

Renato de Macedo Giovanini

SSVEP-EEG Signal Pattern Recognition System for Real-Time Brain-Computer Interfaces Applications

Ilha Solteira

2017

Renato de Macedo Giovanini

SSVEP-EEG Signal Pattern Recognition System for Real-Time Brain-Computer Interfaces Applications

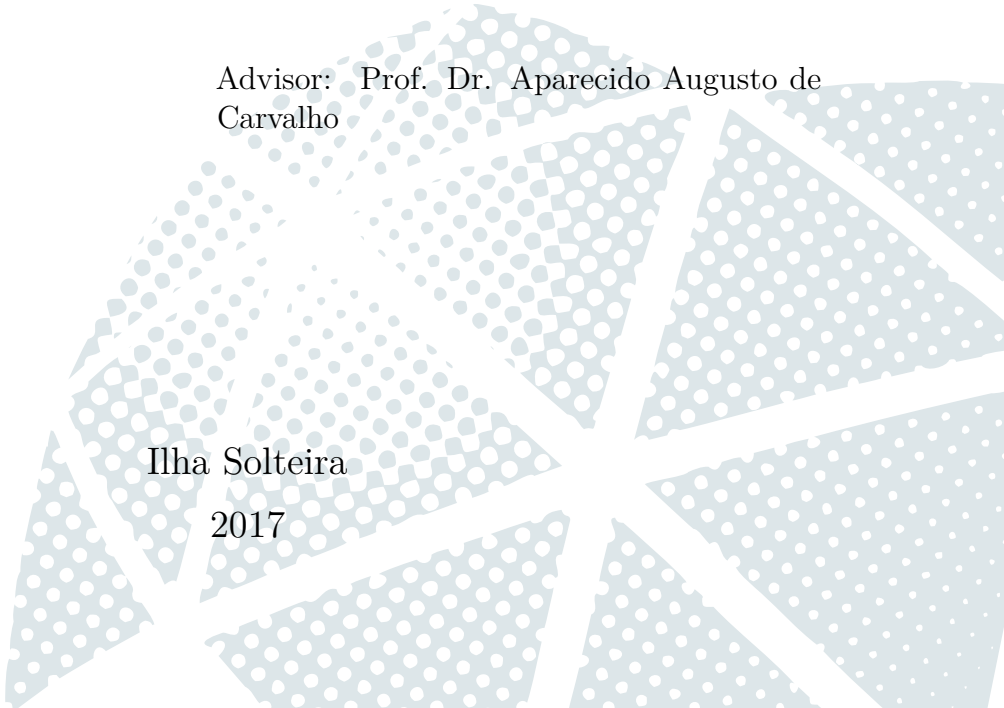
Thesis submitted to the Faculdade de Engenharia de Ilha Solteira as partial requisite to obtain the degree of Master in Electrical Engineering.

Field of Expertise: Automation

Advisor: Prof. Dr. Aparecido Augusto de Carvalho

Ilha Solteira

2017



FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Giovanini, Renato de Macedo.

G512s SSVEP signal pattern recognition system for real-time brain-computer applications
/ Renato de Macedo Giovanini. – Ilha Solteira:[s.n.], 2017
89f.: il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2017.

Orientador: Prof. Dr. Aparecido Augusto de Carvalho
Inclui bibliografia

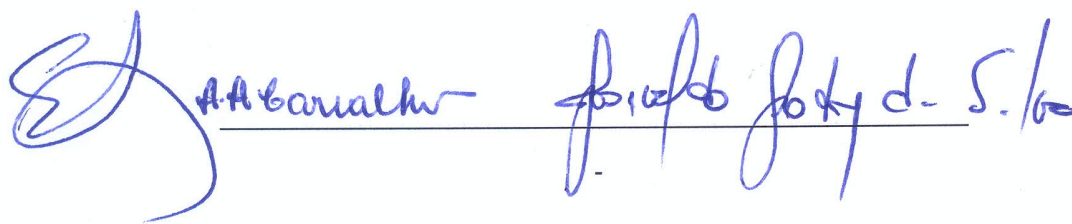
1. Pattern recognition 2. Aprendizado de máquina 3. Interface cérebro-máquina 4. Interface cérebro-computador 5. Python 6. Raspberry Pi.

ATESTADO DE APROVAÇÃO - DEFESA

Atestamos que **RENATO DE MACEDO GIOVANINI**, RA nº: 178999-3, RG nº 44.959.278-9, expedido pela SSP/SP, defendeu, no dia 18/08/2017, a dissertação intitulada **SSVEP-EEG Signal Pattern Recognition System for Real-Time Brain-Computer Interfaces**, junto ao Programa de Pós Graduação em ENGENHARIA ELÉTRICA, Curso de Mestrado Acadêmico, tendo sido 'APROVADO'.

Atestamos ainda que a obtenção do título dependerá de homologação pelo Órgão Colegiado competente.

Ilha Solteira, 18 de agosto de 2017

A handwritten signature in blue ink is written over a horizontal line. The signature is stylized and appears to read "A. A. Baralho" followed by "prof. adj. d. S. 16".

This thesis is dedicated to my family. Giovana, Marli, Jorge. Thank you for everything.

Acknowledgements

To God and Our Lady of Aparecida for the protection.

To Marli and Jorge, my parents, for the examples and understanding. For pointing me the way before I even learned to walk.

To Giovana and Tadeu, for teaching me how to keep fighting, no matter the job, weather or language.

To professor Dr. Aparecido, for the advisement, friendship and encouragement to carry out my researches and challenges.

To professor Dr. Emanuel Woiski for the collaborative examples, mood and disposition to exchanging ideas.

To the FEIS-UNESP employees and technicians, in the name of the technician and friend Everaldo, for all the hard and respectful work.

To my friends and colleagues Luiz, Thiago, Ricardo Taoni, Etieny, Bruno, Ricardo Gouveia, Danilo, Caroline, Mariana, Jorge, Fabian, Alan, Marcela, Marcelo, Guilherme, Fernando and Henrique for all the support, ear lending, laughs, coffees, pizzas, snookers, more coffees, and strength. The road would have been much more difficult and boring without you.

To my friend António Espírito Santo, for show me that not even the ocean can separate a good friendship.

To Roberta, for always having faith in me and my work, even when I myself no longer had. Thank you.

To my grandfather João, for still looking and taking care of me, no matter where he is.

To everyone, thank you.

And all that is now
And all that is gone
And all that's to come
And everything under the Sun is in tune
But the Sun is eclipsed by the Moon.
Roger Waters - Eclipse (The Dark Side of the Moon)

Abstract

There are, nowadays, about 110 million people in the world who live with some type of severe motor disability. Specifically in Brazil, about 2.2% of the population are estimated to live with a condition of difficult locomotion. Aiming to help these people, a vast variety of devices, techniques and services are currently being developed. Among those, one of the most complex and challenging techniques is the study and development of Brain-Computer Interfaces (BCIs). BCIs are systems that allow the user to communicate with the external world controlling devices without the use of muscles or peripheral nerves, using only his decoded brain activity. To achieve this, there is a need to develop robust pattern recognition systems, that must be able to detect the user's intention through electroencephalography (EEG) signals and activate the corresponding output with reliable accuracy and within the shortest possible processing time. In this work, different EEG signal processing techniques were studied, and it is presented the development of a EEG under visual stimulation (Steady-State Visual Evoked Potentials - SSVEP) pattern recognition system. Using only Open Source tools and Python programming language, modules to manage datasets, reduce noise, extract features and perform classification of EEG signals were developed, and a comparative study of different techniques was performed, using filter banks and Discrete Wavelet Transforms (DWT) as feature extraction approaches, and the classifiers K-Nearest Neighbors, Multilayer Perceptron and Random Forests. Using DWT approach with Random Forest and Multilayer Perceptron classifiers, high accuracy rates up to 92 % were achieved in deeper decomposition levels. Then, the small-size microcomputer Raspberry Pi was used to perform time processing evaluation, obtaining short processing times for every classifiers. This work is a preliminary study of BCIs at the Laboratório de Instrumentação e Engenharia Biomédica, and, in the future, the system here presented may be part of a complete SSVEP-BCI system.

Keywords: Pattern recognition. Machine learning. Brain-machine interface. Brain-computer interface. Python. Raspberry Pi. Open-source.

Resumo

Existem, atualmente, cerca de 110 milhões de pessoas no mundo que vivem com algum tipo de deficiência motora severa. Especificamente no Brasil, é estimado que cerca de 2.2% da população conviva com alguma condição que dificulte a locomoção. Com o intuito de auxiliar tais pessoas, uma grande variedade de dispositivos, técnicas e serviços são atualmente desenvolvidos. Dentre elas, uma das técnicas mais complexas e desafiadoras é o estudo e o desenvolvimento de Interfaces Cérebro-Computador (ICMs). As ICMs são sistemas que permitem ao usuário comunicar-se com o mundo externo, controlando dispositivos sem o uso de músculos ou nervos periféricos, utilizando apenas sua atividade cerebral decodificada. Para alcançar isso, existe a necessidade de desenvolvimento de sistemas robustos de reconhecimento de padrões, que devem ser capazes de detectar as intenções do usuário através dos sinais de eletroencefalografia (EEG) e ativar a saída correspondente com acurácia confiável e o menor tempo de processamento possível. Nesse trabalho foi realizado um estudo de diferentes técnicas de processamento de sinais de EEG, e o desenvolvimento de um sistema de reconhecimento de padrões de sinais de EEG sob estimulação visual (Potenciais Evocados Visuais de Regime Permanente - PEVRP). Utilizando apenas técnicas de código aberto e a linguagem Python de programação, foram desenvolvidos módulos para realizar o gerenciamento de datasets, redução de ruído, extração de características e classificação de sinais de EEG, e um estudo comparativo de diferentes técnicas foi realizado, utilizando-se bancos de filtros e a Transformada Wavelet Discreta (DWT) como abordagens de extração de características, e os classificadores K-Nearest Neighbors, Perceptron Multicamadas e Random Forests. Utilizando-se a DWT juntamente com Random Forests e Perceptron Multicamadas, altas taxas de acurácia de até 92 % foram obtidas nos níveis mais profundos de decomposição. Então, o computador Raspberry Pi, de pequenas dimensões, foi utilizado para realizar a avaliação do tempo de processamento, obtendo um baixo tempo de processamento para todos os classificadores. Este trabalho é um estudo preliminar em ICMs no Laboratório de Instrumentação e Engenharia Biomédica e, no futuro, pode ser parte de um sistema ICM completo.

Palavras-chave: Reconhecimento de padrões. Aprendizado de máquina. Interface cérebro-máquina. Interface cérebro-computador. Python. Raspberry Pi. Open-source.

List of Figures

Figure 1 – The neuron with its synaptic connections.	18
Figure 2 – The Action Potential.	19
Figure 3 – The human brain and its lobes.	21
Figure 4 – The different levels of invasiveness of EEG, ECoG and Intracortical recording methods.	24
Figure 5 – The different EEG frequency ranges.	25
Figure 6 – The 10-20 system for electrodes placement.	27
Figure 7 – Frequency domain of the EEG signal of an user controlling his alpha rhythm.	30
Figure 8 – The P300 potential.	31
Figure 9 – Frequency domain of EEG under stimulus of light flickering at 7 Hz. . .	33
Figure 10 – The phenomenon of overfitting.	37
Figure 11 – The Cross Validation procedure.	37
Figure 12 – General Structure of a BCI system.	39
Figure 13 – EEG signal and its periodogram.	43
Figure 14 – Filter Bank Example.	44
Figure 15 – Aliasing caused by reducing the sampling frequency.	46
Figure 16 – Block diagram of the downsampling process.	46
Figure 17 – Application of an anti-aliasing filter before the decimation.	47
Figure 18 – level-3 DWT implementation using filter banks.	49
Figure 19 – Frequency bands of each level of DWT decomposition.	49
Figure 20 – Popular mother wavelets.	49
Figure 21 – Example of the KNN problem.	51
Figure 22 – MultiLayer Perceptron.	52
Figure 23 – EEG from Oz channel of Subject 1 under a 5 Hz stimulus.	58
Figure 24 – Scaled EEG from Oz channel of Subject 1 under a 5 Hz stimulus. . . .	59
Figure 25 – Filter bank with first four harmonics of 5Hz as bandpass.	60
Figure 26 – Periodogram for Oz channel of Subject 1 under 5 Hz stimulation. . . .	61
Figure 27 – Periodogram of the filtered signal for Oz channel of Subject 1 under 5 Hz stimulation.	61
Figure 28 – Structure of the filter banks feature extraction module.	62
Figure 29 – db12 Wavelet.	63
Figure 30 – Level 4 DWT decomposition.	64
Figure 31 – Block diagram illustrating a simplified classification process.	65
Figure 32 – The Raspberry Pi Board (2 nd generation).	66
Figure 33 – Block diagram for time evaluation using filter bank approach.	68

Figure 34 – Block diagram for time evaluation using DWT approach.	68
Figure 35 – KNN accuracy versus number of neighbors for filter banks approach. .	70
Figure 36 – MLP accuracy versus number of neurons in hidden layer for filter banks approach.	70
Figure 37 – RF accuracy versus number of trees for filter banks approach.	71
Figure 38 – KNN accuracy versus number of neighbors for level 4 DWT decomposition.	72
Figure 39 – KNN accuracy versus number of neighbors for level 5 DWT decomposition.	73
Figure 40 – KNN accuracy versus number of neighbors for level 6 DWT decomposition.	73
Figure 41 – KNN accuracy versus number of neighbors for level 7 DWT decomposition.	74
Figure 42 – KNN accuracy versus number of neighbors for level 8 DWT decomposition.	74
Figure 43 – MLP accuracy versus number of neurons in hidden layer for level 4 DWT decomposition.	75
Figure 44 – MLP accuracy versus number of neurons in hidden layer for level 5 DWT decomposition.	76
Figure 45 – MLP accuracy versus number of neurons in hidden layer for level 6 DWT decomposition.	76
Figure 46 – MLP accuracy versus number of neurons in hidden layer for level 7 DWT decomposition.	77
Figure 47 – MLP accuracy versus number of neurons in hidden layer for level 8 DWT decomposition.	77
Figure 48 – RF accuracy versus number of trees for level 4 DWT decomposition. .	78
Figure 49 – RF accuracy versus number of trees for level 5 DWT decomposition. .	78
Figure 50 – RF accuracy versus number of trees for level 6 DWT decomposition. .	79
Figure 51 – RF accuracy versus number of trees for level 7 DWT decomposition. .	79
Figure 52 – RF accuracy versus number of trees for level 8 DWT decomposition. .	80

List of Tables

Table 1 – Summary of Brain Activity Measurement Methods.	24
Table 2 – Data Management and Organization Functions	57
Table 3 – Digital filtering Functions	60
Table 4 – Number of features in function of DWT decomposition level.	71
Table 5 – Mean accuracies for different decomposition levels using MLP classifier.	75
Table 6 – Mean accuracies for different decomposition levels using Random Forest classifier.	75
Table 7 – Execution time using filter banks.	80
Table 8 – Execution time using DWT (seconds).	81

Contents

1	INTRODUCTION	14
1.1	Motivation and Objectives	14
1.2	Thesis Outline	16
2	NEUROPHYSIOLOGICAL FOUNDATIONS	17
2.1	Neurological Principles	17
2.1.1	The Neuron	17
2.1.2	The Action Potential	18
2.2	Physiology of the Brain and the Visual Cortex	20
2.3	Characteristics of Brain Activity and its Measurements Techniques	20
2.4	Principles of Electroencephalography	24
2.4.1	Characterization of Brain Patterns	25
2.4.2	EEG Acquisition Techniques	26
3	EEG BASED BRAIN-COMPUTER INTERFACES AND THEIR PRO- CESSING METHODS	28
3.1	Definitions	28
3.2	Acquisition Paradigms	29
3.2.1	Sensorimotor Rhythms	29
3.2.2	P300 evoked potentials	30
3.2.3	Slow Cortical Potentials	31
3.2.4	Steady-State Visually Evoked Potentials	32
4	SSVEP-EEG BASED BCIS SIGNAL PROCESSING TECHNIQUES	35
4.1	Notions of Machine Learning	35
4.2	BCIs as a Pattern Recognition Systems and its General Structure	38
4.3	Signal Acquisition	38
4.4	Preprocessing	40
4.4.1	Digital Filtering	40
4.4.2	Data Standardization	41
4.5	Feature Extraction	41
4.5.1	Periodogram	42
4.5.2	Filter Banks	44
4.5.3	Signal Downsampling	45
4.5.4	Discrete Wavelet Transform	46
4.6	Classification	50

4.6.1	K-Nearest Neighbors	50
4.6.2	Multilayer Perceptron	51
4.6.3	Random Forest Classifier	52
4.7	Activation of External Devices	53
5	DEVELOPMENT OF AN SSVEP-EEG PATTERN RECOGNITION SYSTEM	54
5.1	Open-Source Data Processing Tools: The Python Language for Scientific Computing	54
5.2	Input Data: The Kolodziej SSVEP-EEG Dataset	56
5.3	Preprocessing	58
5.4	Feature Extraction	58
5.4.1	Filter Banks Development	59
5.4.2	DWT Decomposition	62
5.5	Classification	64
5.6	Execution Time Analysis	66
5.6.1	The Raspberry Pi Microcomputer	66
5.6.2	Structure of the Execution Time Evaluation Test	67
6	EXPERIMENTAL RESULTS AND DISCUSSION	69
6.1	Model Accuracy	69
6.1.1	Filter Banks	69
6.1.2	DWT Coefficients	71
6.2	Execution Time	77
6.2.1	Filter Banks	79
6.2.2	DWT coefficients	80
7	CONCLUSION	82
	BIBLIOGRAPHY	84

1 Introduction

This chapter is divided in two sections. The first one presents the motivations and the objectives for the proposed work, and the second presents the thesis outline.

1.1 Motivation and Objectives

According to the World Health Organization, more than a billion people (15% of the world population, based in 2010 estimate) are estimated to live with some kind of disability. In this group, about 110 million people (2.2%) have significant difficulties in functioning or some severe disability, such as quadriplegia, severe depression or blindness (WHO, 2011). Regarding motor disabilities, in Brazil, specifically, the number is of about 13.6 million people (about 7% of the population) who lives with this condition due to physical, mental or sensory damages. These damages can be originated by various factors, like: spinal cord injury (SCI), strokes, amyotrophic lateral sclerosis (ALS), among others (IBGE, 2010; GNANAYUTHAM; BLOOR; COCKTON, 2005).

