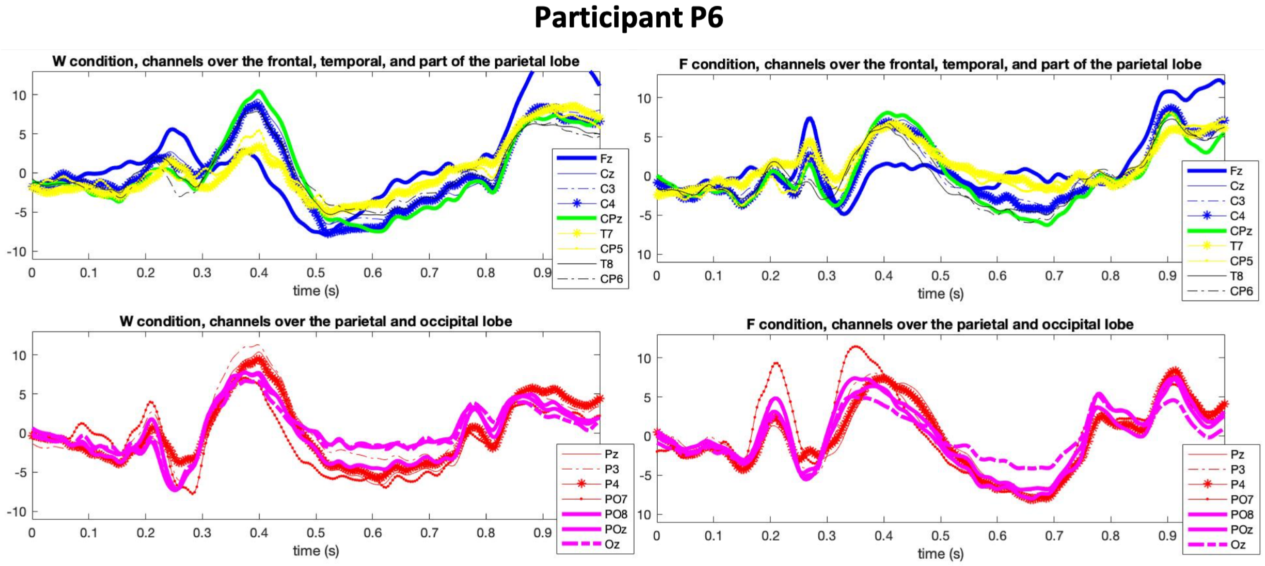


Figure 5.5: Target EEG signals for the 16 channels for F and W conditions, obtained from the participant 6.

Fig. 5.6 shows that all of the face-related stimuli yielded more discriminative features than the highlighted word after stimulus onset, which is in agreement with the online results (please note that the color scales in Fig. 5.6 are different for each condition (right vertical bar)). The most outstanding components in the features were found around 350 ms and 600 ms, which correspond to P300 and N400, respectively. The discriminative components around 450–750 ms, for the face conditions, were mainly located at the channels placed over the fusiform face area related to N400 component.

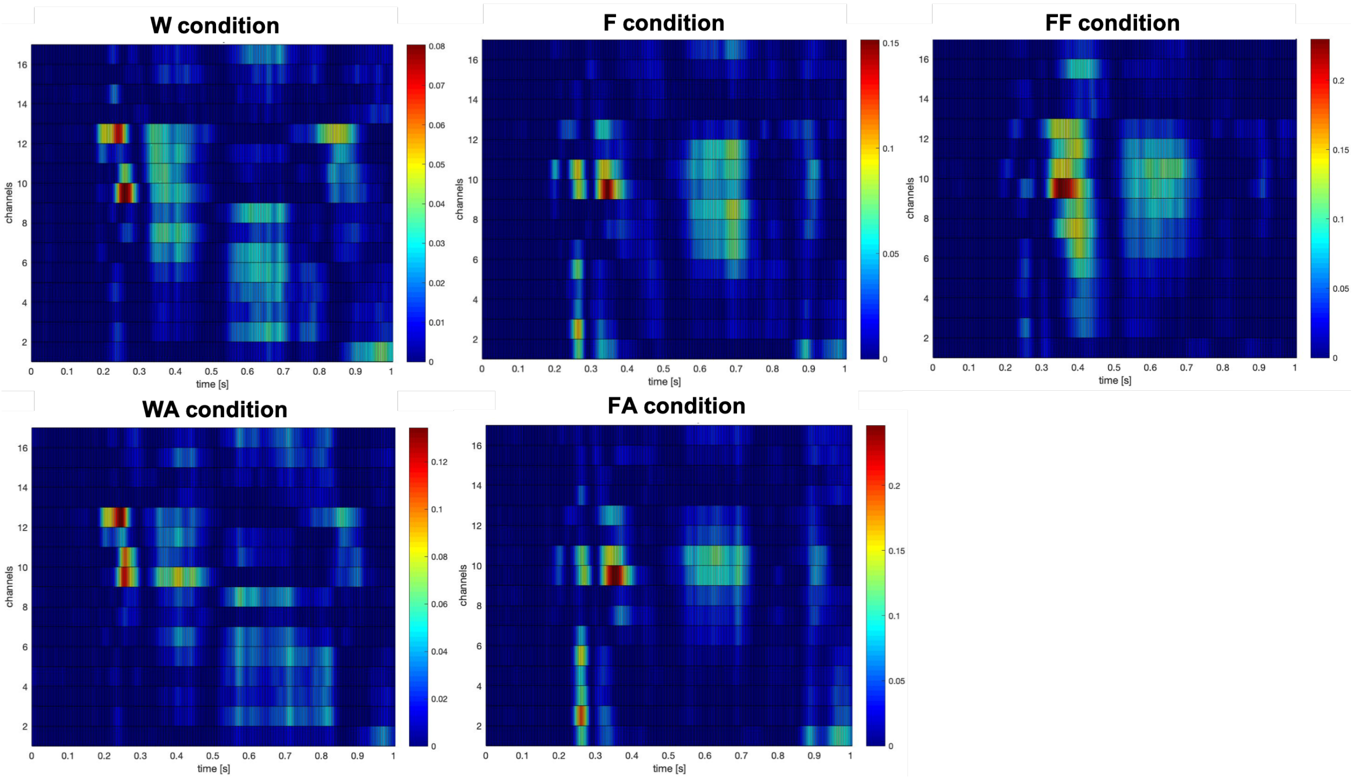


Figure 5.6: R-square color maps between target and non-target taking the 10 participants for the five conditions. (Channel numbers correspond to: 1–Fz, 2–Cz, 3–C3, 4–C4, 5–CPz, 6–Pz, 7–P3, 8–P4, 9–PO7, 10–PO8, 11–POz, 12–Oz, 13–T7, 14–CP5, 15–T8 and 16–CP6). Please note that the color scales are different for each condition (right vertical bar).

5.1.6 Discussion

Previous studies have proposed that superimposing the visual stimuli (characters, words, numbers, etc.) with images of faces, the P300 and N400 components should increase the speed and accuracy of target and non-target selection [40][41]. To further investigate the factors contributing to these effects, we present a BCI based on a hybrid paradigm (famous face images with auditory stimuli), a famous face images paradigm, and a relative's face images paradigm and compared these stimuli to a control paradigm based on text stimuli. We predicted, based on previous literature, that facial stimuli would lead to better performances than the non-facial stimuli, and our results confirmed it. Moreover, we can say that the face stimulus prompts participants to focus attention on the target more actively, concerning the obtained cognitive processes which are proved by the results. In particular, the relative's face, which has been shown to recruit additional face selectivity mechanisms in addition to those for non-relative's face, elicited the most discriminative

ERP features in our experiment. As foreseen, both offline and online results showed that the facial conditions achieved higher classification accuracies and ITRs than the non-facial conditions. The online results indicated that the facial conditions evoked larger ERP components, resulting in the improvement of BCI performance. In agreement, both ITR and eSPM were higher for those conditions. Our study shows that the stimuli of relative's faces improve the ITR of the BCI system in comparison to the rest of the paradigms. The relative's face interface had the best results and there was a ceiling effect on six participants, with online accuracies of 100%, with a mean accuracy of 95.6%. The famous face images paradigm and the hybrid paradigm (famous face images with auditory stimuli) had an online classification of 93.3%. The online results compare positively with similar studies as observed in Table 3.1. and Table 3.2. The control paradigms (non-facial stimuli) presented a classification of 82%. This suggests that stimuli with higher cognitive task requirements, facial stimuli, are more effective than the intensified stimuli of words/characters for the P300-based BCI system.

In the present study, we assessed grand-averaged ERP waveforms elicited by the target and non-target trials in all the paradigms (W, WA - control paradigms and F, FA, FF - proposed paradigms). Besides, we analyzed the difference waveforms of ERPs elicited by the target trials and compared the offline and online classification performance of all five paradigms. As can be seen from the grand average ERP in Fig. 5.2, a positive ERP component was observed at around 140–200 ms on the channel PO7 for the facial stimuli conditions, which may represent the vertex positive potential (VPP), a potential implicated in face-sensitive brain responses reflecting the neural processing of faces [78]. Another positive ERP component between 300 and 450 ms was found, which may well represent the expected P300 [5]. Also, a negative ERP component elicited by the target trials was found at around 450–750 ms on the facial conditions. This ERP component is similar to N400, which is involved in face recognition [19]. Waveform analyses indicated that the face conditions produced more robust P300 and N400 ERPs than the non-facial conditions, similar to the results observed by Kaufmann [40][41]. Both P300 and N400 achieved significantly larger amplitudes for the face-related stimuli than the word stimuli for most of the participants, Fig. 5.3 and Fig. 5.4. The P300 and N400 waves tend to be more pronounced at the parietal-occipital and occipital sites for the face-related stimuli, whereas those evoked by the highlighted word are mainly distributed at the frontal, temporal, and parietal sites. These suggest a higher level of cognitive components in the visual areas derived from the face task, in contrast to the cognitive components reflecting memory updating for the oddball task. These seem to imply that the P300 and N400 cognitive components were modulated by the face information. These cognitive components associated with face stimuli helped to yield more discriminative features than those of the highlighted word (see Fig. 5.6).

A question worthy of consideration is whether the proposed paradigms work properly for disabling subjects. Preceding studies [30][68] found that disable patients have impairments in working memory hence lower and more delayed P300s. However, no studies have stated such impairments to affect face perception and structural encoding. Although the reduced vocabulary presented by the proposed BCI is quite limited when compared with a BCI-speller, it may represent one way to achieve a feasible BCI adapted to the skills and needs of disabling patients. To investigate in detail the applicability for disabled patients, the presented paradigms should be further tested in patients with different cognitive impairments.

5.2 Convolutional Neural Network

This chapter chapter analyzes whether a CNN classifier can be used on a real BCI application. The CNN is compared with the approach currently being used in our lab which consists of a statistical spatial filter combined with a Naïve-Bayes classifier.

5.2.1 Datasets

The datasets were collected from the calibration sessions conducted with the 10 participants who tested the W, F, and FF paradigms. Data and paradigm specifications were already explained in section 4.4.3. Ten possible target ERPs should be detected for each target symbol (10 repetitions per trial, and 14 targets) on the calibration session and 3 to 6 from the online experiments.