People affected by motor disabilities lose, normally, the ability to move one or more limbs. For example, patients with spinal cord injury (SCI), lose the ability to move their lower limbs (a condition called paraplegia) and, in some more severe cases, even the upper ones (quadriplegia). This occurs due to the interruption of the neural path control of muscles by the brain. This motion loss leads, over time, to more complex situations, like cardiovascular diseases, bone demineralization, obesity, among others (IBRAHIM; SHERWANI, 2014). Besides that, these injuries make this group of people extremely depending of other's help. Therefore, it is important to develop technologies, resources and services that allow disabled people to increase their functional capacity, leading them to a state of greater autonomy. The assistive technology research field has that idea as its main goal (TANNUS, 2014).

Assistive technologies refers to a vast variety of devices, techniques and services aimed at helping people with all types of disabilities and/or special education/rehabilitation needs. Specifically, it includes multiple resources capable of providing different forms of support, which are expected to produce significant behavioral and social benefits to the users by reducing the negative impact of their disabilities and related conditions (LANCIONI et al., 2012). The development of new types of assistive technologies affects directly the quality of life of disabled people, and it has an ample concept of application, including from simple and mechanic devices such walking sticks, until utilities developed with high technology such robotic prosthesis and electronic wheelchairs (TANNUS, 2014). However, in the case of people with severely reduced mobility, it is hard to develop a

device that fits properly with the user needs, since most of them need some sort of explicit activation command, such a button pressing or a voluntary movement, impossible to achieve by these people. Thus, in the attempt to find alternative means of activation, brain signals began to be explored since the early 1990s (WOLPAW et al., 2000). Since then, the research field of the Brain-Computer Interfaces (BCI) has gained great visibility.

A BCI, sometimes called Brain-Machine Interface (BMI), is a system that allows the user to communicate with the external world controlling devices without the use of muscles or peripheral nerves. The control is performed through the decoding of the individual's intentions by detecting and analyzing his brain signals, that is, establishing a direct connection between the human brain and a computer or another external device, providing, then, an alternative communication channel. Through the use of a BCI, people with severe motor disabilities can interact with external devices such as wheelchairs, prosthesis, monitors, voice synthesizers among others (OIKONOMOU et al., 2016). Moreover, in addition to the development of new technologies which help disabled people, the research of BCIs heavily contributes to the study and understanding of the brain functioning. This raises, even in the distant future, the hope of increasing the quality of life of all people with severely reduced mobility.

Regarding which type of brain activity to use, there are different approaches, which are referred as BCI paradigms. Each of these paradigms uses brain signals acquired by different forms and with different tools. The most used technique is the electroencephalography (EEG), due to its ease of acquisition and low cost when compared to other brain signal measuring techniques (VOLOSYAK et al., 2010). However, the resolution and reliability of the detectable information through EEG is limited due to a vast number of factors, leading to the need of expensive equipments for acquisition, and/or elaborated methods of signal processing to overcome the noise and correctly detect the brain pattern (WOLPAW et al., 2002).

The development of hardware and software systems for BCIs is highly multidisciplinary. It involves concepts of Engineering, as well as Physics, Neuroscience, Biology, Information Theory, among other subjects. This work is focused on the study of one of the most important parts of a BCI system: the processing of brain signals. This step is responsible to performing operations on the acquired brain signal in order to correctly detect the user's intention and select the corresponding output.

In order to obtain accurate results, the signal processing techniques must be developed through a reliable software framework. There are already some well-established platforms to perform these tasks. Unfortunately, most of them are registered under proprietary licenses, and its acquisition demands the payment of unattractive fees. Nevertheless, there are some open-source framework alternatives that are actually free of charges, multiplatform, well-documented and massively used by the scientific computation

community. As a matter of fact, those are the circumstances of Python programming language and its scientific computing modules. In this sense, the main objective of this work is to develop an open source pattern recognition system for brain signals based on the Steady-State Visually Evoked Potentials (SSVEP), through the comparative study of different pattern recognition and signal processing techniques. The system must have accuracy detection rates and execution time both in an acceptable standard. The system which is developed here will be later used in embedded low-cost BCI applications.

1.2 Thesis Outline

For a better understanding of the theme, this work is divided in seven chapters. The neurophysiological foundations to understanding the brain activity are presented in Chapter 2. In Chapter 3, information about BCI systems using specifically EEG systems as input signal are exposed. The main SSVEP signal processing methods are shown in Chapter 4. In Chapter 5 we covered the experimental procedures used in the development of the pattern recognition system. The experimental results of accuracy and execution time are shown in Chapter 6. Finally, in Chapter 7, the main conclusion of this work are presented, as well as some proposed future works.

2 Neurophysiological Foundations

In this chapter, the neurophysiological origins of brain electrical activity will be presented. Concepts about the physiology of the neuron, the central nervous and visual systems will be briefly explained, as well as the principles of EEG signals and its measurement techniques.

2.1 Neurological Principles

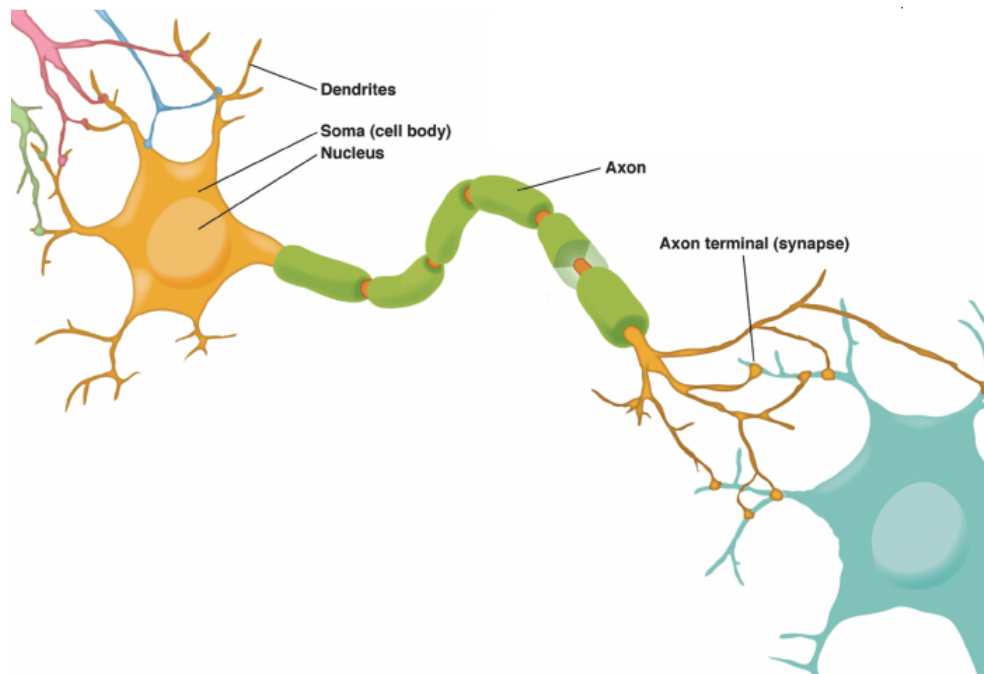
2.1.1 The Neuron

All body tissues are made by cells with specialized functions, and the way they interact determines the functions of human organs. As said by Bear, Connors and Paradiso (2007), The brain is "the most sophisticated and complex organ that nature invented" (BEAR; CONNORS; PARADISO, 2007), and even today the complete understanding of its functioning structure is far from being achieved. The fundamental brain component is called neuron, and it is an elementary node (also called element) of the brain, being fairly more complex than any other cell. The neuron differs from most of the other cells due to its polarized body, and distinct morphological regions, each one with specific functions. Each different behavior of a living being is determined by specific sets of interconnected neurons, and the behavioral function of each neuron is determined by its connections with others (KANDEL et al., 2014). A general structure of a neuron is shown in Figure 1. The dendrites are the neuron parts that receive signals from other neurons. The body or soma contains the nucleus and other biochemical components necessary for cellular function. The information is transmitted from one part of the neuron to next neuron's dendrites through the axon. An important region of the neuron is the synapse, that is located in the terminal region of the axon, where it forms a connection bridge between both neurons. The information transmission process is called *synaptic transmission*, and consists of fundamental electrochemical events called *action potentials* (BYRNE; DAFNY, 1997).

The dendrites receive information from the synapses of other neurons, called pre-synaptic cells. Each neuron can receive information from about 10,000 others. In the same way, one neuron can contact up to 10,000 post-synaptic cells. An illustration of the connection between neuron cells is shown in Figure 1. Adding its capability to perform complex connections to the fact that there is about 100 billion neurons in the human brain, it is easy to state that the potential complexity of biological neural networks is huge. The networks formed have the capacity to receive, process and send information in the form of

electrical impulses at a high transfer rate, allowing the execution of complex and diverse tasks.

Figure 1 – The neuron with its synaptic connections.



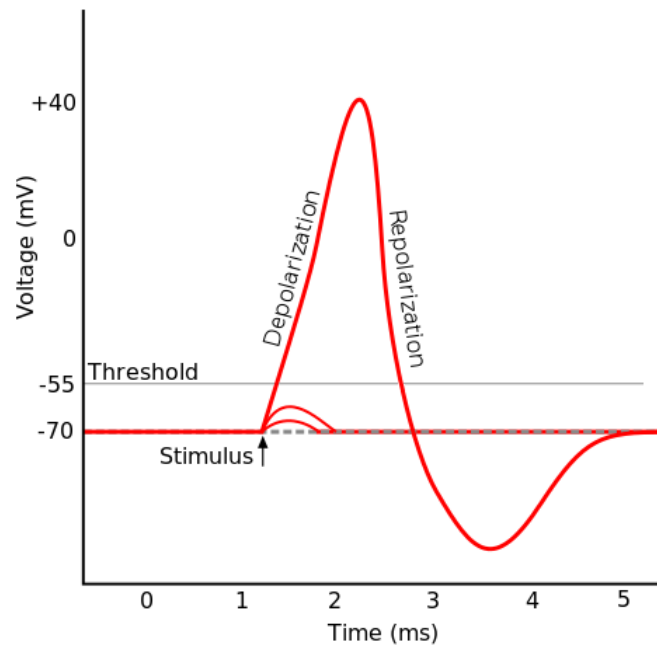
Source: Modified from (BYRNE; DAFNY, 1997).

2.1.2 The Action Potential

Action potentials, also known as *spikes*, are the fundamental units of communication between neurons, and occurs as reactions to external stimuli. The inner and outer sides of a cell membrane exhibit distinct electric potentials, produced by an exchange of ions between both sides. The difference between these two potentials is called membrane potential and has, when in rest, a value around -70 mV . The occurrence of an incoming stimulus reaching the dendrites causes a change in this potential and, if this change reaches a threshold of about -55 mV , there is a inversion of the membrane potential, called action potential (AP). The AP, shown in Figure 2, is seen as a rapid increasing in the membrane potential (depolarization) until $+30\text{ mV}$, followed by an also fast potential fall (repolarization). After this fall, the potential reaches a value below the resting potential, around -90 mV (hyperpolarization), and after a few milliseconds, the membrane returns to its resting potential, characterizing the end of the AP (BYRNE; DAFNY, 1997).

The synaptic transmission generally happens through a chemical path, in which the presence of an electrical impulse in the pre-synaptic axon releases a chemical substance called neurotransmitter. The neurotransmitter allows the impulse to be carried on to the pos-synaptic cell. These electrical impulses transmits an electrical change through

Figure 2 – The Action Potential.



Source: Modified from (GUYTON; HALL, 2006).

the neuron membrane from the point where it started. The information inside neurons is received, processed and sent in the form of action potentials, based on the *neural summation* (BYRNE; DAFNY, 1997). The neural summation is the process that determines when an action potential is generated in neuron axon, as the result of a sum of the action potentials from pre-synaptic cells. Potentials at the pre-synaptic membrane are summed in the cell body. If the depolarization at the axon grows to a value greater than a threshold, then, an action potential occurs and travels along the axon, which is considered as an input in a post-synaptic membrane (GUYTON; HALL, 2006).

The brain receives, processes and passes on information in the form of action potentials, also called *spikes* (CHAPIN et al., 1999). An important functional feature of the brain is that the specificity of information is not defined by the form of the signal, but by the pathway the signal travels in the brain, that is, it is the interpretation of the signal pathways and patterns of pathways that leads to sensation of external stimuli. This feature leads to the fact that the different types of external stimuli are processed by different brain regions, called cortical areas, or *cerebral lobes* (GÜNEYSU; AKIN, 2013; KANDEL et al., 2014).

2.2 Physiology of the Brain and the Visual Cortex

The human brain is the most important organ of the human body. The brain and the Spinal Cord form the Central Nervous System (CNS). Together with the Peripheral Nervous System, which connects the CNS to the limbs and organs, they form the Nervous System, which has fundamental importance in the control of the body functions. All the sensory information is received and processed by the Nervous System (GUYTON; HALL, 2006).

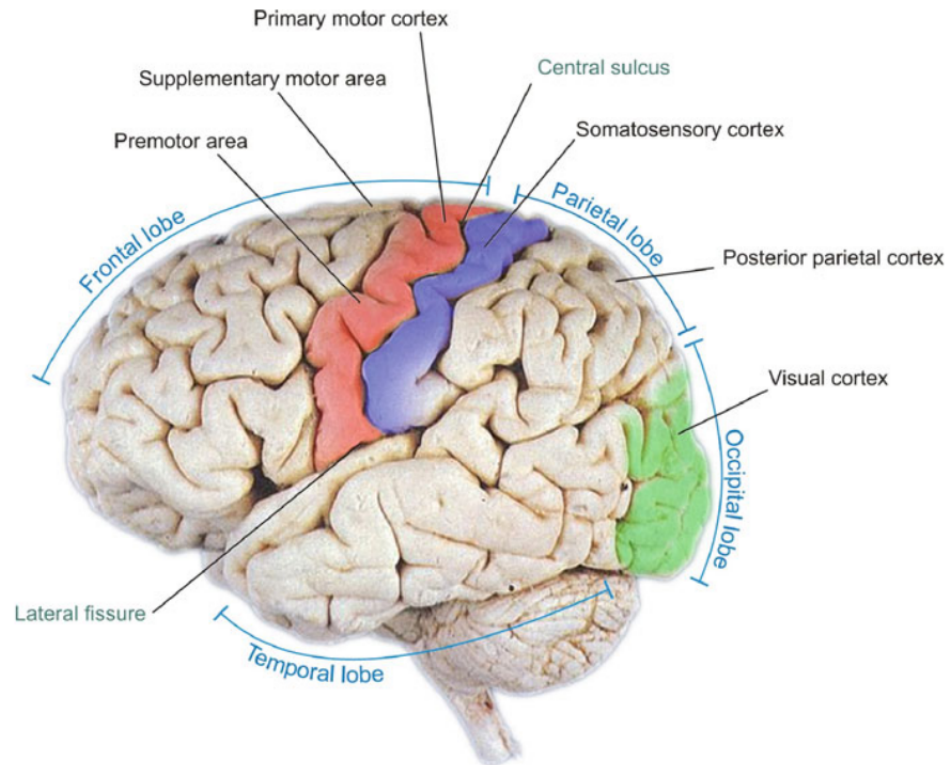
Distinct parts of the brain are specialized in different sensory and behavioral tasks, and all the information processing is done completely and in parallel. As depicted in Figure 3, the cerebral cortex can be divided in four main areas (lobes): the frontal, temporal, parietal and occipital lobes (GUYTON; HALL, 2006; KANDEL et al., 2014). The occipital lobe is located at the posterior region of the brain. Its main function is to process visual information. It has about 30 sub-areas, which are mostly related to processing different visual stimuli, like colors, shape or luminosity. At the front of the brain it is located the frontal lobe, that includes the motor, pre-motor and pre-frontal cortices. While the most part of this lobe plays a large role in voluntary movement, it has other functions like judgment, conscience, personality, planning and empathy. The parietal lobe is located above the occipital and behind the frontal lobe. This lobe is the main processor of sensory information among various modalities, including tactile information, temperature, pain, smells, notions of time and space and most of other environmental information. The parietal lobe also has the function of language processing. Finally, there is the temporal lobe, located above the ears, having the processing of information about memory and perception as its main functions. It is also responsible for auditory perception.

This division of tasks processing by different brain regions is currently widely accepted and used by the scientific community, but it is not strictly exact. Although the sensory informations processing are divided along different brain spatial regions, some of these regions can deal with information from stimuli that are not strictly specified for that lobe. For instance, although the occipital lobe process the great amount of visual stimuli, the temporal lobe also has the ability to process this type of information.

2.3 Characteristics of Brain Activity and its Measurements Techniques

The brain activity generated as a response to external stimuli is manifested in distinct ways and brain areas. Moreover, the same brain signal can have a totally different behavior based on how it was measured. In the same way, some measurement techniques are more complex, expensive or invasive than others. Therefore, the types of brain activity

Figure 3 – The human brain and its lobes.



Source: Modified from (GRAIMANN; ALLISON; PFURTSCHELLER, 2010).

acquisition methods are divided according to: invasiveness, temporal resolution, spatial resolution and electrical or hemodynamic origin.

- Invasiveness: it is the most important feature in the brain signal measuring. Invasiveness consider how deep the sensor penetrates the dura mater (CAGLAYAN, 2014). There are two broad classes of brain measuring technologies - the invasive, in which sensors are implanted directly on the brain surface, or even inside the brain, and the non-invasive, in which external sensors are used to measure brain activity.
- Temporal resolution: it is the smallest time lapse of neural activity that can reliably be observed by the considered method.
- Spatial resolution: it is the smallest neuronal area that can be accessed by the method on focus.
- Origin of activity: the measurement of a brain signal can be done based on its neurophysiological origins, which can be either electrophysiologic or hemodynamic (NICOLAS-ALONSO; GOMEZ-GIL, 2012), as following:
 - Electrophysiologic: the brain functioning is based on electrochemical transmitters, which exchange information between the neurons in the form of ionic

currents. There are some techniques that measure these ionic currents, like electroencephalography (EEG), electrocorticography (ECoG), magnetoencephalography (MEG) and Intracortical neuron recording.

- Hemodynamic: during the neuronal activity of information exchanging, the blood releases glucose and oxygen to the active neurons at a higher rate than the inactive ones. Hence, there is an excess of oxyhemoglobin and an exchange of oxyhemoglobin for deoxyhemoglobin in the active area. This difference can be measured by the use of techniques such as functional magnetic resonance imaging (fMRI) and near-infrared spectroscopy (NIRS).

Invasive technologies provide larger signal-to-noise ratio (SNR) and temporal and spatial resolution than the non invasive ones, allowing, through its use, the ability to measure the activity of small groups of neurons, or even a single neuron. On the other hand, they usually cover smaller brain regions than the non-invasive techniques. Moreover, invasive methods require surgical procedures to implant the sensors, that often lead to adaptation issues or even implant rejection. Hence, such techniques tend to be more complex and expensive than the non-invasive ones (TAN; NIJHOLT, 2010).

Non-invasive techniques are less expensive, more secure, portable and easier to implement than the invasive ones. The use of non-invasive methods allows the development of simpler and more accessible BCIs. They, however, offer lower SNR and worse temporal and spatial resolutions, which leads to the need of better acquisition systems, including software techniques and approaches that can overcome these drawbacks and distinguish patterns of brain activity from the noise.

There are different techniques to measure brain activity, and the most commonly used in BCIs are presented. Table 1 summarizes and give some values to the measuring methods, based on its characteristics (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

- Intracortical Neuron Recording: it measures the electric activity inside the gray matter of the brain, through the use of an array of electrodes surgically implanted. Through this method, is possible to acquire the activity of small groups of neurons, or even one single neuron (CHAPIN et al., 1999). This method provides much higher spatial and temporal resolution than all the others, reducing the need for powerful post-processing software and denoising techniques. However, it is an extremely invasive method and, due to its difficult of implementation, it is also less applied when comparing with other techniques.
- Electrocorticography (ECoG): it is a technique that measures the electrical activity in the cerebral cortex through electrodes placed on the surface of the brain. It is less invasive than the intracortical recording, but it is still considered an invasive

method, requiring a surgical procedure, like a craniotomy, to implant the electrode grid. Although it has worse spatial and temporal resolution than the intracortical recording, those resolutions are still higher than the ones with non-invasive techniques (HAKEN, 2013).

- Functional Magnetic Resonance Imaging (fMRI): it is a non-invasive technique that detects brain signals with hemodynamic origins, that is, changes in blood volume and oxygenation levels in brain activation areas during neural activity. This method is performed through the application of electromagnetic fields using MRI scanners, and it has a much better spatial resolution compared to others. However, fMRI have lower temporal resolution due to the reaction time of body hemodynamic responses, which makes it unsuitable for real time BCI applications (TAN; NIJHOLT, 2010).
- Near Infra-Red Spectroscopy (NIRS): is a non-invasive optical technique to measure brain signals that uses infra-red lasers to detect metabolic changes in neural activity. The emitted light penetrates the skull, and measures the changes in oxyhemoglobin and deoxyhemoglobin rates. This method has temporal resolution of the order of 100 milliseconds, a good value when compared to other non-invasive methods, but its spatial resolution is reduced due to hemodynamic response delay.
- Magnetoencephalography (MEG): is a non-invasive technique based on magnetic induction. The intracellular currents flowing inside the brain during the neural activity generates magnetic fields that can be measured from outside the head. These fields are acquired by the use of scanners based on superconducting devices, which are extremely sensitive to magnetic disturbances produced by neural activity. The advantage of this method is that the magnetic fields are less distorted by the skull and scalp than optic signals from fMRI and electric fields from electroencephalography. Consequently, the spatial resolution provided by this method is higher than the others. However, MEG scanners are expensive and bulky, which makes it seldomly present and almost unusable in BCI devices (TAN; NIJHOLT, 2010).
- Electroencephalography (EEG): is a well-known non-invasive way to acquire brain signals. This method has operation principles similar to the MEG but, in this case, the measured current has electrical origins. EEG consist of the acquiring and recording the instant electrical currents generated by neural activity. This technique was invented and used for the first time by Hans Berger (BRONZINO, 1999) and it is, until today, widely used in detection and characterization of serious diseases, epilepsy, quantification of pharmacological agents, among others. EEG signals are easily recorded by electrodes placed on the scalp. This method offers higher temporal resolution, due to the rapid electric activity from the neurons. However, the impossibility to monitor small groups of neurons using external electrodes

compromises the spatial resolution provided by this method. Furthermore, the electric currents are heavily affected by the skull, scalp and hair impedances and, consequently, the acquired signal is weak, that is, has a low SNR (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

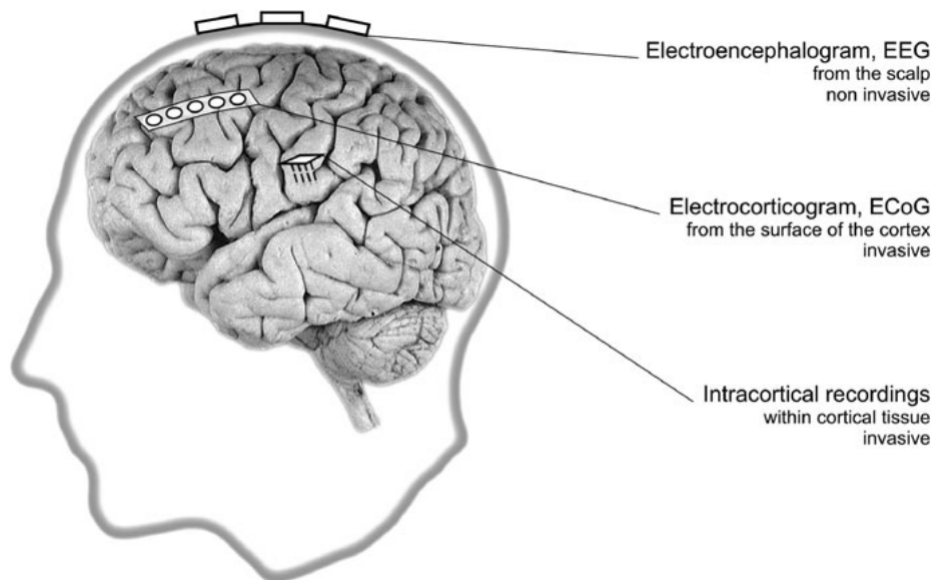
In Figure 4 an illustration of the invasiveness level of the three most used acquisition methods (EEG, ECoG and Intracortical neuron recording) is shown.

Table 1 – Summary of Brain Activity Measurement Methods.

Measurement Method	Type of activity	Temporal Resolution	Spatial Resolution	Risk
EEG	Electrical	0.05s	10mm	Non-Invasive
MEG	Magnetic	0.05s	5mm	Non-Invasive
ECoG	Electrical	0.003s	1mm	Invasive
Intracortical	Electrical	0.003s	0.5mm to 0.05mm	Invasive
fMRI	Metabolic	1s	1mm	Non-Invasive
NIRS	Metabolic	1s	5mm	Non-Invasive

Source: Modified from (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

Figure 4 – The different levels of invasiveness of EEG, ECoG and Intracortical recording methods.



Source:(GRAIMANN; ALLISON; PFURTSCHELLER, 2010).

2.4 Principles of Electroencephalography

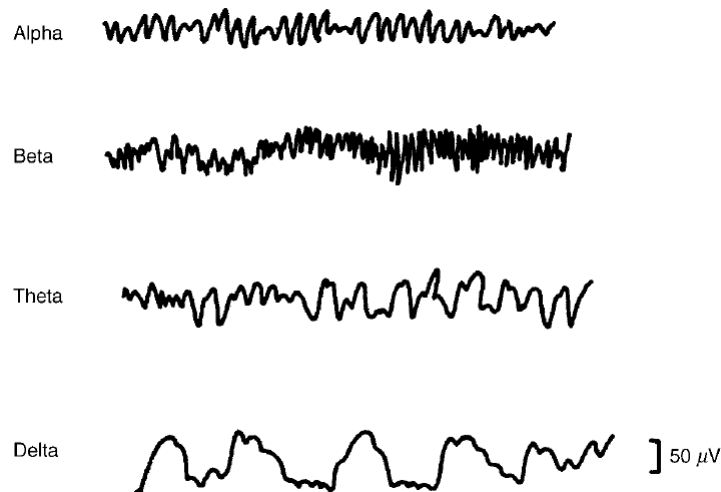
Electroencephalography is the most well-known brain activity measurement technique and it is widely used in BCI systems researches. Due to its low cost, ease of

implementation and non-invasiveness, its patterns have been used in several areas: biochemistry, circulatory system study, hormonal characteristics, neuroelectrical, behavioral, among others. In the past, the EEG interpretation was limited to visual verifications by an extremely qualified human professional. Nowadays, due to the development of signal processing and pattern recognition methods, as well as the evolution in the acquisition and processing hardware systems, the use of computational techniques to denoising and pattern recognition has become common (BRONZINO, 1999). The measured EEG signal consists of spontaneous voltage fluctuations that are present even without any sensory input or external stimulation. It is composed of quasi-sinusoidal rhythms with low amplitude, normally from 10 to 150 μV and, due to its low amplitude, it can almost be mistaken for a stochastic signal or noise. The spontaneous EEG signal (i.e. the EEG signal without any external stimuli) frequency range lies between 0.5 and 50 Hz, and it is categorized in different ranges: Delta, Theta, Alpha and Beta waves (WEBSTER; HENDEE, 2006), to be detailed in the next section.

2.4.1 Characterization of Brain Patterns

The clinic relevance of spontaneous EEG frequency range is commonly divided in different categories, classified based on the state of consciousness. The different EEG bands are shown in Figure 5 (BRONZINO, 1999; KANDEL et al., 2014; WEBSTER; HENDEE, 2006).

Figure 5 – The different EEG frequency ranges.



Source: Modified from (WEBSTER; HENDEE, 2006).

- Delta waves (from 0.1 to 3.5 Hz): with predominant occurrence in the frontal lobe, the delta waves are detected mostly in neonatal and infants. In healthy adults the delta waves can be recorded during sleep stages. The occurrence of a large amount

of delta activity in awake adult subjects may be related to neurological diseases. Due to its low amplitude, lying below $100\ \mu V$, these waves may be confused with artifacts signals.

- Theta waves (from 4 to 7.5 Hz): in a healthy and awake adult subject, the theta rhythm is usually related with states of emotional stress, disappointing, frustration, and sometimes with moments of inspiration and deep meditation. It occurs mainly in the parietal and temporal lobes and, just like Delta waves, has low amplitude, which makes it difficult to measure. Large and persistent amounts of theta activity in an awake adult are abnormal and are caused by several diseases.
- Alpha waves (from 8 to 13 Hz): the alpha rhythm is found predominantly in the occipital lobe and is strictly related with relaxation and relative mental inactivity. Their amplitude increases when the eyes are closed and the body is relaxed. When the eyes are open or some mental effort is made, the alpha activity is suppressed.
- Beta waves (from 14 to 22 Hz, usually): beta waves are found in almost every healthy adult, mostly over the parietal and frontal lobes. They can be related with a state of focus, attention. or even with motor tasks.

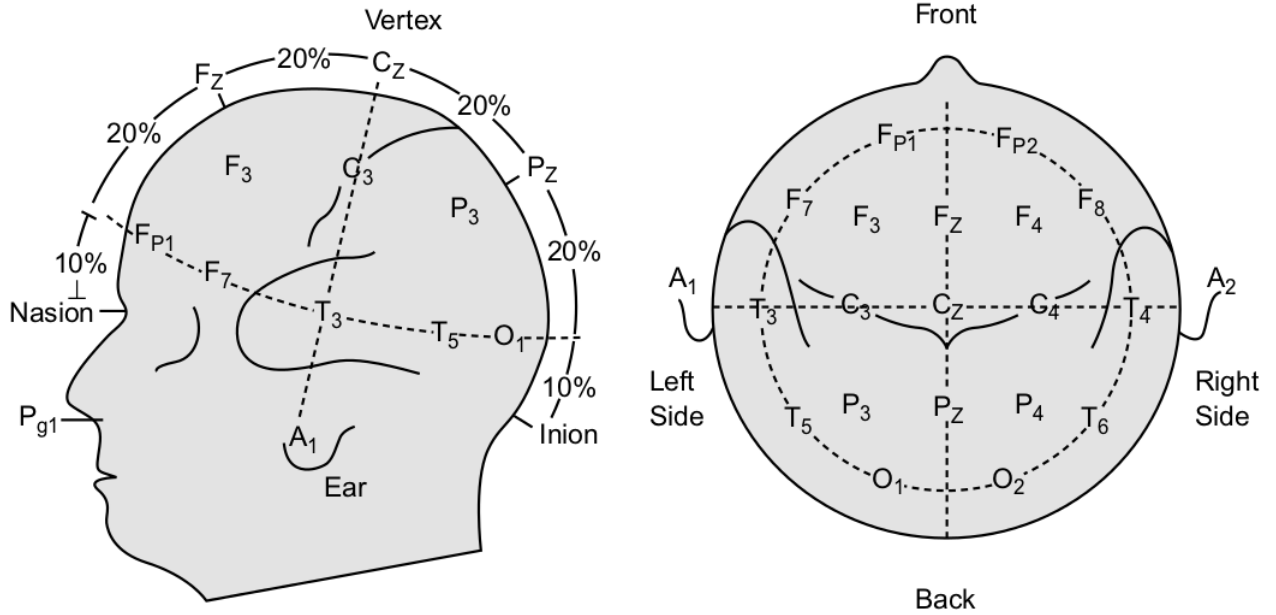
Although the division of the EEG frequency range in subregions contains useful information and is frequently used, that must be assumed with care. There is some evidence of a certain independence of these bands, but the accuracy of where exactly those boundaries are may vary from person to person or even for the same person, based on age or mental state. Similarly with other biomedical signals, there is a large variance among the factors (BRONZINO, 1999; WEBSTER; HENDEE, 2006).

2.4.2 EEG Acquisition Techniques

The EEG signal is acquired from the scalp through the use of electrodes, commonly consisting of disks of non-polarizable metals like Ag-AgCl or gold (WEBSTER; HENDEE, 2006). The signal is measured as the difference of potential between an electrode placement point on the head and a reference electrode. In principle, the electrical potential could be recorded from any point at the head surface, so a standardization of the electrodes placement on the scalp is necessary. There are several some systems for electrode placement, and the most commonly used is the 10-20 system, shown in Figure 6. The letters in the electrodes placement locations correspond to the brain region in which it is located: O for the occipital, P for the parietal, C for the central, F for the frontal and T for the temporal lobes. the letter A represents the ear lobes, where the reference electrodes are placed. This system uses two points of reference in the head: the *nasion*, located at the top of the nose, at the same level as the eyes, and the *inion*, located at the base of the skull. The

electrode placement is determined by marking, at interval of 10% and 20%, the distances starting from these points (TANNUS, 2014; WEBSTER; HENDEE, 2006).

Figure 6 – The 10-20 system for electrodes placement.



Source: (WEBSTER; HENDEE, 2006).

Due to the high sensitivity of EEG to noises and body artifacts, and to its low amplitude, the electronic device responsible to its acquisition should be precisely designed, with high values (at least 100 dB) of Common Mode Rejection Ratio, low input impedances and artifact shielding. Consequently, most of commercial EEG acquisition devices are expensive, which increases the overall cost of a BCI system. Besides that, even when the EEG signal is correctly recorded, there is a need for reliable pattern recognition systems that are able to perform the decoding and classification of the signal with the highest possible accuracy. In this work were used EEG signals previously recorded and made freely available by Kolodziej, Majkowski and Rak (2015) for academic purposes.

3 EEG Based Brain-Computer Interfaces and their Processing Methods

The initial concepts of Brain-Computer Interfaces have already been briefly presented in Section 1.1. In this chapter, these concepts will be presented more deeply, as well as the definitions and the general structure of a BCI system, its main blocks, its acquisition approaches and the most commonly used EEG based signals.

3.1 Definitions

A Brain-Computer Interface is a system that allows the user to control devices without muscle activity. The activation and control are performed by only from the subject's decoded brain activity. In a strict manner, a BCI system measures brain activity, processes it, and sends command signals that reflect the user intentions (GRAIMANN; ALLISON; PFURTSCHELLER, 2010). These activation signals are then driven to an output, such as an electronic prosthesis, a cursor in a monitor or a wheelchair.

There are some features to be met in order to classify a system as a Brain-Computer interface:

1. The activation signals must be measured directly from the brain;
2. A real time feedback must be provided to the user;
3. The user must be able to choose when to execute a task, that is, the control must be fully intentional.

BCI systems can be classified both by the invasiveness of the brain acquisition method and the brain strategies and patterns that generate commands.

Regarding invasiveness, the BCIs are classified by invasive or non-invasive, depending on the method used for brain activity acquisition, which have already been explained in Section 2.3.

Considering the mental strategies, there are some patterns that a BCI system detects and classifies. These patterns are related to a specific event or task, and are generated by the user through the modulation of his brain activity, achieved by the means of a process called mental strategy, or paradigm. The paradigm determines what the user must do to produce patterns in his brain activity that the BCI can detect and classify.

There are several different types of mental strategies, and among those, the main ones are the selective attention and motor imagery approaches (PINTO, 2011).

Regarding BCIs based on selective attention, an external stimulus must be provided by the BCI system so that the user can focus his attention on it. The stimuli can be somatosensory (e.g. temperature) or auditory, but the most used type of stimulus is visual, like flashing lights flickering at distinct and known frequencies. The selective attention by visual stimulus can be achieved through two different BCI paradigms: the P300 potential and the steady-state visual evoked potentials (SSVEP), to be detailed in this work.

In the case of BCIs based in motor imagery, changes in brain activity, that occurs due to the imagination of a limb movement, are used as activation signal. These changes can consist of an increase or decrease of oscillatory activity in a specific frequency band of spontaneous EEG, and are called sensorimotor rhythms (GRAIMANN; ALLISON; PFURTSCHELLER, 2010; GUYTON; HALL, 2006).

3.2 Acquisition Paradigms

Brain activity might serve as control signals and paradigms for BCI systems. In this section, only the most used paradigms for EEG based BCIs will be described: sensorimotor rhythms, P300 evoked potentials, slow cortical potentials and steady-state visual evoked potentials (SSVEP).