For the training database, the number of P300 epochs to detect is 14 x 10, and there are 14 x 10 x 7 - 140 = 840 epochs labelled ”non-P300”. The number of samples labeled ”P300” and ”non-P300”, for the online experiments, vary from participant to participant. For P1, P8, P9, and P10 the number of epochs chosen was 3, hence the number of P300 to detect is 15 x 3 = 45 and the number of non-P300 is 15 x 3 x 7 - 45 = 270. For P4 the number of epochs is 4, therefore, 15 x 4 = 60 number of P300 to detect and is 15 x 4 x 7 - 60 = 360. P3, P5, and P6 have 5 number of epochs which corresponds to 75 sets of signal samples labeled ”P300” and 450 sets labeled ”non-P300”. For P2 and P7 the number of epochs chosen was 6, therefore, 90 number of P300 to detect and 540 number of non-P300 to detect.

5.2.2 Experimental results

Fig. 5.7 presents the offline binary classification rate in percentage for each classification method for each W, F, and FF conditions and for each participant. When the number of epochs is n, it indicates that only the n first epochs are considered. If n is equal to one, there is only one P300 possible response for determining a symbol (called single trial detection).

The goal of this analysis is to compare the performance of the CNN classifier with the performance of the FCB spatial filter + Naïve-Bayes classifier, therefore, the differences between the conditions (W, F, and FF) are not relevant for the analysis. The mean accuracy of the 10 participants is shown in Fig. 5.7 (mean). For the three tested conditions, ”FCB+Naïve-Bayes” has a binary classification accuracy higher than 85% for all the number of epochs. The CNN has a low accuracy for initial n’s, nevertheless, as n increases the accuracy progressively increases as well. The best accuracy achieved by CNN is when $n = 6$ with 82.38%, 82.85%, and 89.52% for the three conditions. The progression of the accuracy concerning n is not linear, in some cases adding more epochs does

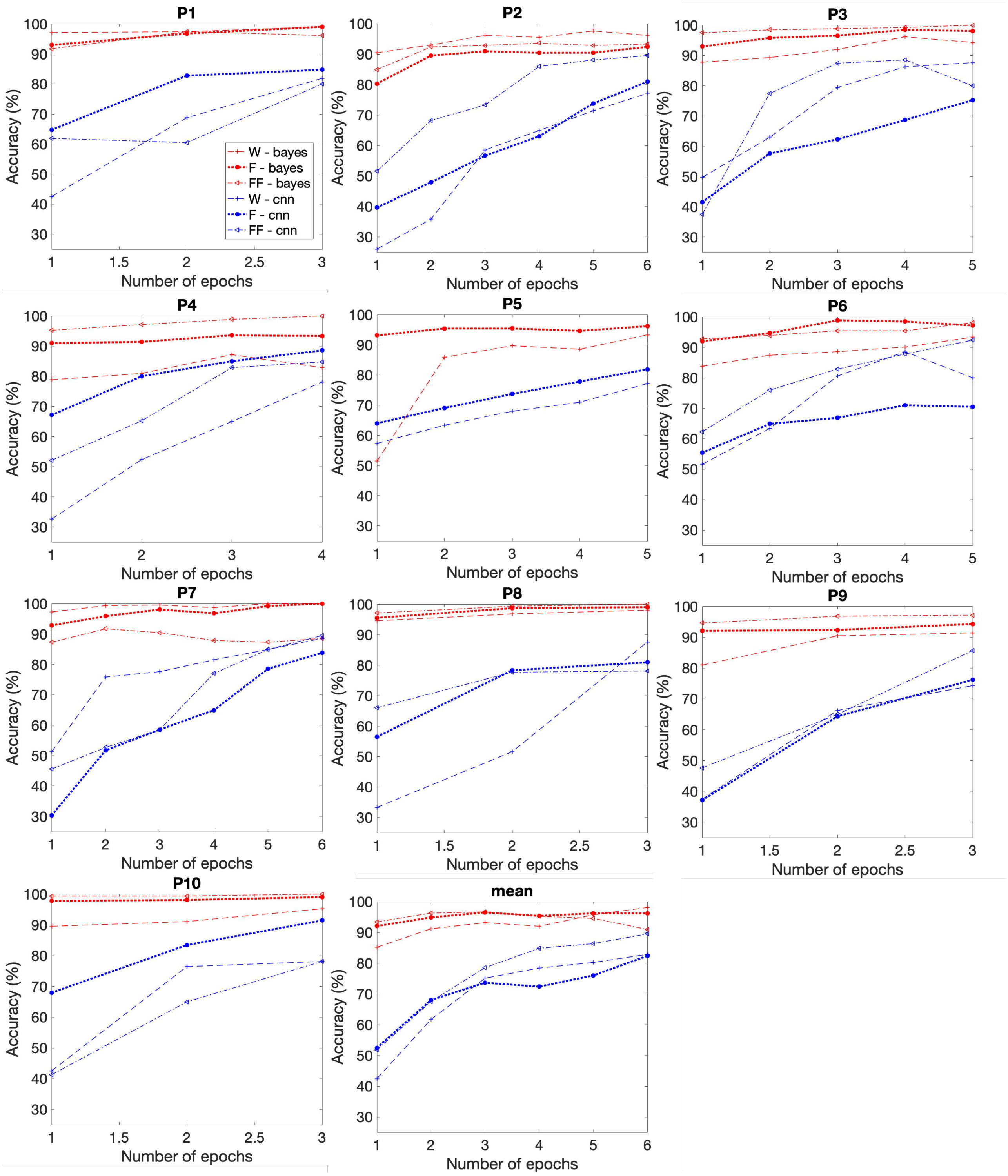


Figure 5.7: Percentage of the character recognition rate for the different classifiers, FCB Spatial filter + Naïve-Bayes and CNN, for the conditions W, F, and FF. In the legend ”bayes” refers to FCB spatial filter plus Naïve-Bayes.

not necessarily increase accuracy. Fig. 5.8 simulates the online ITR, in bits per minute (bpm), concerning the number of considered epochs, i.e., over the time required for the recognition of a word. The performance of the Naïve-Bayes classifier presents higher accuracies than the CNN classifier.

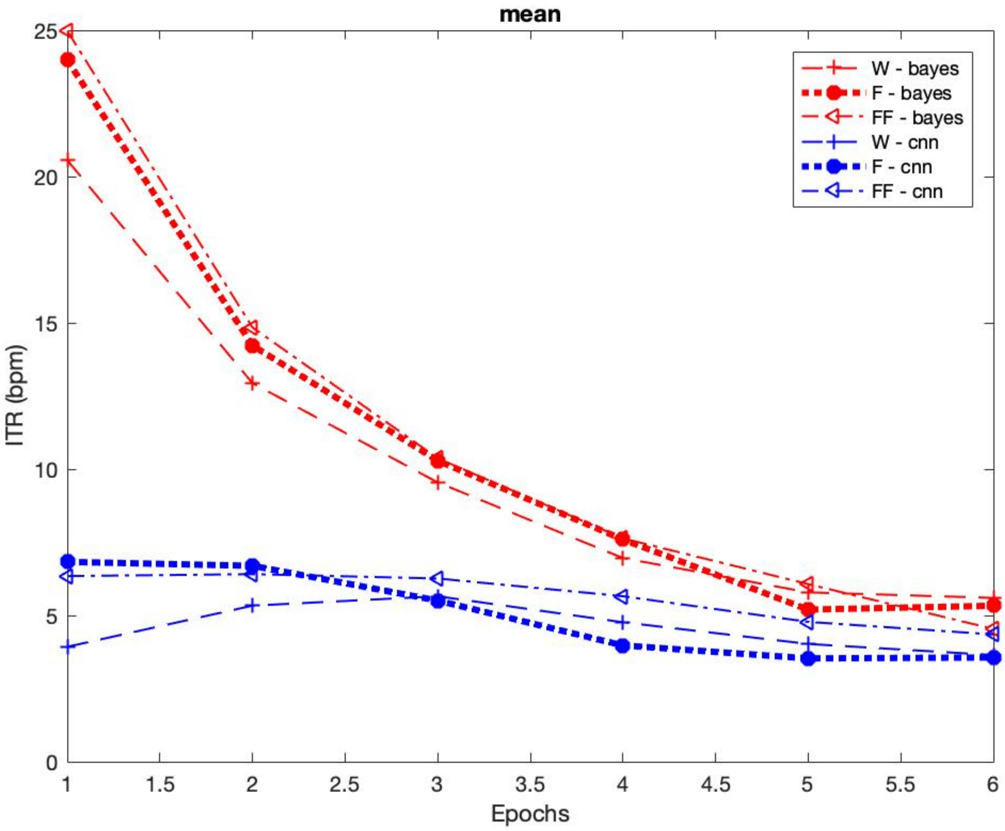


Figure 5.8: ITR (in bits per minute) in relation to the number of epochs (mean of the 10 participants per condition).

5.2.3 Discussion

One of the goals of the BCI community is to make BCI systems more practical and reliable. Deep neural networks have shown state of the art performance in computer vision and speech recognition and thus have great promise for other learning tasks, such as multidimentional EEG data. The CNN method does not require any electrode selection before the classification. All of the electrodes are handled with no neuroscience knowledge about the best electrodes or some preceding features selection, i.e., it does not consider a prior set of selected features or high-level features as input. The classification is done directly on the EEG signals signals after basic preprocessing. It classifies a signal without directly considering the usual shape of the expected signal to detect. Another important aspect of CNN is that it may allow deeper analysis of brain activity, particular features can be discovered in the training step. This method does not separate the different parts of the classification (features selection, spatial filters, etc.) as most of the classifiers, instead, a CNN has the ability to learn and extract all of the appropriate features from the input data automatically by optimizing the weight parameters of each filter through the forward and backward propagation to decrease the classification errors. These advantages are important for its implementation in a real BCI, where its all embedded approach can highlight the subject particularities without any tuning. Thus, although the CNN performed worse than the current approach, the aforementioned CNN properties are very relevant and give the motivation to further explore deep learning approaches.

By increasing n, the CNN accuracy becomes more similar to the accuracies on the Bayes method. The CNN is capable of adequately classifying the data, however, the algorithm needs to be improved. The CNN performs a spatial convolution with kernel (1, C), where C denotes the number of electrodes, in the first layer which makes the CNN not able to learn temporal features well. Every column in the input matrix contains a set of C signal samples. These samples come from C electrodes at a certain sampling time point. The spatial convolution operation converts each column of spatial data into an abstract datum in a feature map. The spatial convolution layer outputs several feature maps, which are given as input to the temporal convolution layer. These feature maps are abstract temporal signals instead of raw temporal signals. Thus, the spatial convolution operation leads to losing raw temporal information, which means loosing important P300-related features. As a result, the network can not learn temporal features well. A possible solution is to apply both a spatial and temporal convolution in the first layer, allowing the CNN to learn features from raw temporal and spatial information simultaneously. We also need to take into consideration that this is an exploratory work with a limited number of datasets, where the CNN was trained with a small amount of data. CNNs usually require a big amount of training datasets to present high performances, however collecting more data would require a lot of time and it was not possible.

6

Conclusions and further work

The goal of this dissertation was to develop a standalone framework to implement visual and auditory paradigms in applications outside the Matlab/Simulink system. The framework developed is not only a framework for the rapid development of feedback and stimulus applications but also a platform to run neuroscientific tests independent from BCI systems. Three P300-based conditions using the faces paradigm (famous face images, relative's face images, and famous face images + auditory) were proposed and developed, and two P300-based on text stimuli conditions (words and words + auditory) were replicated from [3] to serve as control paradigms. The performance of the proposed BCIs was significantly improved in comparison to the control P300-based BCIs. The best online accuracy, ITR, and effective SPM was achieved by the relative's face condition and compares favorably with the state-of-the-art performances. An analysis of the discriminative ERP features was performed. The relative's face condition showed to recruit additional face selectivity mechanisms in addition to those for non-relative's face, eliciting the most discriminative ERP features. Future work will focus on the use of faces in ERP–BCI controlled applications, in particular, to test it with complete locked-in state patients.