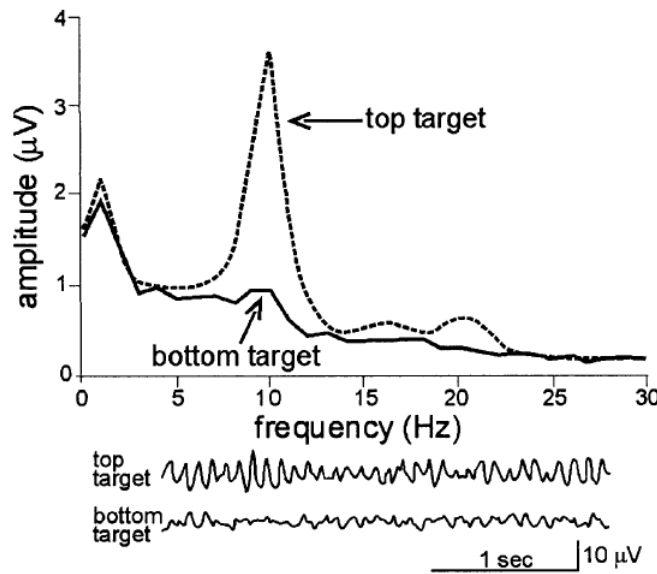
3.2.1 Sensorimotor Rhythms

The sensorimotor rhythms (SMR) are evoked responses based on motor imagery. They comprise mu rhythms and beta activity, with bandwidth between 7 and 13 Hz. The amplitude of the sensorimotor rhythms varies when there is brain activity related to motor tasks (only the imagination of movement is enough to modify the amplitudes). The sensorimotor rhythms consists of two different brain signal occurrences, called event-related desynchronization (ERD) and event-related synchronization (ERS). The occurrence of ERD involves an amplitude suppression in the specified frequencies, and ERS involves an amplitude increase. The frequency domain of the EEG of an user controlling his alpha rhythm, that consists in mu and beta rhythm acquired at sensorimotor cortex, is shown in Figure 7. In this figure, the EEG is acquired from electrodes placed over sensorimotor cortex. The user is trained to control the amplitude of his mu rhythm (8-12 Hz), in order to move a cursor to one of two targets, located at top and bottom of a screen. It is possible to see, through analysis of frequency content of the user's EEG signals, that the control is focused in the mu-rhythm frequency band (WOLPAW et al., 2002).

The sensorimotor rhythms are related to the moment that a movement is performed, or even just imagined, such that even motion disabled people seems to be able to control

this type of activity (KÜBLER et al., 2005; PFURTSCHELLER et al., 2000). Other applications of SMR based BCIs has been documented in the field of stroke neurorehabilitation (CHAUDHARY; BIRBAUMER; RAMOS-MURGUIALDAY, 2016; MURGUIALDAY et al., 2013) and neurofeedback (HUSTER et al., 2014; WOOD et al., 2014). However, it is not easy to learn how to control these rhythms. BCIs using this paradigm generally demands long periods of training (GRAIMANN; ALLISON; PFURTSCHELLER, 2010; NICOLAS-ALONSO; GOMEZ-GIL, 2012).

Figure 7 – Frequency domain of the EEG signal of an user controlling his alpha rhythm.



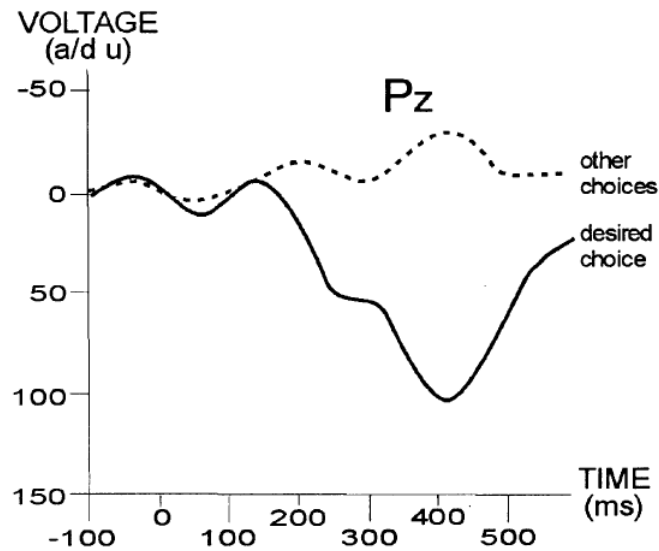
Source: Modified from (WOLPAW et al., 2002).

3.2.2 P300 evoked potentials

The P300 evoked potentials are positive peaks in the EEG when the user experiences an oddball stimulus, that is, these potentials are generated when the user is surprised by a unexpected event. The BCI system using this paradigm was proposed firstly by Farwell and Donchin (1988), where a monitor screen with a matrix of alphabet letters, numbers and symbols was displayed. The user focused his attention on the desired letter, while rows and columns was flickering randomly. When the desired letter blinked a P300 potential was generated, detected by the BCI system and driven to an external device. In Figure 8 it is possible to see an illustration of the P300 phenomenon. In that case, a matrix of possible choices is shown in a screen in front of the user. While user's EEG is acquired from parietal lobe, the choices blink randomly and in succession. Only the desired choice generates a P300 potential, as illustrated by the solid line, characterized by the increasing of potential in Pz electrode about 300 ms after the event. With this acquisition approach, users with extremely reduced mobility were able to communicate

verbally, using voice synthesizers. Several related studies have been conducted since then (DONCHIN; SPENCER; WIJESINGHE, 2000; HOFFMANN et al., 2008; NIJBOER et al., 2008; SELLERS; DONCHIN, 2006; YIN et al., 2013), making spellers and synthesizers the main applications of P300 based BCI.

Figure 8 – The P300 potential.



Source: Modified from (WOLPAW et al., 2002).

The use of the P300 paradigm in BCIs does not require any user's training period. However, due to the low SNR in EEG signals, the detection of targets in just one trial is difficult, and the target must be flashed several times before an output is obtained. Hence, the transmission rate of this type of BCI is low. Most applications show transfer rate of about 25 bits per minute, and latest studies have shown better results, like in Yin et al. (2013), where a transfer rate of 63.56 bits per minute was achieved.

3.2.3 Slow Cortical Potentials

Slow Cortical Potentials (SCP) are very low frequency (below 1Hz) voltage changes in EEG activity, related to the level of cortical activity, where negative and positive changes in the electric potential are directly correlated to increasing and decreasing neuronal activity (NICOLAS-ALONSO; GOMEZ-GIL, 2012). Both, paralyzed and healthy users, can be trained to achieve the ability to voluntarily change their SCP levels, using it to activate external devices. SCP based BCIs were firstly presented by Birbaumer et al. (2000), where a "thought-translation" device, consisting of a screen with several options, symbols and letters was activated by completely paralyzed users through modulation of slow cortical signals. This BCI paradigm is as not widely used as the others, due to the great difficulty in training to self-regulate the SCP.

The ability of modulating SCP is susceptible to many factors like psychological and physical state, motivation, trainer-patient relationship, sleep quality, among others. In some cases, weeks or even months of user's training are required in order to achieve a good capacity in SCP modulation, what makes this paradigm hard to control. Another reason for the difficulty in using SCPs is the low information transfer rate, due to its extremely low frequency behavior (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

3.2.4 Steady-State Visually Evoked Potentials

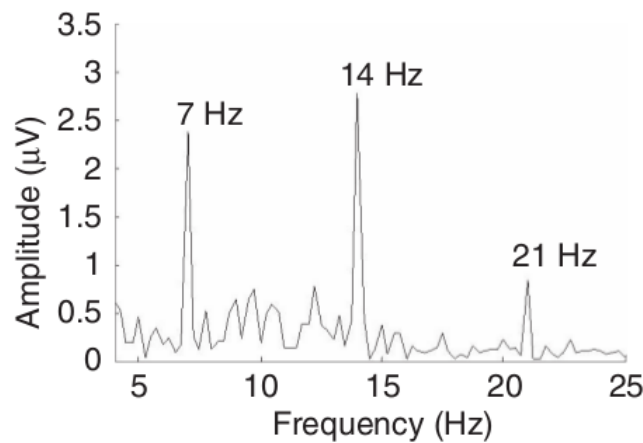
Visual evoked potentials (VEPs) are measurable changes in brain electric potentials, that occur as response to short visual stimuli (CREEL, 2012). This changes are measurable through EEG acquisitions over the visual cortex, and are classified according to two main parameters:

- Morphology of the optical stimuli: the optical stimuli can be performed by flashing lights or by graphic patterns like reversed checkerboards (checkerboards showed on a screen, that changes the check from white to black and vice-versa)
- Stimulus frequency: The visual stimuli can be individual, generating just one response, or continuous, generating consecutive responses. When continuous, it can be classified in two distinct categories depending on the frequency with which the stimuli occurs. In frequencies below 5 Hz, the response generated by the brain ceases before another one can be initiated. In this case, the response is a Transient Visual Evoked Potential (TVEP). On the other hand, when the user is stimulated by frequencies greater than 5 Hz, the last physiological response still exists when the next is generated, resulting in a superposition of evoked potentials. This superposition is characterized by quasi-sinusoidal rhythms synchronized with the stimulation frequency and are called Steady-State Visual Evoked Potentials (SSVEP) (GRAIMANN; ALLISON; PFURTSCHELLER, 2010; PINTO, 2011).

The visual stimuli can be individual, generating just one response, or continuous, generating consecutive responses. When continuous, it can be classified in two distinct categories depending on the frequency with which the stimuli occurs. In frequencies below 5 Hz, the response generated by the brain ceases before another one can be initiated. In this case, the response is a Transient Visual Evoked Potential (TVEP). On the other hand, when the user is stimulated by frequencies greater than 5 Hz, the last physiological response still exists when the next is generated, resulting in a superposition of evoked potentials. This superposition is characterized by quasi-sinusoidal rhythms synchronized with the stimulation frequency and are called Steady-State Visual Evoked Potentials (SSVEP).

The SSVEPs are brain responses that are precisely synchronized with flickering visual stimuli. When an user focuses his gaze on a visual stimulus (a blinking led for example) the responses generated in the occipital lobe have the form of quasi-sinusoidal rhythms at exactly the same frequency of the stimulus. Consequently, frequency components with the same value of the stimulus frequency and its harmonics are detectable through the use of spectral analysis techniques. The intensity of the frequency component depends directly on how focused is the user in that stimulus. Then it is possible to develop SSVEP-based BCIs with various stimuli of different frequencies, each one corresponding to a desired output for the system (GRAIMANN; ALLISON; PFURTSCHELLER, 2010; WANG et al., 2008). In Figure 9, a representation of the EEG signal frequency domain of an SSVEP-based BCI is shown, for an user under stimulus of a led flickering at 7 Hz. It is possible to see, analyzing this figure, the peaks both at stimulus frequency (7 Hz) and its first two harmonics (14 and 21 Hz).

Figure 9 – Frequency domain of EEG under stimulus of light flickering at 7 Hz.



Source: Modified from (WANG et al., 2008).

The advantage of this approach is that the SSVEP response is reliable across different users and usage sections, and it can allow fairly rapid communication. Moreover, this paradigm does not need significant training periods. An user can produce a SSVEP response even without training at all (GRAIMANN; ALLISON; PFURTSCHELLER, 2010). However, the disadvantage of the SSVEP-based BCIs is that the user must keep his gaze fixed on the screen or stimulus device. Consequently, this type of BCI can only be used by patients with control of the eyes and neck movements, being unsuitable for advanced stages of Amyotrophic Lateral Sclerosis (ALS), for example, or other cases of pathologies that cause severe reduced mobility. Besides that, the flickering lights used in this technique can cause fatigue to the user, decreasing the BCI reliability and transfer rate over time. In order to overcome such effects, researches have been made to decrease fatigue or at least delay its onset. In Tello et al. (2015) a study of which colors cause less

fatigue without decreasing the Information Transfer Rate (ITR). In another work, [Diez et al. \(2013\)](#) reached good ITRs using SSVEP devices with higher frequencies (37, 38, 39 and 40 Hz) flickering stimuli, that seemed to be more comfortable to users than the stimulation with low frequencies.

Due to ease of implementation, low cost, relative reliability and no need of training, several applications for SSVEP based BCIs are found in literature. [Sugiarto, Allison and Gräser \(2009\)](#), [Xu et al. \(2013\)](#) and [Chang et al. \(2016\)](#) developed spellers to increase the communication ability of patients with ALS. In these works the BCIs developed are called hybrid BCIs because uses both SSVEP and P300 paradigms. In [Silva \(2007\)](#), [Müller, Bastos-Filho and Sarcinelli-Filho \(2011\)](#) and [Tannus \(2014\)](#), they developed a SSVEP-based BCIs to control automatized wheelchairs. A stimulation apparatus with four LEDs blinking at distinct frequencies was set in front of the user. Depending on which LED the user focuses on, a different command is activated, moving the wheelchair from one to another position. There are other works using SSVEP-based BCIs, like prosthesis control ([MULLER-PUTZ; PFURTSCHELLER, 2008](#)) and rehabilitation by the use of Functional Electrical Stimulation (FES) technique ([GOLLEE et al., 2010](#); [IBRAHIM; SHERWANI, 2014](#)).

In the next chapter, the main signal processing techniques for SSVEP-EEG based BCIs are going to be presented.

4 SSVEP-EEG based BCIs Signal Processing Techniques

The signal processing module is one of the most important parts in a BCI system. It is in this module that the acquired brain signals are processed, decoded and translated into control and communication signals used to activate external devices. In this chapter, the main techniques and methods used to process SSVEP-EEG signals will be presented.

4.1 Notions of Machine Learning

Since the invention of computational systems, people have been trying to make a computer learn. However, a method that accomplishes this task in a similar process to what the human brain performs still does not exist today. The Machine Learning (ML) research field is a subset of Artificial Intelligence (AI) field, and has as central idea the development of systems, processes and algorithms that make computers able to perform tasks without being previously explicitly programmed and, at the same time, improving its performance and experience based on past events (MUNOZ, 2014).

A well-known definition about Machine Learning was stated by Mitchell et al. (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ". The process E is called learning process *or training process*, and its performed using mathematical learning models based on previously recorded data, also called *training data*. If these data are labeled, that is, there is prior knowledge about the desired output of each observation, the models are known as *supervised*. On the other hand, if there is no knowledge about the desired output, they are called *unsupervised* models. Another way to classify a ML model is based on its type of output data. If the output of the model is characterized by continuous data, the ML model is said *regression*. In the case of discrete output values, it is a *classification* problem (WOISKI, 2017).

The ML algorithms are able to predict the result of future observations after a training period learning from the data *features*. Data features are common and measurable characteristics for every instance (observation) in the dataset, that contain useful information about the discriminatory characteristics of the data (WOISKI, 2017; BISHOP, 2006). However, the direct insertion of the observations in the pattern recognition system is not recommended, because this procedure can frequently result in a model with low accuracy and high computational cost, due to the phenomenon know as *curse of dimensionality*. This phenomenon states that the amount of data needed to properly describe the differ-

ent output classes increases exponentially with the dimensionality of the feature vector (BISHOP, 2006; JAIN; DUIN; MAO, 2000). Depending on the amount of features in a input vector, the system can even become computationally infeasible. To overcome this, the idea is to find and extract useful features from the signal that are fast to compute and, at the same time, keep the discriminatory characteristics between different observations. This step is called *feature extraction* (BISHOP, 2006; MITCHELL et al., 1997).

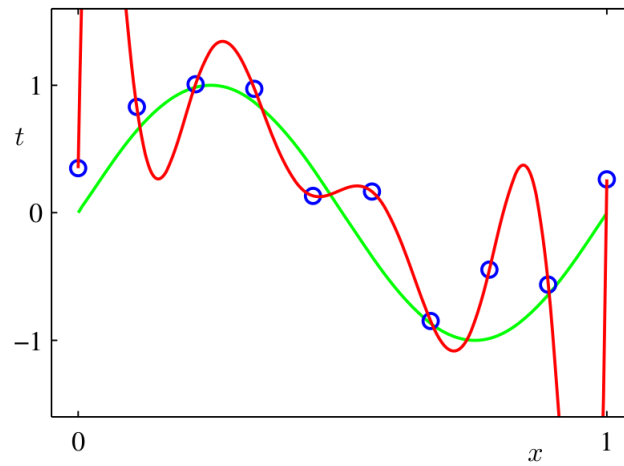
The learning process is the step where the parameters of the model are determined. Each classifier has its main parameters and hyperparameters. Parameters are variables directly related to the mathematical classification model, and whose value can be estimated from data. They are required for the model when making predictions of new data. The hyperparameters, by its turn, are configurations sets external to the model, which change the way the learning process works, and whose value cannot be estimated from data. They are often used in processes to help estimate model parameters, and can often be set using heuristics (THORNTON et al., 2013).

A constant care to be taken when working with ML algorithms is the model generalization. The developed model should be able to adapt, or "fit", to the data which it was trained. At the same time, it must to be able to provide enough generalization to unknown data, avoiding the phenomenon called *overfitting*, a phenomenon that occurs when the model is over-adjusted to the training . In this case, the model adapts so well to the training data that cannot provide generalization to new data. This occurs in models with excessive complexity, or when exists too much features compared to the number of samples (BISHOP, 2006).

As an example, considering the case in which the training process is performed using all the available data, the classification accuracy tests will have high values for known data. However, this model is not generalized, because there is no guarantee that new data will have similar behavior to the training data. In Figure 10 an illustration of this case is shown. In this figure, the green curve represents a true function, the blue circles are measured observations, and the red curve is an overfitted model. It is possible to observe that, although fit perfectly to all the training data, there is no generality in the model, not being a reliable representation to the original function. An overfitted model presents great accuracy when predicting data with same or similar value to the training data. However, it has poor performance when predicting new data.

To overcome this overfitting, a common step when performing data analysis is to split the available data in two parts: train and test sets. The first one is used in model fitting and in parameter adjustment, while the second is used only in the final stage, in order to evaluate the estimator robustness and accuracy. This way, the test set behaves as unknown data, what gives a scenario much more realistic than using all data to train the estimator.

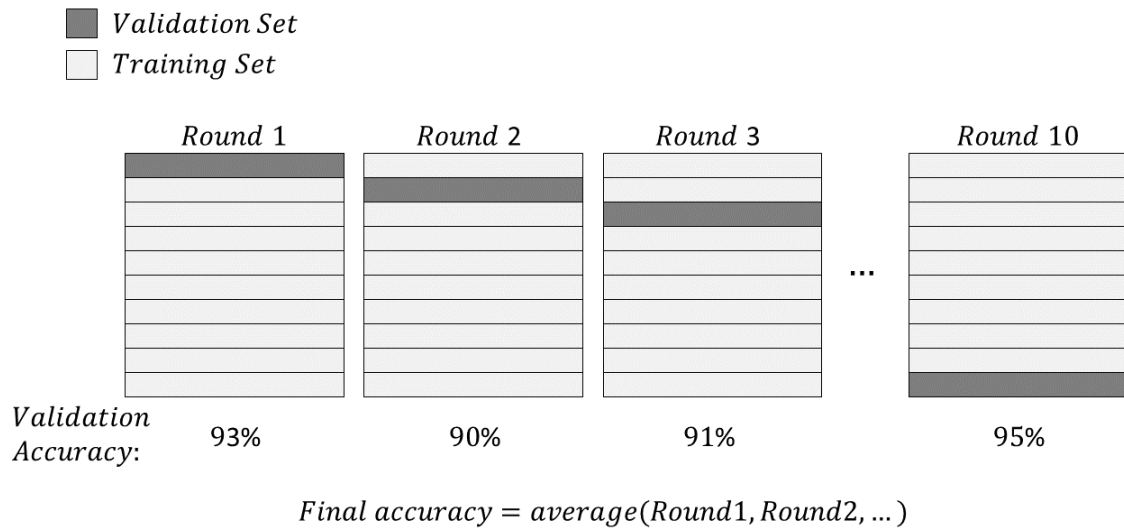
Figure 10 – The phenomenon of overfitting.