Another issue was addressed in this dissertation, a CNN approach proposed in [11] was implemented offline and it was compared with the approach currently being used at ISR-UC, based on a statistical spatial filter and a Bayes classifier. An advantage of the CNN classifier is that it does not consider a prior set of selected features or high-level features as input, also, the classification is done directly on the EEG signals after basic preprocessing. The ”statistical spatial filter + Bayes classifier” approach showed better performances than the CNN, nevertheless, the CNN was capable of adequately classifying the data. Although the CNN performed worse than the current approach, the aforementioned CNN properties are very relevant and give the motivation to further explore deep learning approaches. Future work will focus on improving the CNN by performing both spatial convolution and temporal convolution in the first layer. Another intersting aspect to continue this exploratory work is to try to make the CNN subject invariant, which would allow the use of BCIs without the calibration tests. This issue requires the collection of large amount of datasets to train the CNN.

7

Appendix A

Platforms for BCIs

1. BCI2000 [65] supports a variety of data acquisition systems, brain signals, and study/feedback paradigms. BCI200 provides support for 19 different data acquisition systems including the major EEG amplifiers. It also provides a customized auditory/visual stimulation that is synchronized with the signal acquisition. The BCI2000 platform separates the BCI system in different modules on different levels, in the first level, we have four modules: source (signal acquisition and storage) module, the signal processing, user application, and operator interface (system control). The Source module acts as an input and receives neural signals from the brain. The signal processing module transforms the brain signals into an output signal. The User application module uses the output signal to drive the specific BCI program in use. The modules communicate with each other over TCP/IP connections. The second level is modularized into a sequence of pre-processing filters that operate on the signals, which have a common programming interface. A third level exists for re-usable software building blocks, that support the creation of new processing filters or application modules. BCI2000 provides a comprehensive document for the programmers that describes the data structures, data types, internal events in the system, and it explains the process on how to extend BCI20000 with new modules like the acquisition, signal processing or application. BCI2000 is written in C++ but provides a programming interface that allows accessing system parameters, data signals, and event information intuitively. This platform also allows the signal processing code to be written in MATLAB and includes Python compatibility. For more compatibility with other programming languages and external applications, BCI2000 can be loadable as a library and can be wrapped into an application that accesses this library. Also, the BCI2000 internal state can be sent over UDP connection (user datagram protocol) which an external application can access.

2. OpenViBE [9] is a free and open-source software platform for designing, testing, and using brain-computer interfaces. Similar to the BCI200 the OpenViBE also consists of a set of modules, these modules can easily be consolidated to develop functional interfaces. The software modules are acquisition, pre-processing, processing, and visualization of brain data. It also supports various acquisition devices such as EEG or MEG amplifiers. The OpenViBE

platform can be integrated with other applications. The OpenViBE platform includes several useful tools: acquisition server, designer, 2D visualization tools, and sample scenarios for BCIs or neurofeedback applications.

3. TOBI [1] is a cross-platform set of interfaces that connect parts with different BCI systems, to facilitate distributed BCI research between different systems and platforms. These different systems transmit the data acquired, extracted features, classifier outputs, and events in a standardized way. The design of this Common Implementation Platform (CIP) is based on a pipeline system. The data is acquired via a data acquisition system and delivered to a data processing module. Since the TOBI platform supports multiple processing streams different processing pipes exist. The modules are connected by four interfaces A, B, C and D. The interface A transmits raw biosignals, B transmits signal features such as band power, C transmits detected classes and class labels and D is used to transmit events and markers within the CIP.

4. BCI++ [10] is an open-source framework that provides a set of tools for the development of BCIs. It is composed of two principal modules, which establish communications over TCP/IP connections. One module is the Hardware Interface Module (HIM) for signal acquisition, storage, visualization, and real-time processing. HIM has a main block that takes care of tasks common to all protocols and loads plug-ins, in which the user develops the algorithms. Algorithms for signal processing can be designed in C/C++ and MATLAB. BCI++ has many tools that simplify the development of updates without developing from scratch. HIM supports many signal acquisition devices and provides compatibility with some of the devices developed in selected laboratories. The other module is called AEnima which provides a Graphical User Interface (GUI), and it's for the creation and management of different protocols based on the 2D/3D graphics engine. AEnima is based on a sophisticated graphics engine so a more realistic experience is provided for the user.

5. xBCI [72] is a general-use platform for developing online BCIs. The platform provides an easy system development tool with extendable and modular system design, GUI-based system development, multi-threaded parallel processing, multi-OS support, and open-source. The developers can design and build different types of BCI systems by combining these different components: basic mathematical operations, data processing, data acquisition, network communications, data visualization, experiment control, real-time feedback presentation. The user can also develop new components, in C++ or a scripting language. Every component is completely independent as a plug-in, they are executed in their thread.

6. Pythonic feedback framework (Pyff) [76], was created for the development of experimental paradigms and a platform to run neuroscientific experiments. The goal is to create BCI stimulus and feedback presentation applications fast and easy. The framework is implemented in Python which is a high-level programming language that can be relatively easy to learn. The framework is also compatible with many different BCI systems, by coupling Pyff with the rest of the BCI system using UDP connections to transport data from the system to the Pyff

and XML to wrap arbitrary data in a format that Pyff can read. Pyff also supports the TOBI interface for communications between the BCI system modules. Pyff has four parts, the first one is the feedback controller which receives signals from the BCI system and translates and forwards them to the feedback and stimulus application. The second part is the graphical user interface that controls the feedback controller remotely over the network. The third is a set of feedback paradigms and stimuli, and the fourth part is a collection of base classes which provides methods and functionality shared by feedback and stimulus applications.