Source: Modified from (BISHOP, 2006).

Although randomly splitting the available data into train and test sets is far better than adjusting the estimator with all the data, even so there is a possibility that the data were not randomly separated, also leading the model to overfitting, or increasing the bias. One way to avoid this is to perform a method called *cross validation*.

Figure 11 – The Cross Validation procedure.



Source: From author.

There are different types of Cross Validation, where one of the most well-known is the procedure called K-fold Cross Validation. This technique consists of, after splitting the dataset in train and test sets, split the train set again, now in K parts. The training is then performed using K-1 subsets, and the test is performed using the last subset, called validation set. An illustration of this process is shown in Figure 11. The process is then

repeated until all K subsets have been used as validation set, and the overall accuracy of the model evaluation is given by the mean value among all computed accuracies (KOHAVI et al., 1995).

4.2 BCIs as a Pattern Recognition Systems and its General Structure

The main objective of a BCI system is to decode brain signals, choosing right output based on the input signals. Thus, it is valid to consider a BCI as a pattern recognition system. In this sense, there is approaches based both in regression (FRUITET; MCFARLAND; WOLPAW, 2010) or classification. The use of classification techniques is the most popular approach to BCIs, since generally the outputs choices are finite and well-defined. Specifically in the case of SSVEP-based BCIs, one can observe that it is possible to define this approach as a classification problem, since there is definite output classes (the stimulation frequencies) and a supervised learning process, since there is information about the output classes, that is, the output classes are labeled. In this work, a BCI is considered as a pattern recognition system, and different classification techniques are investigated.

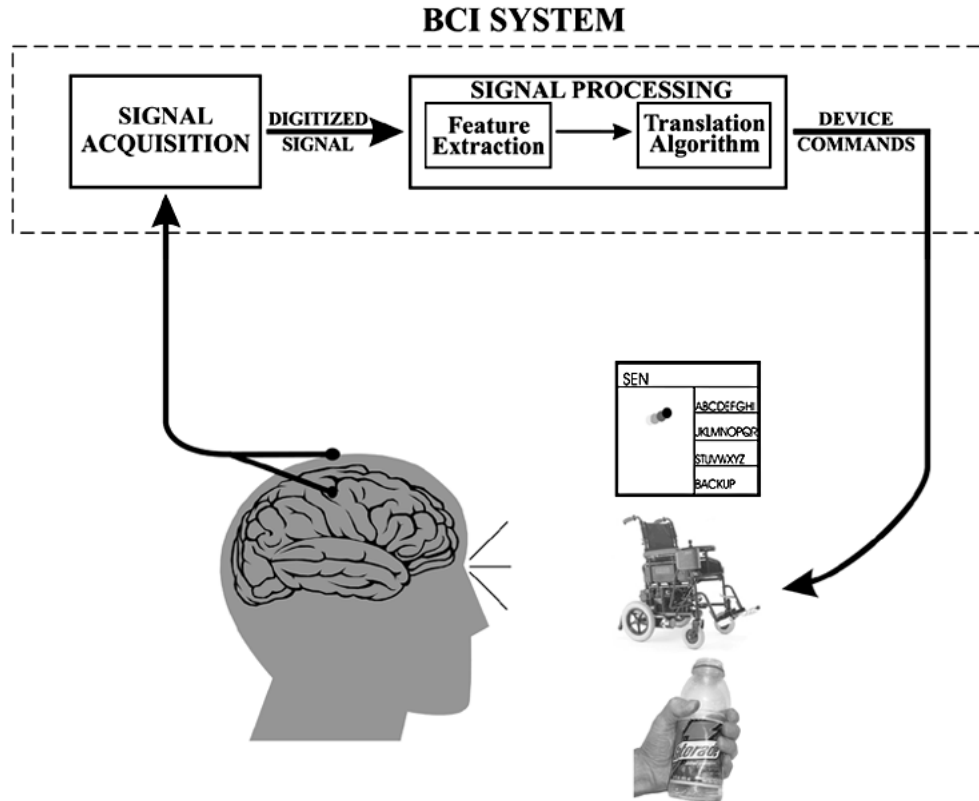
The components of a BCI system may vary, but it follows a general structure, shown in Figure 12. Each of these blocks is of great importance in the correct operation of a BCI system, and they are explained in the following sections.

4.3 Signal Acquisition

The signal acquisition step comprises the devices, procedures and techniques used to generating stimuli, acquire the EEG activity and, when is the case, store it to posterior processing. Specifically, this block comprises:

- Preparing of the electrodes: EEG signals are highly susceptible to external interferences and noise, due to its poor SNR. One of the regions where the signal is more impaired is at the tissue-electrode interface, due to high and variable impedance values present in this region (up to $k\Omega$) (KAPPENMAN; LUCK, 2010), leading to voltage drops and, consequently, to further SNR decrease. In order to fix this issue, this impedance value should kept as low as possible, and this is achieved using a conductive ionic gel before placing the electrodes on the scalp.
- Electrode positioning and acquisition protocols: as presented in Section 3.2, there are different brain patterns used in BCI applications. Depending on which will be used, specific acquisition protocols must be followed. At the same time, the

Figure 12 – General Structure of a BCI system.



Source: (OIKONOMOU et al., 2016)

electrodes must be placed on the region of the scalp where the brain pattern occurs. For example, in the case of sensorimotor rhythms, only an EEG acquisition device is needed, and the electrodes are placed on the pre-motor cortex, at the frontal lobe. In the case of SSVEP stimulation, the electrodes must be placed on the occipital lobe (visual cortex) and, in addition to the EEG device, a visual stimulation system is needed, like a flickering led arrangement or a monitor with changing patterns.

- Biosignal amplifiers and acquisition systems: they are electronic devices dedicated to acquisition of signals from biological sources, that is, the reduction of a continuous-time signal in a sampled discrete-time signal. Bioelectrical signals has some particular characteristics that makes its proper acquisition a non-trivial task: low SNR and amplitudes (between $1 \mu\text{V}$ and 100 mV) (WEBSTER; HENDEE, 2006), high susceptibility to noise and interferences from body and power line, unbalanced output impedances, among others. To overcome these challenges, the acquisition system used for these tasks should have some specific characteristics: high Common Mode Rejection Ratio (CMRR) and input impedances, high gain, different filtering stages including band-pass to separate the desired frequency band, high-pass to suppress low frequency components from body signals, band-reject to suppress the power line

interference, and low-pass anti-aliasing filters (BRONZINO, 1999). Besides that, the design of such systems have also to take to account safety requirements to protect the patient and the equipment, from any hazard caused by electrical discharges. It is also desirable that the acquisition system has reduced weight and volume, to facilitate embedded applications. Due to the high complexity to build such devices, commercial EEG acquisition devices are expensive, making the development of low cost BCI systems even more difficult.

- Data storage: sometimes it is interesting to storage the data after an acquisition. Even when working with real time BCIs, using these data is important in the development and study of EEG signal processing techniques, or in the classification models training/testing process. The prerecorded data can also be made available, so that others researches can work in similar areas. In this case, all precautions relating to organizing, data compressing and protocol documenting should be taken.

4.4 Preprocessing

The objective of the preprocessing step is to increase the SNR of the input signal, preparing it to be inserted in the pattern recognition system. The characteristic of low SNR in EEG signals makes this type of signal to be susceptible to several types of interferences. Some of them come from the body itself, like electromyography (EMG), electrocardiography (ECG) and electrooculography (EOG), that are activities generated by the muscles, hearth and eyes, respectively. Other interferences are related to technical issues, like 50/60 Hz power line interference, cable movements and impedance fluctuations (TEPLAN et al., 2002). In this sense, different techniques are used to minimize the effects of these interferences, increasing the quality of the overall system.

4.4.1 Digital Filtering

A digital filter is a system that performs mathematical operations on a discrete-time signal (generally a sampled continuous-time signal) in order to modify its frequency content. Digital filters implementations have significant advantages over analog filter circuits, such as Ease to build and test, requirement of only simple mathematical operations (addition, subtraction and multiplication) in its computation, and non-susceptibility to parameter variations due to aging, manufacturing changes or temperature and humidity conditions (OPPENHEIM; SCHAFER; BACK, 2015).

Based on its impulse response, digital filters can have finite impulse response (FIR) or infinite impulse response (IIR). In some cases, FIR filters have some advantages over

IIR filters, like inherent stability, no need of feedback, and linear phase. The difference equation of a FIR filter of order N is shown in Equation 4.1.

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_Nx[n-N] = \sum_{i=0}^N b_i x[n-i] \quad (4.1)$$

where b_i is the value of the impulse response of at each instant i for $0 \leq i \leq N$, also known as filter coefficients. Using digital filters, it is possible to decrease the influence of undesired frequency bands containing interferences.

Section 5.4 will shown that, besides preprocessing, it is possible to design digital filters to extract useful features from the EEG signals. As explained in Section 3.2.4, the SSVEP stimulation is well localized in frequency, that is, there is an increase in the spectral density amplitude at the value corresponding to the stimulation frequencies and its harmonics. In this sense, it is possible to design filters with passband at the frequencies of interest, and extract the occurrence of a SSVEP in that frequencies.

4.4.2 Data Standardization

The standardization (sometimes called normalization) of data sets is a well known statistical technique used to approach the data of a standard normal distribution, that is, a Gaussian distribution with zero mean and unit variance. The process is normally performed through subtracting the mean value and of each element of the dataset, and then dividing it by the standard deviation (MILTON; ARNOLD, 2002). This procedure is often required in machine learning approaches, because some types of classifiers have a inaccurate behavior when the data is not in the form of a standard normal distribution. In this case, if the input data of the classifier has great variance, the classifier will also have it (BISHOP, 2006).

Another reason to perform data standardization, when dealing specifically with EEG signals, is the low amplitude of the data values, in the order of microvolts. The standardization in this case is recommended in order to increase the quality of data visualization, and avoid precision errors caused by truncation of the values.

4.5 Feature Extraction

As explained in Section 4.1, it is not recommended to insert input data directly in the pattern recognition system, in order to avoid the curse of dimensionality. In the case of a time series like a raw EEG signal, each instant of time that the signal was recorded is taken as a feature. Hence, the dimensionality of the feature vector increases as the duration of the signal or the sampling frequency increase. Therefore, a more compact representation of the signal is needed, and this is one of the most challenging tasks in the

BCI systems, and in the development of pattern recognition systems in general: find a reliable and compact representation for the input signals. This is achieved by extracting features from the data.

The feature extraction consists of a transformation of variables from a dimensional space to another, where the variables have more significance. The input signals are then redefined, making the task easier to be solved. The resulting output of a feature extraction block is called *feature vector*, and it is the data used to perform the classification (BISHOP, 2006; ALPAYDIN, 2014). In this sense, choosing the correct set of features for a signal is an important step, since the correct execution of this process makes the overall system more reliable, due to the greater significance of the chosen features. At the same time, as the feature vector generally has a smaller dimension than the original input vector, the pattern recognition system becomes more computationally efficient.

There are several feature extraction techniques applied to SSVEP based BCIs. In the case of SSVEP signals, the feature vector should have relevant information on whether are stimulation and from which of the frequencies. As shown in Section 3.2.4, a useful characteristic of SSVEP signals is the synchronization of the EEG components with the stimulation frequencies. In this sense, it is valid to choose techniques based on spectral analysis to extract useful features, that consists of the estimation of how the signal power is distributed over the frequency domain (OPPENHEIM; SCHAFER; BACK, 2015). In this work we explored the techniques of the periodogram and filtering with FIR filter banks to estimate the power spectral density. A signal undersampling was performed on the signal, aiming to reduce the dimensionality of its feature vector. Also, the results were compared to another technique, the Discrete Wavelet Transform, used to feature extraction and dimensionality reduction.

4.5.1 Periodogram

The periodogram is one of the simplest, most known and with less costly methods to estimate the Power Spectral Density (PSD) of a signal. It is classified as a non-parametric method, since it is assumed there is no prior knowledge of the signal to be analyzed.

The Discrete Fourier Transform (DFT) gives the spectral content of a signal, in amplitude and phase. Being $x[n] = x[0], x[1], \dots, x[n - N]$ a sequence of finite length N , its DFT is given by equation 4.2.

$$X(f_k) = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi f_k n) \quad (4.2)$$

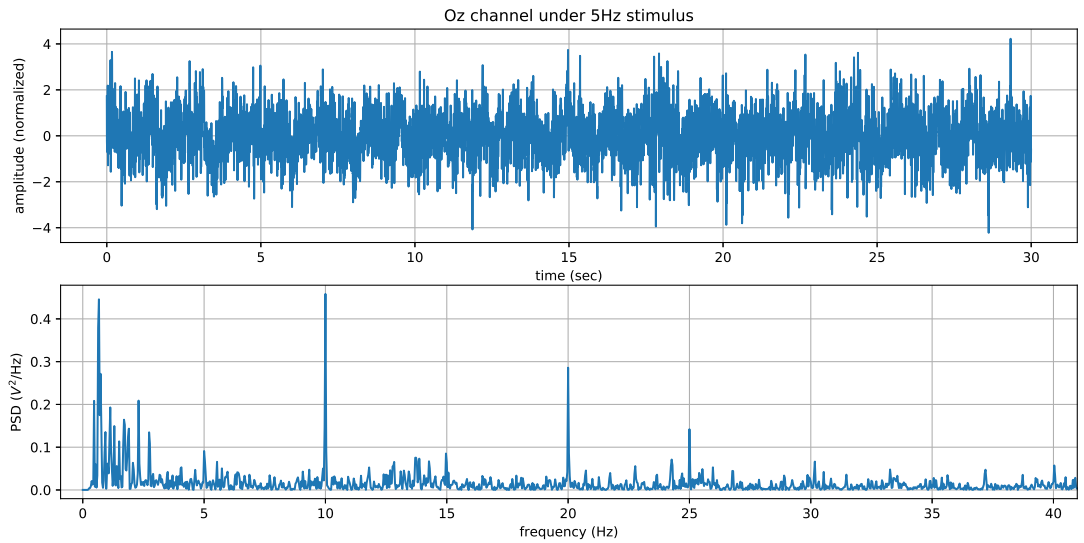
where $f_k = k/N, k = 0, \dots, N - 1$. The magnitude of the DFT is written as:

$$|X(f_k)| = |X(\frac{k}{N})| = |X(k)| = \left| \sum_{n=0}^{N-1} x[n] \exp\left(\frac{-j2\pi nk}{N}\right) \right| \quad (4.3)$$

However, the DFT method ignores the random noise inherent to the acquisition process. The periodogram, by its time, consider the signal as a stochastic process, taking in account statistical characteristics of the signal that are ignored in DFT. Hence, the periodogram becomes a statistically more consistent method than the raw DFT. There are several ways to perform a PSD estimation, where the most known is given by equation 4.4 (OPPENHEIM; SCHAFER; BACK, 2015). In Figure 13 it is possible to see a segment of 30 seconds of EEG recording from Oz channel of a subject under stimulation of a light flickering at 5 Hz and its respective periodogram obtained using hamming window. In the upper graph is shown the signal as it was recorded, in time domain. As can be seen, it is impossible to extract useful information from it. Obtaining the periodogram of this segment of signal, as shown in the lower graph, relevant information about the stimulation frequency and its harmonics can be extracted.

$$P(f_k) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] \exp(-j2\pi f_k n) \right|^2 \quad (4.4)$$

Figure 13 – EEG signal and its periodogram.



Source: By author.

The relationship between the periodogram and the DFT is given by the equation 4.5.

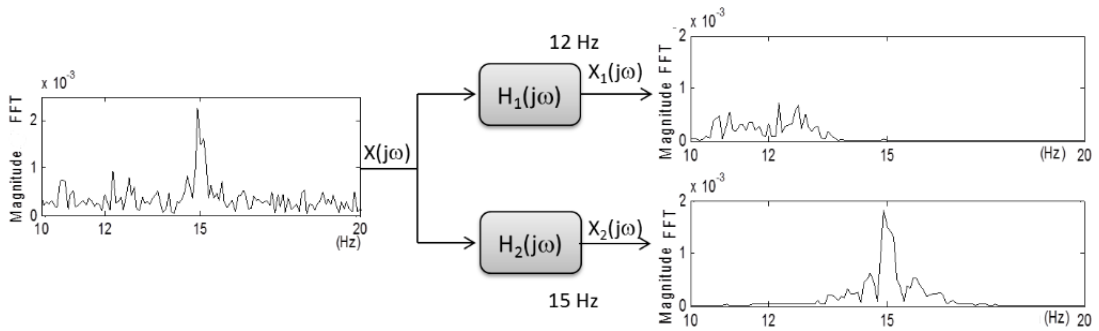
$$P(f_k) = \frac{1}{N} |X(f_k)|^2 \quad (4.5)$$

The periodogram is a simple technique that gives a reliable representation of the signal power distribution over the frequency. However, this method present great variance between its coefficients, that do not decrease independently of the amount of data (OPPENHEIM; SCHAFER; BACK, 2015; OIKONOMOU et al., 2016). Furthermore, this method is susceptible to issues related to spectral leakage and frequency resolution, due to finite length of the sequence, what makes it not the most recommended method when working with non-stationary signals, as is the case of SSVEP-EEG signals (OIKONOMOU et al., 2016).

4.5.2 Filter Banks

Filter banks consist of sets of filters organized in such way the input signal, being inserted in these filters, is divided in frequency components. As the SSVEP occurs in the region of stimulus frequency, a proper arrangement of filter banks could isolate the interest signal. As an example, given by Leite (2016) and presented in Figure 14, there is a subject under 15 Hz stimulus. A BCI capable of distinguish between stimuli at the values of 12 Hz and 15 Hz, and it is intended to detect if he is under stimulus of one of these frequencies. His brain activity is acquired and the EEG channel of interest is, then, applied to both filters. The filters attenuate frequency components outside its cutoff frequencies and, after the filtering, one can observe, through each filter output, that the SSVEP occurrence pass through the 15 Hz filter, what do not occurs with the 12 Hz filter. In this way, it is possible to discriminate one frequency from another.

Figure 14 – Filter Bank Example.



Source: Modified from (LEITE, 2016).

4.5.3 Signal Downsampling

In signal processing, downsampling a sequence $x[n]$ consists of reducing the sampling rate in which this sequence was generated. The sampling rate (Ω_s) is given by $1/T$, where T is the sampling period, that is, the time interval in which each sample is acquired. When a signal is downsampled at a integer factor M , it generates an approximation of what the original sequence would be if it were sampled with a sampling frequency of M/Ω_s , that is, with a sampling period $T' = MT$ (OPPENHEIM; SCHAFER; BACK, 2015).

When working with machine learning and time series, the signal features can be chosen as the instants of time at which the signal was sampled. As said in Section 4.5, increasing the number of features lead to the need of obtaining a number exponentially larger of samples, due to the curse of dimensionality. In this sense, in order to decrease the number of features, a signal downsampling can be performed (ABRAHAM et al., 2014; ESLING; AGON, 2012).

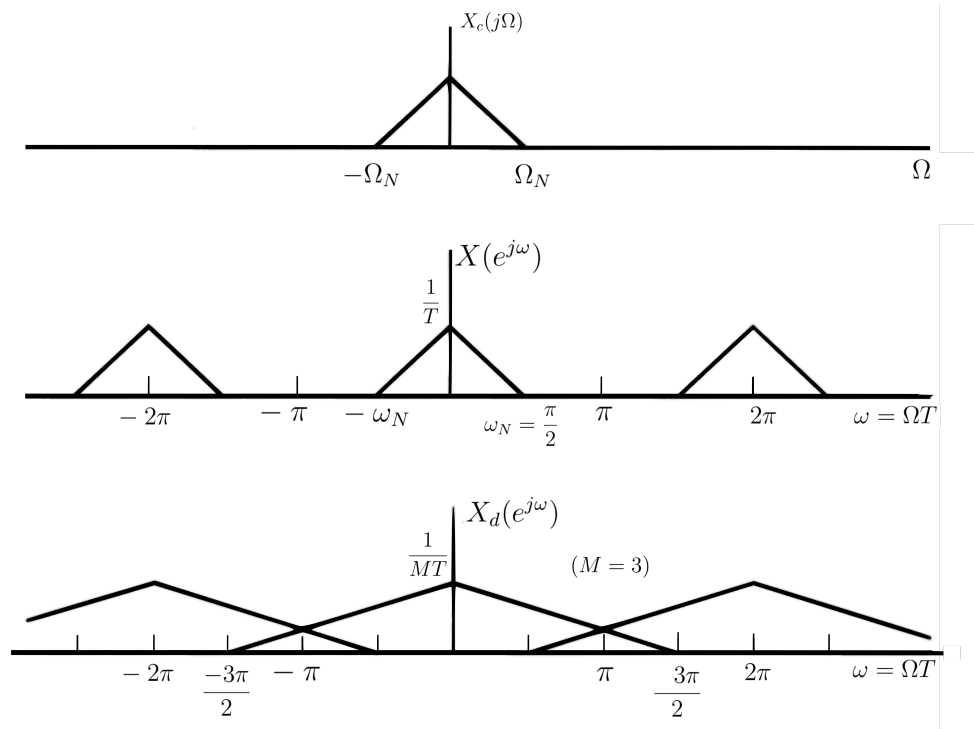
In practice, downsampling a signal consists of discarding part of its samples in an organized way. To reduce the sampling frequency by an integer factor M , for example, it is necessary to take one sample at each M . However, simply throw samples away makes the spectra of the sampled signal be overlapped, and high frequency components are misinterpreted as low frequency signals. This phenomenon is known as *aliasing*. To avoid this issue, the sampling rate must follow the *Nyquist criterion*, that states that the signal must be sampled at a rate greater than or equal to $2\Omega_N$, where Ω_N is the Nyquist frequency, also known as the greatest frequency component present in a continuous-time signal. The aliasing phenomenon is illustrated in Figure 15. In that figure, $X_c(j\Omega)$ is the frequency content of a continuous-time signal $x[n]$, $X(e^{j\omega})$ is its discrete-time frequency content, Ω_N is its highest analog frequency component (known as *Nyquist's Frequency*), ω_N is its Nyquist discrete-time frequency, T is the sampling period and M is the downsampling factor (in the case of this example, 3). It can be seen, in the lower graph of Figure 15 that, after performing the downsampling, the signal spectrum stretches, causing the interference among the signal's high frequency contents.

To avoid aliasing occurrence, the process of downsampling a sequence is performed in two steps, elucidated in the block diagram of Figure 16:

In Figure 17 are shown graphs illustrating this process. In this case, $H_d(e^{j\omega})$ is the frequency response of the anti-aliasing low-pass filter, shown in the upper graph, $\tilde{X}(e^{j\omega})$ is the frequency response of the sequence after being filtered, and $\tilde{X}_d(e^{j\omega})$ is the frequency response of the downsampled sequence.

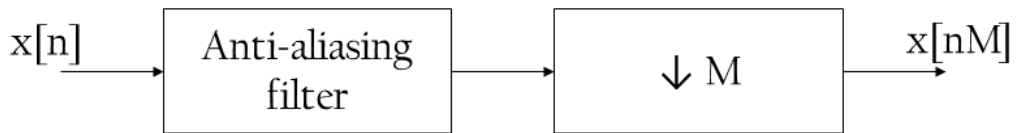
It is noteworthy that, although applying a low-pass filter prevent aliasing, frequency contents present in the original sequence are loss. In fact, the main drawback of this method is that, by discarding samples with no prior knowledge, there is a possibility to

Figure 15 – Aliasing caused by reducing the sampling frequency.



Source: Modified from (OPPENHEIM; SCHAFFER; BACK, 2015).

Figure 16 – Block diagram of the downsampling process.



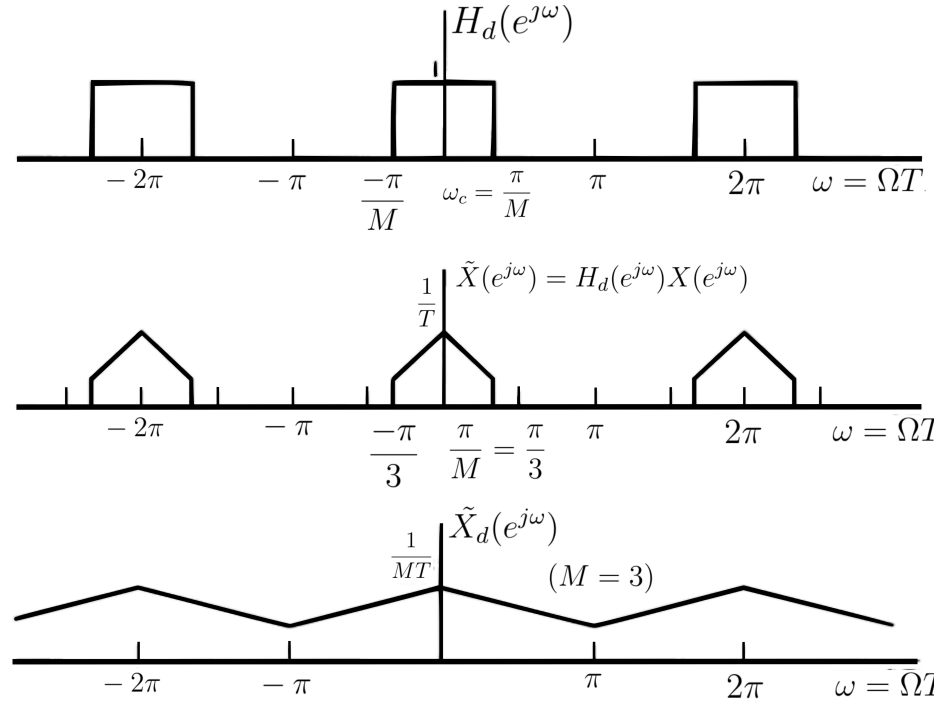
Source: Modified from (OPPENHEIM; SCHAFFER; BACK, 2015).

throw away samples with significant importance to signal analysis. To overcome this problem, in this work was studied another technique to perform the signal decimation and feature extraction: the Discrete Wavelet Transform.

4.5.4 Discrete Wavelet Transform

The spectral methods described until now have the Fourier Transform (FT) as key tool. The general idea of the Fourier Transform is to represent the signal as a superposition of complex exponentials, that is, real cosines and complex sines. Although well represented in frequency, sines and cosines are not well located in time, that is, there is no information about time in a sinusoidal wave representation. Hence, using the FT causes the loss of the information related to time events in that signal.

Figure 17 – Application of an anti-aliasing filter before the decimation.



Source: Modified from (OPPENHEIM; SCHAFER; BACK, 2015).

One way to increase time resolution when working with the FT is to decrease the window length of the signal. Applying the FT in several segments of the signal, with short duration windows, allows an approximated representation of how the frequency content of the signal changes over time. However, this method is affected by issues related to window length and spectral leakage (OPPENHEIM; SCHAFER; BACK, 2015). There is always a trade-off between time and frequency resolution. This is called *Principle of Uncertainty* (GAO; YAN, 2011).

One of the most used tools to overcome this trade-off is the Wavelet Transform (WT). The WT is widely used in many fields of science and engineering, and allows the time-frequency representation through the applications of windows with variable time length. Windows with bigger lengths are used to detect low frequencies contents of the signal, and windows with smaller lengths, to detect higher frequencies. This results in a representation with good resolution both in time and frequency (OCAK, 2009; OIKONOMOU et al., 2016; GAO; YAN, 2011).

The Wavelet Transform is an interesting technique to extract features of a SSVEP-EEG signal and has been used in analysis of irregular or non-stationary signals, with rapid or sharp events, that would be undetectable through the Fourier Transform (KOŁODZIEJ; MAJKOWSKI; RAK, 2015; OIKONOMOU et al., 2016; LIU et al., 2014; DIEZ et al.,

2011). The Continuous Wavelet Transform (CWT) is given by equation 4.6.

$$CWT(a, b) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) dt \quad (4.6)$$

where a and b are called scaling and shifting parameter, respectively, and $\psi(t)$ is called *mother wavelet*, that consists of the window to be applied.

Computationally, the CWT is extremely expensive, due to the need to calculate all of the possible coefficients. To overcome this issue, an alternative way to obtain the coefficients was proposed by Mallat (1989), that consists of calculating the coefficients in a dyadic form, that is, with base 2. This method makes the computation much more efficient. In this case, a and b are replaced by 2^j and $2^j k$, respectively. This method is called Discrete Wavelet Transform (DWT), and its computation is given by the equation 4.7.

$$DWT(j, k) = \frac{1}{\sqrt{|2^j|}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-2^j k}{2^j}\right) dt \quad (4.7)$$

In practice, the DWT implementation is performed by filter banks, as shown in Figure 18 through a representation of a level-3 decomposition using DWT. In the first level, the signal $x[n]$, with Nyquist frequency f_N and sampled at a sampling rate f_s is convolved by a low-pass (LP) and by a high-pass (HP) filter. The output of these convolutions gives the lower and higher frequencies representations of the signal, called Approximation (A_1) and Details (D_1), respectively. The LP and HP filters are generally quadrature mirror filters, that is, their frequency response magnitudes are related by the equation 4.8 (IVANCEVIC; IVANCEVIC, 2010).

$$|H_{LP}(e^{j\Omega})| = |H_{HP}(e^{j(\pi-\Omega)})| \quad (4.8)$$

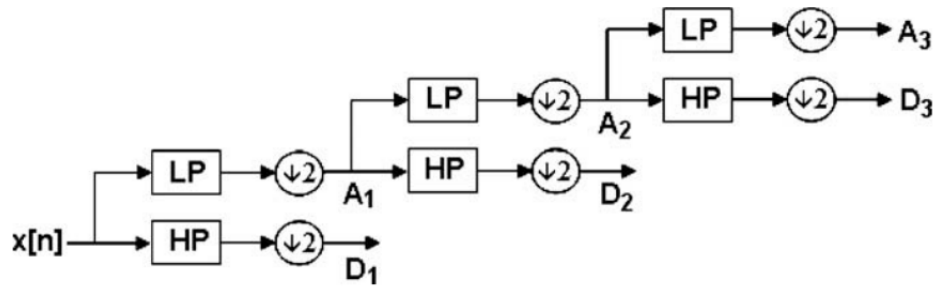
being Ω the continuous-time frequency. Their frequency response magnitudes are mirrored at the point $\Omega = \pi/2$. Therefore, with the same coefficients, its behavior is complementary.

After passing by the filters HP and LP, half of frequencies of the signal are removed. Thus, it is possible to remove half of the samples according to Nyquist's criterion (IVANCEVIC; IVANCEVIC, 2010), and the process is then repeated until the desired decomposition level.

At each decomposition, time resolution is decreased by 2, because the number of samples is half than in the previous step. However, as the frequency band is half than the previous, as shown in Figure 19, the frequency resolution is doubled (FUGAL, 2009).

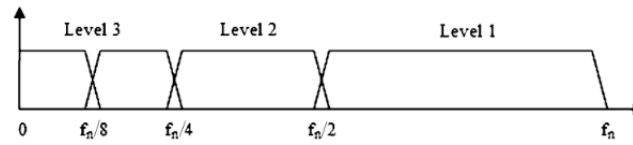
The HP and LP filters are composed by the selected mother wavelet, also called wavelet filter. There is a huge number of mother wavelets classes, and the choice is based on the behavior of the event to be detected (FUGAL, 2009). The study and development

Figure 18 – level-3 DWT implementation using filter banks.



Source: (OCAK, 2009)

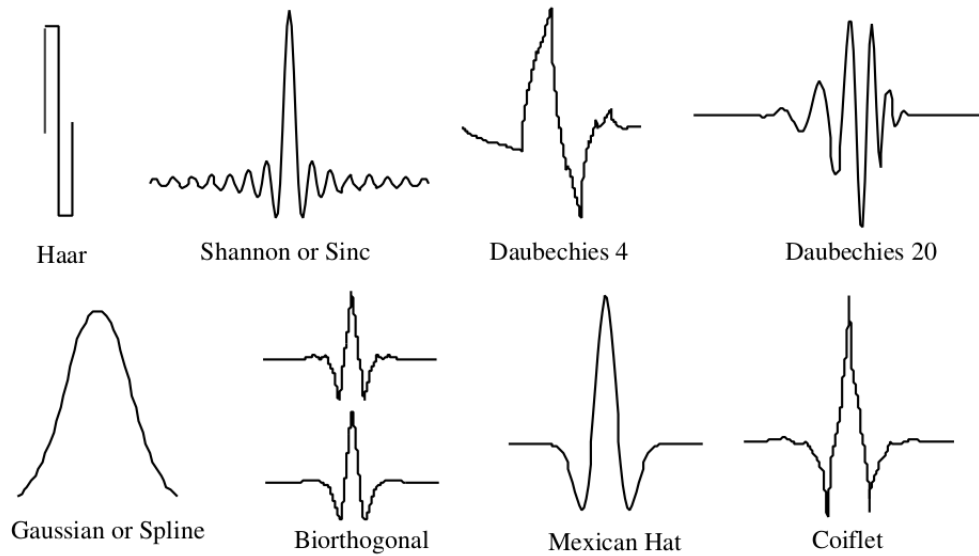
Figure 19 – Frequency bands of each level of DWT decomposition.



Source: (IVANCEVIC; IVANCEVIC, 2010)

of mother wavelets is an active research field, and some of the most used are shown in Figure 20.

Figure 20 – Popular mother wavelets.



Source: (FUGAL, 2009)

4.6 Classification

The final signal processing block of a BCI system is the classification. This stage consists of the pattern recognition system itself, and it is responsible to build assimilation models to all the possible output classes, that is, all the commands that the BCI will perform. The model is built based on the input data, and it must be able to correctly classify the data to which it was trained, as well as data never seen before (OIKONOMOU et al., 2016). Three types of classifiers were tested in this work, each one of them being widely used in SSVEP based BCIs.

4.6.1 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) is one of the simplest and most popular classifiers in the machine learning field. It was firstly proposed by Fukunaga and Narendra (1975), and its central idea is to determine the label, that is, the class of a sample based on samples coming from a test set that are in their neighborhood. This classifier has three main characteristics:

- Instance based: the modeling is done using instances, that is, rows of the feature vector. All the information about the training data are stored in the classifier;
- Competitive Learning: this classifier uses competition between instances to determine the output class;
- Lazy Learning: a model is not built until a prevision is required.

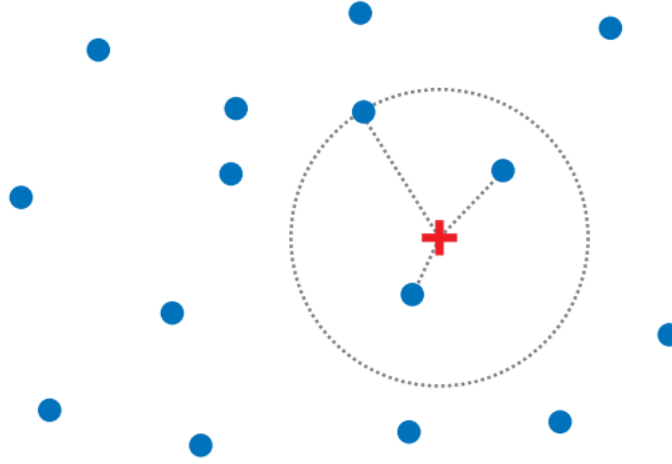
The KNN problem is stated as follows (GARCIA; DEBREUVE; BARLAUD, 2008): let $R = \{r_1, r_2, \dots, r_m\}$ a set of reference points (train data) in a dimensional space with dimension d . Let $Q = \{q_1, q_2, \dots, q_n\}$ a set of query points (test data) in the same space. The KNN algorithm consists of search the K nearest neighbors of each point $q_i \in Q$ in the reference set R given a specific distance. Generally, the distance used is the Euclidean, given by the equation 4.9.

$$D_{euclidean} = \left[\sum_{i=1}^m (r_i - q_i)^2 \right]^{\frac{1}{2}} \quad (4.9)$$

An example of the KNN problem is shown in Figure 21. In this case, $K = 3$ and the space has two dimensions, that is, the problem has two features.

Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems. It is a robust method and it is indicated when working with noisy data. Besides that, the greater is the amount of data, the effectiveness

Figure 21 – Example of the KNN problem.



Source: (GARCIA; DEBREUVE; BARLAUD, 2008)

increases (PEDREGOSA et al., 2011). However, some drawbacks of this classifier lies in its computational cost, that can even become unpractical when working with large datasets. Another detail is that the classification is extremely dependent of its neighbors and the training set. This method is often used to SSVEP detection due its simplicity and robustness (ISCAN; DOKUR; DEMIRALP, 2011; KWAK; MÜLLER; LEE, 2015; OIKONOMOU et al., 2016).

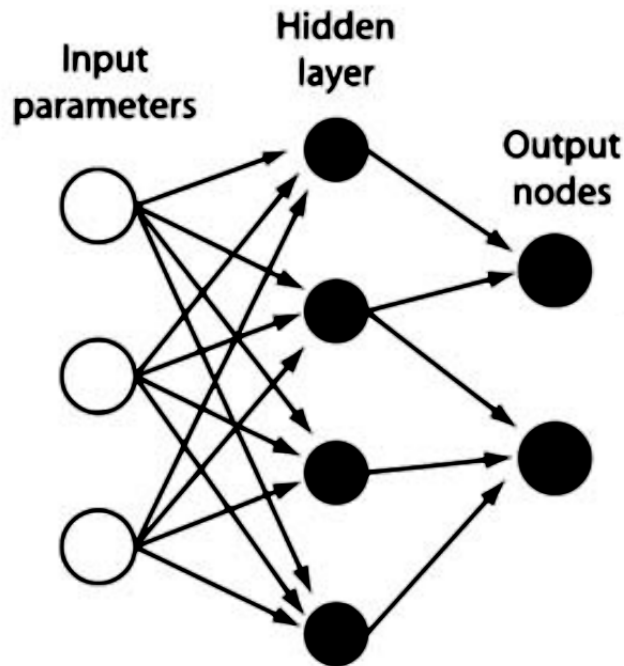
4.6.2 Multilayer Perceptron

The Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN). Nowadays, the ANNs are widely used to solve several challenges in different fields: pattern recognition, prediction, optimization, control, associative memory, among others (JAIN; MAO; MOHIUDDIN, 1996). The MLP is an improvement of the fundamental unit in neural networks: the Single Layer Perceptron (SLP). The SLP is a binary classifier, and it is defined by the equation 4.10.

$$f(x) = \begin{cases} 1, & \text{if } w \cdot x + b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.10)$$

Although being able to perform simple binary classifications, the SLP can't perform more complex operations. Furthermore, the SLP can only perform correct classification in input data that are linearly separable. These drawbacks are overcome when the SLP are used as networks (HAYKIN et al., 2009). This configuration, shown in Figure 22, is called Multilayer Perceptron.

Figure 22 – MultiLayer Perceptron.



Source: Modified from (DARLING; GUILAK, 2008)

The MLP is a supervised learning algorithm that can learn a function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^o$ from a dataset, where n is the number of dimensions (features) of the input and o is the number of dimensions of the output. Given an input feature vector $X = \{x_1, x_2, \dots, x_n\}$ and a target function y . Each neuron in the hidden layer transforms the values of the previous layer with a weighted sum $w_1x_1 + w_2x_2 + \dots + w_nx_n$, followed by a non-linear activation function (hyperbolic, tangent, sigmoid, among others). The output layer receives the values from the last hidden layer and transform it in output values, mapping them to the target function y . This makes the MLP capable of approximating a function, linear or not, for classification or regression.

The main advantages of the MLP are the capability to learn linear and non-linear models, even in real-time. On the other hand, the MLP has some drawbacks as the sensibility to scaling data, and the high model complexity that it can reach depending on the number of the hidden layers, that may make the system computationally expensive (HAYKIN et al., 2009).

4.6.3 Random Forest Classifier

Random Forest algorithm was presented firstly by Breiman (2001). Since then, it has been known as one of the most accurate machine learning algorithms, but is relatively new in the field of BCIs (STEYRL et al., 2016). The Random Forest algorithm is based

on decision trees, that are models able to map complex input spaces into simpler, discrete or continuous input spaces, splitting an original problem in several simpler and smaller ones (FANELLI et al., 2013).

Trees tend to learn highly irregular patterns, that is, there is a tendency to overfit the training sets. Random Forest are collections of decision trees, each one trained on a randomly subset of the input data, what helps reducing the overfitting. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the training set, with the goal of reducing the variance (FRIEDMAN; HASTIE; TIBSHIRANI, 2001). In this sense, this classifier is not sensible to the number of features, and can be used even when the number of features is greater than the number of observations.

4.7 Activation of External Devices

After classifying the signal among the classes, a BCI must activate the corresponding output. As shown in Section 3.2.4, there is several applications to SSVEP based BCIs, from spellers to robotic prosthesis. Besides activating a device, a BCI output also should provide user feedback. in SSVEP based BCIs, generally this feedback is visual, but it can be auditory or somatosensory (PINTO, 2011).

5 Development of an SSVEP-EEG Pattern Recognition System

In this chapter are presented the experimental procedure used to develop an SSVEP-EEG pattern recognition system. We introduce the tools used to develop the signal processing steps, as well as to estimate the reliability of the evaluated models and the processing time. All the code developed in this work is available and updated on the following public GitHub repository: https://github.com/rgiovanini/ssvep_files.

5.1 Open-Source Data Processing Tools: The Python Language for Scientific Computing

When performing scientific tasks, one of the most important tools for the scientist is the software framework that is used to perform calculations, simulate phenomena, develop and evaluate mathematical models, acquire, clean and process data, and so on. Due to the importance of the tasks to which the software will be submitted, there is a great need for reliability and robustness. At the same time, the programming language used in these frameworks should be easy to understand (even for non specialist programmers) and have a good performance in complex computation. On this account, several scientific computation software tools are available, such as the following strong and traditional programming languages like MATLAB, Mathematica, Mathcad, LabVIEW, among others. However, there are some fundamental issues with these products:

- These algorithms are proprietary, which means that the code inside the functions is not available to be modified or visualized. The user has to assume that the vendor's implementation is correct;
- The License fees, both for the standard version and for each tool box for such softwares, are usually very high;
- The portability is often a difficult task. Programs developed on newer versions may not run in older versions of the software, or in another machine with a different operating system;
- The software interoperability, that is, the ability of different softwares to communicate and work together, is impaired when proprietary softwares are used. These systems allow data types and processes used only for systems from a same manufacturer, forcing the user to purchase different softwares from a same manufacturer.

Aiming to provide an alternative tool to proprietary software products, different Open-Source programming languages have been developed. The Open-Source software is a development model that promotes a free licensing to design a software product, as well as its universal redistribution, which allows anyone to consult, examine, modify and reuse the product. An Open-Source Software does not just mean free access to the source code. This philosophy allows programmers and developers all around the world to be able to contribute with software projects individually or participate in groups of developers, incrementing in the enhancement of projects as well as in the creation of new ones. Currently there are several projects in active development, not just in the field of scientific programming, using Open-Source licenses: the Operating Systems GNU/Linux (and all its distributions), GNU, Android, the Mozilla project, the Apache Software, the Human Genome Project, and some programming languages like Python, Ruby and Perl, and several other projects. Some hardware companies also develop boards with similar open-hardware licenses, like Arduino and BeagleBoard. In this work, the chosen language was the Python programming language.

Created by Guido Van Rossum in 1991, Python is a high-level, multi-platform and free-of-charges programming language. Currently registered under an liberal Open-Source license, its structure is open and available to anyone who wants to contribute with new functionalities, documentation or bug fixing. Because of that, there is an extremely large and reliable source of information provided by the users' community.

Some advantages of Python over other proprietary languages are:

- Being free of charges, and totally open;
- Its code structure is easy to read and understand, with powerful standard libraries to general purpose tasks and useful datatypes like tuples, lists and dictionaries, that help to organize the code;
- When implemented in the correct way, it can be faster than other high level languages;
- General Purpose and multiplatform.

Python is a multi-platform language, that is, it works in different Operating Systems, like Windows, Unix-based platforms (e.g. Linux, FreeBSD, MacOS) and others. Since this language is native in Linux distributions, it is available in embedded applications like the one used in this work, the Raspberry Pi microcomputer.

The open-source base of Python also allows the development of tools for almost every work field in the Information Technology, from web-based application to specific scientific tasks. There are four specific tools that were used in this work: the array

management module Numpy, the scientific tools module Scipy, the machine learning framework Scikit-learn and a package to work with wavelets, PyWavelets.

Numpy and Scipy are frequently cited and used in scientific computation software development. Numpy ([NUMPY, 2016](#)) is an array management module that provides sophisticated functions to perform operations with multidimensional arrays. Besides that, Numpy scripts are internally performed in C language, so the use of Numpy arrays gives a faster code than using standard Python objects.

Scipy is a Python-based ecosystem for open-source software for mathematics, science and engineering, that is, Scipy is a module comprising several other Python modules frequently used in scientific computation, such as Numpy, a module to N-dimensional array management; Scipy library, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization and statistics; matplotlib, a mature and popular plotting package that provides publication-quality 2D and 3D plotting; Pandas, a module that provide high-performance and easy-to-use data structures; and IPython, a Python interactive programming interface ([SCIPY, 2017](#)):

Scikit-learn is a powerful module with functions that are used for data mining, data analysis and machine learning. It has several regression, classification and clustering algorithms, and it is widely used both in research and industry ([PEDREGOSA et al., 2011](#)).

PyWavelets is a free and Open-Source wavelet transform software for the Python programming language. It combines a simple high level interface with low level C and performance, making possible to easily perform operations with Continuous and Discrete Wavelet Transforms ([PYWAVELETS, 2017](#)).

5.2 Input Data: The Kołodziej SSVEP-EEG Dataset

The input block of any BCI system is the acquisition system. In this work, EEG data previously recorded from other tests were used, since the main objective was to investigate different EEG signal processing techniques and develop a reliable pattern recognition system. In this sense, it was used a set EEG data under SSVEP stimulation, which is developed and provided by [Kołodziej, Majkowski and Rak \(2015\)](#). This dataset is licensed under an Open-Source Creative Commons license, which means that the contents of this dataset can be freely used in scientific research and investigation, as long as the author's names are properly cited.

The experimental protocol used to acquire the signals consists of EEG records of five volunteers at the ages of 23, 25, 31, 42 and 46. The subjects were placed comfortably seated on a chair. A green LED of 1 cm diameter was placed at a distance of about 1

meter from the eyes of the person. EEG signals were recorded using the EEG acquisition device g.USBamp with 16 active electrodes.

The subjects were stimulated with the LED flickering at four individual frequencies: 5, 6, 7 and 8 Hz, during a test period of 30 seconds, at a sampling frequency of 256 Hz. The electrodes were placed according to the international 10-20 system at positions: O2, AF3, AF4, P4, P3, F4, Fz, F3, FCz, Pz, C4, C3, CPz, Cz, Oz and O1. The signals were recorded using a Butterworth bandpass filter (0.1-100 Hz) and notch filter (48-52 Hz). Finally, all the collected data were saved as MATLAB files (.mat) and made available on the Internet.

Since all the EEG signals are stored as MATLAB files, it were created functions to manage the data, like import .mat files into numpy arrays and create sets of specific data, according to the user's wishes. For this task, it was used the module `scipy.io`, which has several functions to deal with data input and output of different formats, including .mat files. Table 2 shows the headings of these functions, as well as a brief description of them.

Table 2 – Data Management and Organization Functions

Function name and arguments	Description
<code>select_eeg_all()</code>	Import all EEG files to an array
<code>select_eeg_channel(input_array, channel)</code>	Selects only the channel of interest
<code>select_eeg_frequency(freq)</code>	Select the tests with the specified frequency
<code>select_eeg_subject(subject)</code>	Select the tests with the specified subject
<code>normalize_array(input_array)</code>	Normalize the array values based on the maximum value element
<code>create_dataset(channel_list, save_path)</code>	Create a dictionary containing all the specified channels of interest.

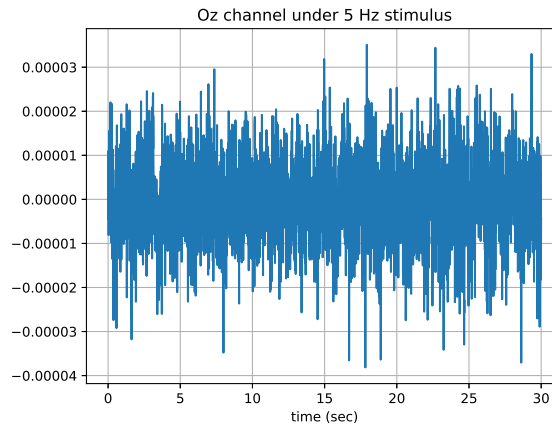
Source: From author.

As a data visualization example, a EEG signal from the Oz derivation (located exactly at the center of the visual cortex) for Subject 1 under a 5 Hz stimulus is shown in Figure 23.

Before performing preprocessing step, all the data was sorted in one single array. In this way, the operations are performed in the entire dataset, making the overall signal processing faster and simpler. The data array was sorted as follows.

The dataset consists of 20 .mat files (5 subjects, 4 frequencies). Each file comprises 16 channel EEG trial. Each trial takes 30 seconds, and it is sampled at 256 Hz. Thus, each test has 7680 points. The arrays were stacked in order to form a single array containing all the tests. Therefore, the resulting array (called input array) has a dimension of 320 (5 subjects x 4 frequencies x 16 channels) instances by 7680 points. The chosen arrangement to stack the individual arrays in order to form the input array was by class, that is, the tests are sorted by stimulus frequency, starting from 5 Hz to 8 Hz. Thus, the creation of a target vector is facilitated, as will be shown on Section 5.5.

Figure 23 – EEG from Oz channel of Subject 1 under a 5 Hz stimulus.



Source: From author

5.3 Preprocessing

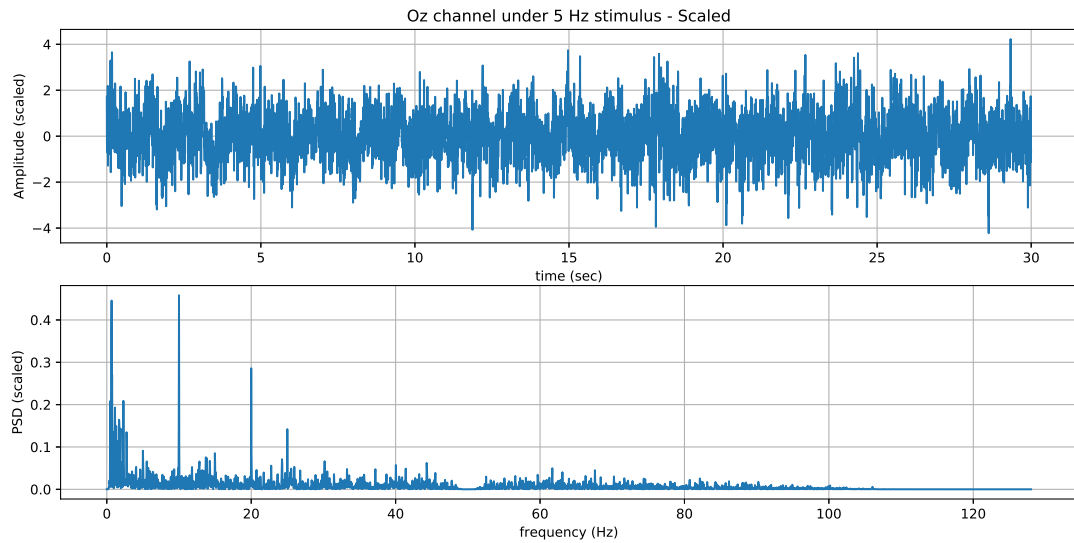
The preprocessing block is where the acquired signal receives a first processing, filtering, standardization or normalization, in order to prepare the signal to feature extraction block. In the case of the data used in this work, some preprocessing was already performed by the authors of the dataset (KOŁODZIEJ; MAJKOWSKI; RAK, 2015). As explained in Section 5.2, the data is available after two digital filtering steps, a band-pass of 0.1 to 100 Hz, to limit the influence of low (other body signals) and high (fluorescent lamps, radio frequency noise) frequency components; and a notch (band reject) filter, from 48 to 52 Hz, to decrease the power line interference.

It is possible to observe in Figure 23 that the amplitude values are in volts, lying around -30 and $30\mu V$. Working with such small values can bring a lot of inconveniences, such as issues with visualization, precision errors and even low accuracies and some types of classifiers (e.g. MLP). On this account, it was performed a standardization, or scaling, of the data, using the module `sklearn.preprocessing.scale`. After this step, all input data should have zero mean and unit standard deviation. In Figure 24, it is possible to observe the difference in the amplitude of the signal. Also, it is possible to notice the influence of the band-pass and notch filters.

5.4 Feature Extraction

To perform the classification, there is the need to extract some significant features from the signal, in order to increase the accuracy and reliability of the classifier. At the same way, it is necessary to reduce the number of features of the input array. In this work we analyzed two approaches to feature extraction and dimensionality reduction: performing

Figure 24 – Scaled EEG from Oz channel of Subject 1 under a 5 Hz stimulus.



Source: From author

signal downsampling and then applying a filter bank to extract useful spectral information, and applying a denoising and decimation using the Discrete Wavelet Transform.

5.4.1 Filter Banks Development

Using the module `scipy.signal`, which has several tools to analyze and design processing systems, it was developed some functions, aiming to facilitate the design and application of digital filters to the signal.

As seen in Section 3.2.4, the amplitude peaks occur even in the stimulation frequency and its harmonics. In some cases the amplitude peak is greater at second or third harmonic than at the first. In order to give the user ability to explore and analyze harmonics of his choice with different filters bandwidths, the functions `generate_band_tolerance` and `generate_harmonics` were created. With these functions, it is possible to easily create arrays corresponding to the number of desired harmonics to be analyzed and the pass-band bandwidths of each band. These arrays are used as arguments for another developed function, called `design_fir_butter`. This function perform the design of a FIR butterworth filter using the windowing method. The user can perform the design of different types (high-pass, low-pass, band-pass and band-reject) with the desired number of bands. This function return arrays specifying the filter impulse response coefficients and the results of its frequency response, internally performed. Regarding the filter order, the default value is 101, although the user can to change this value to any order.