Bibliography

[1] Brendan Z Allison, Stephen Dunne, Robert Leeb, José del R Millán, and Anton Nijholt. Recent and upcoming bci progress: overview, analysis, and recommendations. In *Towards Practical Brain-Computer Interfaces*, pages 1–13. Springer, 2012.

[2] Xingwei An, Johannes Höhne, Dong Ming, and Benjamin Blankertz. Exploring combinations of auditory and visual stimuli for gaze-independent brain-computer interfaces. *PloS one*, 9(10):e111070, 2014.

[3] Sara Barbosa, Gabriel Pires, and Urbano Nunes. Toward a reliable gaze-independent hybrid bci combining visual and natural auditory stimuli. *Journal of neuroscience methods*, 261:47–61, 2016.

[4] Gerhard Bauer, Franz Gerstenbrand, and Erik Rumpl. Varieties of the locked-in syndrome. *Journal of neurology*, 221(2):77–91, 1979.

[5] Edward Bernat, Howard Shevrin, and Michael Snodgrass. Subliminal visual oddball stimuli evoke a p300 component. *Clinical neurophysiology*, 112(1):159–171, 2001.

[6] Niels Birbaumer. Brain-computer-interface research: coming of age. *Clinical neurophysiology*, 2006.

[7] Niels Birbaumer, Francesco Piccione, Stefano Silvoni, and Moritz Wildgruber. Ideomotor silence: the case of complete paralysis and brain–computer interfaces (bci). *Psychological research*, 76(2):183–191, 2012.

[8] K Bötzel and O-J Grüsser. Electric brain potentials evoked by pictures of faces and non-faces: a search for "face-specific" eeg-potentials. *Experimental Brain Research*, 77(2):349–360, 1989.

[9] Laurent Bougrain and Guillaume Serrière. Classification of brain signals with openvibe. *Brain–Computer Interfaces 2: Technology and Applications*, pages 211–225, 2016.

[10] Clemens Brunner, Giuseppe Andreoni, Lugi Bianchi, Benjamin Blankertz, Christian Breitwieser, Shin'ichiro Kanoh, Christian A Kothe, Anatole Lécuyer, Scott Makeig, Jürgen Mellinger, et al. Bci software platforms. In *Towards Practical Brain-Computer Interfaces*, pages 303–331. Springer, 2012.

[11] Hubert Cecotti and Axel Graser. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):433–445, 2010.

[12] Srivas Chennu, Abdulmajeed Alsufyani, Marco Filetti, Adrian M Owen, and Howard Bowman. The cost of space independence in p300-bci spellers. *Journal of neuroengineering and rehabilitation*, 10(1):82, 2013.

[13] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.

[14] Aniana Cruz, Gabriel Pires, and Urbano J Nunes. Double errp detection for automatic error correction in an erp-based bci speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(1):26–36, 2018.

[15] Gaochao Cui, Qibin Zhao, Jianting Cao, and Andrzej Cichocki. Hybrid-bci: Classification of auditory and visual related potentials. In *Soft Computing and Intelligent Systems (SCIS), 2014 Joint 7th International Conference on and Advanced Intelligent Systems (ISIS), 15th International Symposium on*, pages 297–300. IEEE, 2014.

[16] Jerome Daltrozzo and Christopher M Conway. Neurocognitive mechanisms of statistical-sequential learning: what do event-related potentials tell us? *Frontiers in human neuroscience*, 8:437, 2014.

[17] Maarten De Vos, Katharina Gandras, and Stefan Debener. Towards a truly mobile auditory brain–computer interface: exploring the p300 to take away. *International journal of psychophysiology*, 91(1):46–53, 2014.

[18] Martin Eimer. Effects of face inversion on the structural encoding and recognition of faces: Evidence from event-related brain potentials. *Cognitive Brain Research*, 10(1-2):145–158, 2000.

[19] Martin Eimer. Event-related brain potentials distinguish processing stages involved in face perception and recognition. *Clinical neurophysiology*, 111(4):694–705, 2000.

[20] Lawrence Ashley Farwell and Emanuel Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523, 1988.

[21] Adrian Furdea, Sebastian Halder, DJ Krusienski, Donald Bross, Femke Nijboer, Niels Birbaumer, and Andrea Kübler. An auditory oddball (p300) spelling system for brain-computer interfaces. *Psychophysiology*, 46(3):617–625, 2009.

[22] AUSTRIA G.TEC MEDICAL ENGINEERING GMBH. g.usbamp - gtec's highest accuracy biosignal data acquisition and processing system, 2019.

[23] S Halder, M Rea, R Andreoni, F Nijboer, EM Hammer, SC Kleih, N Birbaumer, and A Kübler. An auditory oddball brain–computer interface for binary choices. *Clinical Neurophysiology*, 121(4):516–523, 2010.

[24] Sebastian Halder, Eva Maria Hammer, Sonja Claudia Kleih, Martin Bogdan, Wolfgang Rosenstiel, Niels Birbaumer, and Andrea Kübler. Prediction of auditory and visual p300 brain-computer interface aptitude. *PloS one*, 8(2):e53513, 2013.

[25] Bin He. *Neural engineering*. Springer Science & Business Media, 2007.

[26] N Jeremy Hill, Thomas N Lal, Karin Bierig, Niels Birbaumer, and Bernhard Schölkopf. An auditory paradigm for brain-computer interfaces. In *Advances in neural information processing systems*, pages 569–576, 2005.

[27] N Jeremy Hill, Erin Ricci, Sameah Haider, Lynn M McCane, Susan Heckman, Jonathan R Wolpaw, and Theresa M Vaughan. A practical, intuitive brain–computer interface for communicating 'yes' or 'no'by listening. *Journal of neural engineering*, 11(3):035003, 2014.

[28] NJ Hill and Bernhard Schölkopf. An online brain–computer interface based on shifting attention to concurrent streams of auditory stimuli. *Journal of neural engineering*, 9(2):026011, 2012.

[29] JE Hoffman. Event-related potentials and automatic and controlled processes. *Event-related brain potentials; Basic issues and applications*, 1990.

[30] Ulrich Hoffmann, Jean-Marc Vesin, Touradj Ebrahimi, and Karin Diserens. An efficient p300-based brain–computer interface for disabled subjects. *Journal of Neuroscience methods*, 167(1):115–125, 2008.

[31] Johannes Höhne, Konrad Krenzlin, Sven Dähne, and Michael Tangermann. Natural stimuli improve auditory bcis with respect to ergonomics and performance. *Journal of Neural Engineering*, 9(4):045003, 2012.

[32] Johannes Höhne, Martijn Schreuder, Benjamin Blankertz, and Michael Tangermann. A novel 9-class auditory erp paradigm driving a predictive text entry system. *Frontiers in neuroscience*, 5:99, 2011.

[33] Richard W Homan, John Herman, and Phillip Purdy. Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology*, 66(4):376–382, 1987.

[34] Han-Jeong Hwang, Soyoun Kim, Soobeom Choi, and Chang-Hwan Im. Eeg-based brain-computer interfaces: a thorough literature survey. *International Journal of Human-Computer Interaction*, 29(12):814–826, 2013.

[35] DA Jeffreys. A face-responsive potential recorded from the human scalp. *Experimental brain research*, 78(1):193–202, 1989.

[36] Richard Jung and Wiltrud Berger. Hans bergers entdeckung des elektrenkephalogramms und seine ersten befunde 1924–1931. *Archiv für Psychiatrie und Nervenkrankheiten*, 227(4):279–300, 1979.

[37] Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, 17(11):4302–4311, 1997.

[38] Ivo Käthner, Andrea Kübler, and Sebastian Halder. Comparison of eye tracking, electrooculography and an auditory brain-computer interface for binary communication: a case study with a participant in the locked-in state. *Journal of neuroengineering and rehabilitation*, 12(1):76, 2015.

[39] Ivo Käthner, Selina C Wriessnegger, Gernot R Müller-Putz, Andrea Kübler, and Sebastian Halder. Effects of mental workload and fatigue on the p300, alpha and theta band power during operation of an erp (p300) brain–computer interface. *Biological psychology*, 102:118–129, 2014.

[40] Tobias Kaufmann, SM Schulz, Claudia Grünzinger, and Andrea Kübler. Flashing characters with famous faces improves erp-based brain–computer interface performance. *Journal of neural engineering*, 8(5):056016, 2011.

[41] Tobias Kaufmann, Stefan M Schulz, Anja Köblitz, Gregor Renner, Carsten Wessig, and Andrea Kübler. Face stimuli effectively prevent brain–computer interface inefficiency in patients with neurodegenerative disease. *Clinical Neurophysiology*, 124(5):893–900, 2013.

[42] Andrea Kübler and Niels Birbaumer. Brain–computer interfaces and communication in paralysis: Extinction of goal directed thinking in completely paralysed patients? *Clinical neurophysiology*, 119(11):2658–2666, 2008.

[43] Andrea Kübler, Adrian Furdea, Sebastian Halder, Eva Maria Hammer, Femke Nijboer, and Boris Kotchoubey. A brain–computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients. *Annals of the New York Academy of Sciences*, 1157(1):90–100, 2009.

[44] Andrea Kübler, Nicola Neumann, Jochen Kaiser, Boris Kotchoubey, Thilo Hinterberger, and Niels P Birbaumer. Brain-computer communication: self-regulation of slow cortical potentials for verbal communication. *Archives of physical medicine and rehabilitation*, 82(11):1533–1539, 2001.

[45] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[46] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[47] Damien Lesenfants, Dina Habbal, Z Lugo, M Lebeau, Petar Horki, E Amico, Christoph Pokorny, F Gomez, A Soddu, Gernot Müller-Putz, et al. An independent ssvep-based brain–computer interface in locked-in syndrome. *Journal of neural engineering*, 11(3):035002, 2014.

[48] Qi Li, Shuai Liu, Jian Li, and Ou Bai. Use of a green familiar faces paradigm improves p300-speller brain-computer interface performance. *PloS one*, 10(6):e0130325, 2015.

[49] Mingfei Liu, Wei Wu, Zhenghui Gu, Zhuliang Yu, FeiFei Qi, and Yuanqing Li. Deep learning based on batch normalization for p300 signal detection. *Neurocomputing*, 275:288–297, 2018.

[50] Ana Lopes, Joao Rodrigues, Jorge Perdigao, Gabriel Pires, and Urbano Nunes. A new hybrid motion planner: applied in a brain-actuated robotic wheelchair. *IEEE Robotics & Automation Magazine*, 23(4):82–93, 2016.

[51] Ana C Lopes, Gabriel Pires, and Urbano Nunes. Assisted navigation for a brain-actuated intelligent wheelchair. *Robotics and Autonomous Systems*, 61(3):245–258, 2013.

[52] Joseph N Mak and Jonathan R Wolpaw. Clinical applications of brain-computer interfaces: current state and future prospects. *IEEE reviews in biomedical engineering*, 2:187–199, 2009.

[53] Ran Manor and Amir B Geva. Convolutional neural network for multi-category rapid serial visual presentation bci. *Frontiers in computational neuroscience*, 9:146, 2015.

[54] Laurie A Miner, Dennis J McFarland, and Jonathan R Wolpaw. Answering questions with an electroencephalogram-based brain-computer interface. *Archives of physical medicine and rehabilitation*, 79(9):1029–1033, 1998.

[55] Kevin P Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18:60, 2006.

[56] Ela I Olivares, Jaime Iglesias, Cristina Saavedra, Nelson J Trujillo-Barreto, and Mitchell Valdés-Sosa. Brain signals of face processing as revealed by event-related potentials. *Behavioural neurology*, 2015, 2015.

[57] Gabriel Pires, Urbano Nunes, and Miguel Castelo-Branco. Gibs block speller: Toward a gaze-independent p300-based bci. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 6360–6364. IEEE, 2011.

[58] Gabriel Pires, Urbano Nunes, and Miguel Castelo-Branco. Statistical spatial filtering for a p300-based bci: tests in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis. *Journal of neuroscience methods*, 195(2):270–281, 2011.

[59] Gabriel Pires, Urbano Nunes, and Miguel Castelo-Branco. Comparison of a row-column speller vs. a novel lateral single-character speller: assessment of bci for severe motor disabled patients. *Clinical Neurophysiology*, 123(6):1168–1181, 2012.

[60] Gabriel Pereira Pires. *Biosignal Classification for Human Interface with Devices and Surrounding Environment*. PhD thesis, 2011.

[61] Christoph Pokorny, Daniela S Klobassa, Gerald Pichler, Helena Erlbeck, Ruben GL Real, Andrea Kübler, Damien Lesenfants, Dina Habbal, Quentin Noirhomme, Monica Risetti, et al. The auditory p300-based single-switch brain–computer interface: paradigm transition from healthy subjects to minimally conscious patients. *Artificial intelligence in medicine*, 59(2):81–90, 2013.

[62] Bruno Rossion, Salvatore Campanella, Carlos M Gomez, A Delinte, Damien Debatisse, Laura Liard, Stéphanie Dubois, Raymond Bruyer, Marc Crommelinck, and J-M Guerit. Task modulation of brain activity related to familiar and unfamiliar face processing: an erp study. *Clinical Neurophysiology*, 110(3):449–462, 1999.

[63] Michael D Rugg. Event-related potentials in phonological matching tasks. *Brain and language*, 23(2):225–240, 1984.

[64] Lisa D Sanders and Helen J Neville. An erp study of continuous speech processing: I. segmentation, semantics, and syntax in native speakers. *Cognitive Brain Research*, 15(3):228–240, 2003.

[65] Gerwin Schalk and Jürgen Mellinger. *A practical guide to brain–computer interfacing with BCI2000: General-purpose software for brain-computer interface research, data acquisition, stimulus presentation, and brain monitoring*. Springer Science & Business Media, 2010.

[66] Martijn Schreuder, Benjamin Blankertz, and Michael Tangermann. A new auditory multiclass brain-computer interface paradigm: spatial hearing as an informative cue. *PloS one*, 5(4):e9813, 2010.

[67] ERIC Sellers, GERWIN Schalk, and EMANUEL Donchin. The p300 as a typing tool: tests of brain computer interface with an als patient. *Psychophysiology*, 40(S1):s77, 2003.

[68] Eric W Sellers and Emanuel Donchin. A p300-based brain–computer interface: initial tests by als patients. *Clinical neurophysiology*, 117(3):538–548, 2006.

[69] Martin I Sereno. Brain mapping in animals and humans. *Current opinion in neurobiology*, 8(2):188–194, 1998.

[70] Leila Azinfar Setare Amiri, Ahmed Rabbi and Reza Fazel-Rezai. A review of p300, ssvep, and hybrid p300/ssvep brain- computer interface systems.

[71] Hongchang Shan, Yu Liu, and Todor P Stefanov. A simple convolutional neural network for accurate p300 detection and character spelling in brain computer interface. In *IJCAI*, pages 1604–1610, 2018.

[72] I Putu Susila, Shin'ichiro Kanoh, Ko-ichiro Miyamoto, and Tatsuo Yoshinobu. xbci: a generic platform for development of an online bci system. *IEEJ Transactions on Electrical and Electronic Engineering*, 5(4):467–473, 2010.

[73] Samuel Sutton, Margery Braren, Joseph Zubin, and ER John. Evoked-potential correlates of stimulus uncertainty. *Science*, 150(3700):1187–1188, 1965.

[74] George Townsend, BK LaPallo, CB Boulay, DJ Krusienski, GE Frye, CKea Hauser, NE Schwartz, TM Vaughan, Jonathan R Wolpaw, and EW Sellers. A novel p300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clinical Neurophysiology*, 121(7):1109–1120, 2010.

[75] George Townsend, BK LaPallo, CB Boulay, DJ Krusienski, GE Frye, CKea Hauser, NE Schwartz, TM Vaughan, Jonathan R Wolpaw, and EW Sellers. A novel p300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clinical neurophysiology*, 121(7):1109–1120, 2010.

[76] Bastian Venthur, Simon Scholler, John Williamson, Sven Dähne, Matthias S Treder, Maria T Kramarek, Klaus-Robert Müller, and Benjamin Blankertz. Pyff—a pythonic framework for feedback applications and stimulus presentation in neuroscience. *Frontiers in neuroscience*, 4:179, 2010.

[77] Seul-Ki Yeom, Siamac Fazli, Klaus-Robert Müller, and Seong-Whan Lee. An efficient erp-based brain-computer interface using random set presentation and face familiarity. *PloS one*, 9(11):e111157, 2014.

[78] Yu Zhang, Qibin Zhao, Jing Jin, Xingyu Wang, and Andrzej Cichocki. A novel bci based on erp components sensitive to configural processing of human faces. *Journal of neural engineering*, 9(2):026018, 2012.