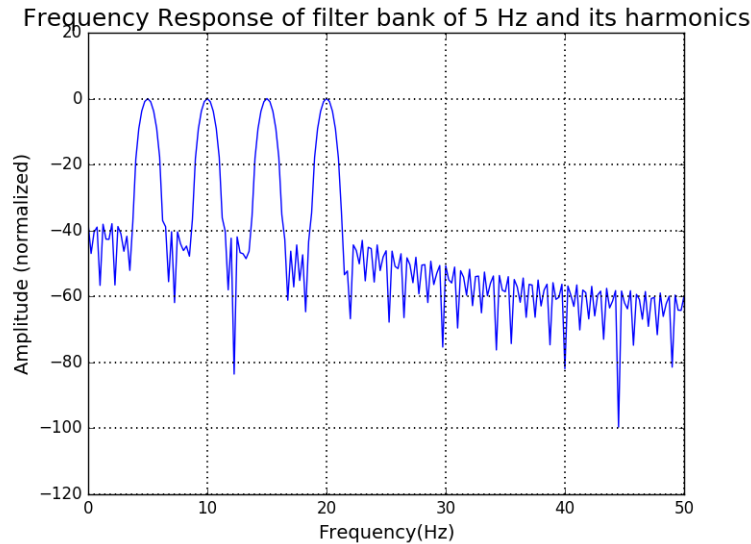
Table 3 – Digital filtering Functions

Function name and arguments	Description
<code>design_fir_butter(cutoff, f_sampl, order=101, band='pass')</code>	Design butterworth filter
<code>apply_fir_filter(coeff, input_array, axis)</code>	Convolution sum between filter and input signal
<code>generate_band_tolerance(freq, tol)</code>	Generate an array with specified band widths
<code>generate_harmonics(input_freq, number_of_harmonics)</code>	Generate an array with the specified harmonics

Source: From author.

In order to perform the feature extraction step with the available EEG signals, filter banks with pass-bands in the first four harmonics were developed. The designed filter bank to stimulation frequency of 5 Hz and its first four harmonics, with a tolerance of 0.3 Hz, is presented in Figure 25. One can observe that the specifications of the filter are all met.

Figure 25 – Filter bank with first four harmonics of 5Hz as bandpass.

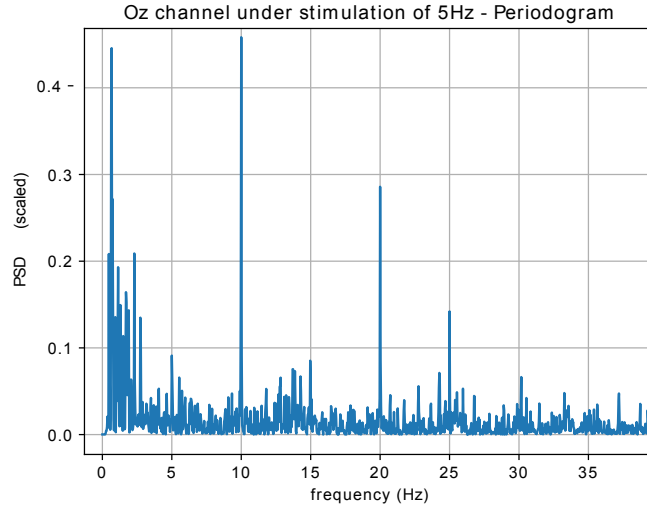


Source: From author

Through the application of the function `scipy.signal.periodogram`, the periodograms for each instance were obtained. The Oz channel periodogram of the Subject 1 under 5 Hz stimulation is shown in Figure 26.

It is noteworthy the occurrence of visible peaks at the stimulation frequency (5Hz) and its four harmonics. These are the relevant features that will be acquired in the case of this feature extraction method. This frequency peaks will compose the pattern to be recognized. However, there are a lot of undesirable signal components, especially at low frequencies. Using the function `apply_fir_filter` it is possible to easily apply the filter bank developed to the input data, obtaining both the filtered signal and its periodogram as function output. It is observable that the undesirable frequency components are then

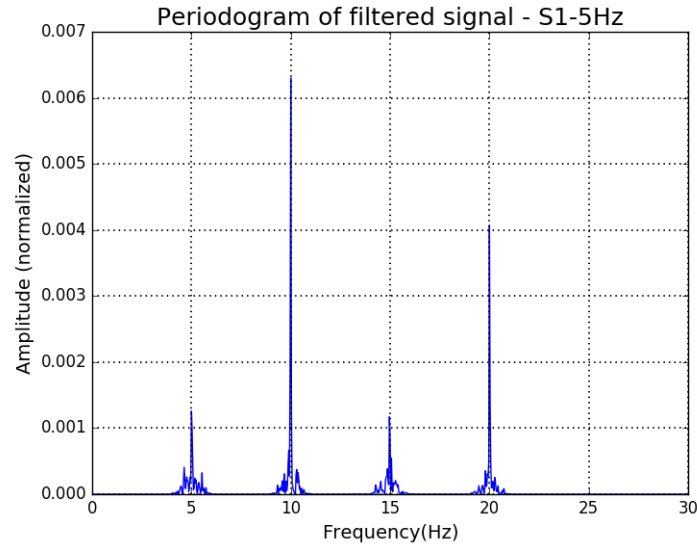
Figure 26 – Periodogram for Oz channel of Subject 1 under 5 Hz stimulation.



Source: From author

suppressed, as shown in Figure 27.

Figure 27 – Periodogram of the filtered signal for Oz channel of Subject 1 under 5 Hz stimulation.



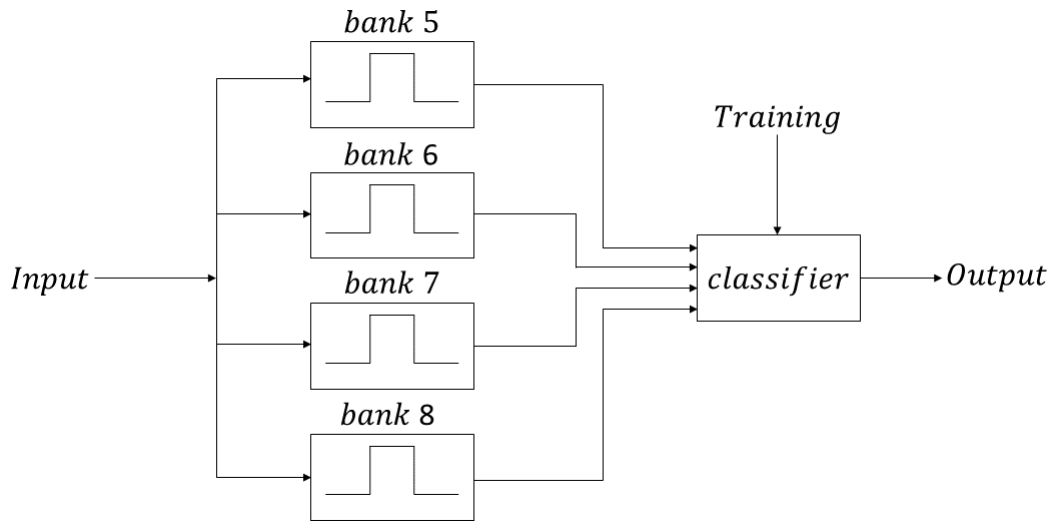
Source: From author

Aiming to reduce the effects caused by the curse of dimensionality, a signal down-sampling using the function `scipy.signal.decimate` was performed after the filtering step. This function perform the decimation of signal samples at a desired rate. To avoid the phenomenon of aliasing, shown in Section 4.5.3, this function perform a anti-aliasing

low-pass filtering before the decimation step. The chosen decimation rate to perform the downsampling was 100. This value was selected because one of the objectives of this work was to produce a comparison between the filter bank and the DWT decomposition feature extraction methods. As will be seen in Section 5.4.2, different decomposition levels were performed and, to obtain a similar number of features to that one achieved with the deeper DWT decompositions, downsampling by 100 was enough. With this value, the original number of features (7680) is reduced to 77.

The test structure for the filter bank approach is shown in Figure 28. In order to carry out tests involving all the possibilities, the input array was applied to all the filter banks, and the output arrays of each filter were stacked together, composing the feature vector for this case. The feature vector was then driven to the classification step.

Figure 28 – Structure of the filter banks feature extraction module.



Source: From author

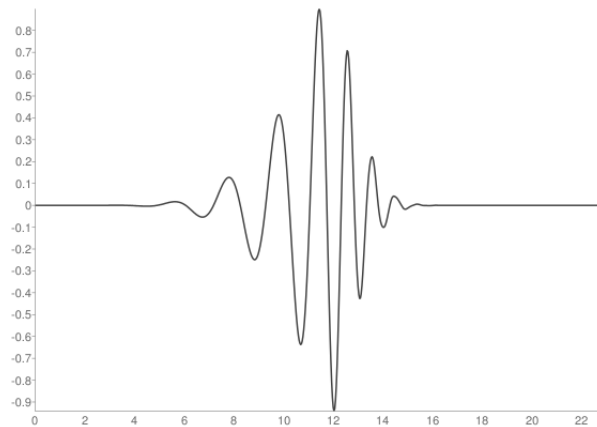
5.4.2 DWT Decomposition

The use of DWT is recommended when there are frequency components or short-time events inside the signal that are difficult to detect with traditional spectral techniques. In the case of this work, the subjects are under stimulation during all the tests, and the influence of visual stimulation can be seen through the use of the periodogram technique. However, there is a need for reduce considerably the number of features before the classification, and performing the usual downsampling step unrestrictedly can cause the loss of relevant information about the signal. Therefore, an alternative way to reduce the number of features is needed. DWTs are widely used to compressing data (FUGAL, 2009; GAO; YAN, 2011), so this technique was chosen for this work.

The DWT is performed through convolving the input signal with a wavelet filter, also called mother wavelet. As seen in Section 4.5.4, there are several wavelet families. In this work it was used a wavelet filter from Daubechies family, specifically the Daubechies-12 (db12) wavelet, shown in Figure 29. This choice was justified due to the shape of the pattern to be detected, the SSVEP occurrence, is a quasi-sinusoidal signal present in the EEG, at the same frequency of the stimulus. Choosing a wavelet filter with similar shape of a sinusoidal signal, a good correlation between the two signals was expected.

Using the PyWavelets module, DWTs in different decomposition levels were performed and analyzed, from level 4 until level 8, which is the maximum decomposition level for this signal length and for this wavelet filter. All the computational steps involving wavelet transforms were performed using `pyWavelets`, a Python package with intuitive functions able to perform operations with continuous and discrete wavelet transforms. As an DWT decomposition example, in the Figure 30 is shown a level 4 DWT decomposition of an EEG signal from the Oz channel, under a 5Hz stimulus. In this Figure, it is possible to observe the different decomposition levels of the Approximation and Details coefficients.

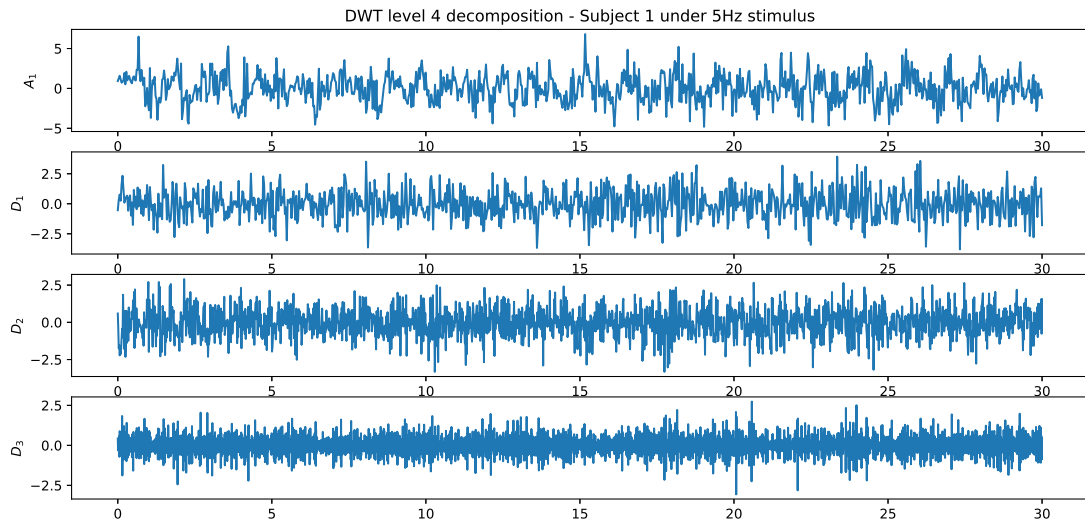
Figure 29 – db12 Wavelet.



Source: (PYBYTES, 2017)

The DWT is often used to detect rapid events in non-stationary signals. This is done by analyzing the high frequency coefficients (Details). However, in this work, since the subjects were under stimulation during all the tests, there is a SSVEP component through all recorded data. In this sense, only the Approximation coefficients were used, since the SSVEP coefficients have low frequency characteristic. Therefore, in the case of DWT decomposition used as feature extraction method, the feature vector obtained was the Approximation coefficients for each test, and the Details coefficients were removed.

Figure 30 – Level 4 DWT decomposition.



Source: From author

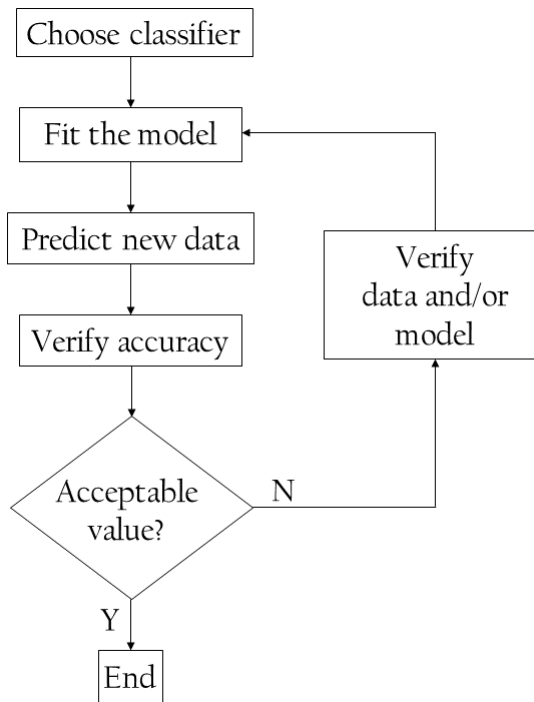
5.5 Classification

The classification step is the pattern recognition system itself. This block comprises a classification model (also called estimator) that is able to correctly determine which output class the input data belongs to. The general classification process comprises some basic steps, presented in the block diagram shown in Figure 31. After the preprocessing and feature extraction steps, the classification method must be chosen, and then fitted with the training set from the input data. After this, the test accuracy is evaluated using the test data, also from input data. If the obtained accuracy is not satisfactory, changes should be made in data representation and/or in classifier hyperparameters, and the process is repeated.

In order to perform the learning process, it is necessary to provide some representation about the desired outputs, called target vector. In the case of classification problems, that is a vector containing the correspondent outputs for each instance in the input vector. In this sense, arrays were created containing the labels 5, 6, 7 and 8, corresponding to each stimulus frequency, as output classes.

All of these steps, and also several complementary tools, are readily implemented using the package `sklearn`. This package was used in data treatment, model fitting, data predicting and model evaluation. The three classifiers used in this work, K-Nearest Neighbors, Multi-layer Perceptron and Random Forest, were loaded using the functions `sklearn.neighbors.KNeighborsClassifier`, `sklearn.neural_network.MLPClassifier`

Figure 31 – Block diagram illustrating a simplified classification process.



Source: From author

and `sklearn.ensemble.RandomForestClassifier`.

One of the main functionalities of the `sklearn` package is the flexibility that it offers when setting model parameters and hyperparameters. By default, all hyperparameters are already set using sensible default values, being possible to rapidly get the estimator ready to work. In this sense, a search for the best value of the main hyperparameter for each classifier was performed, being these:

- K-Nearest Neighbors: The number of neighbors, K ;
- Multi-layer Perceptron: The number of neurons in the hidden layer;
- Random Forest: the number of trees.

The accuracy of the model was evaluated using the function `sklearn.metrics.accuracy_score`. This function takes as input both the values predicted by the models and the truth values, generated by the target vector. As output, this function returns the accuracy score for the predictions, based on the equation 5.1.

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \quad (5.1)$$

where $\hat{y}_i$ is the predicted value of the i -th sample, y_i is the corresponding true value, and $1(x)$ is the indicator function.

Regarding model evaluation, a K-fold Cross Validation method with $K=6$ was used in order to evaluate the model reliability in function of the hyperparameter variations. All of these methods were applied using the functions `sklearn.model_selection.KFold` and `sklearn.model_selection.cross_val_score`.

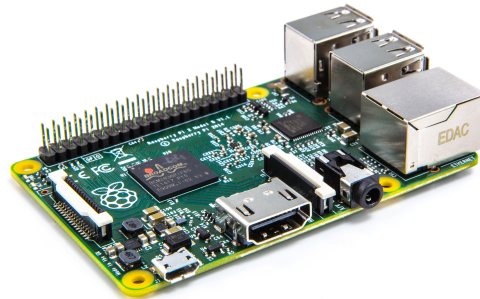
5.6 Execution Time Analysis

5.6.1 The Raspberry Pi Microcomputer

The great majority of BCI devices uses clinical EEG acquisition systems to measure brain signals, and personal computers (PC) to perform the preprocessing, feature extraction, classification and translation steps. This equipment have generally weight and dimensions that difficult a user to carry the BCI devices from one place to another. Furthermore, these equipments have high power consumption, making necessary the use of batteries, what would increase even more the weight, volume and, consequently, overall cost of the system.

Nowadays, due to the development of microelectronics, powerful and small sized microcomputers have been developed and used as embedded processing units in several applications. One of these microcomputers is the Raspberry Pi, which is shown in Figure 32. It is a small, versatile and powerful microcomputer, with dimensions of only 85x56 mm and weighting 45g, that can be embedded in a mobile device and used to process amounts of data that are difficult to deal using common-use microcontrollers. Its low power consumption (about 2A of current drawing) when compared to a desktop computer, makes it possible to power it with batteries, since a load control circuit is used (DONGARE; SHAH, 2016; YI et al., 2015).

Figure 32 – The Raspberry Pi Board (2nd generation).



Source: Element14.

The Raspberry Pi model 2 is based on the BCM2835 microprocessor, from Broadcom. This microprocessor is built based on ARM Cortex-A7 architecture, with 900 MHz of CPU capacity, what makes it capable to run Operating Systems (OS). In fact, there are several OS available to this board, most of them based on Linux distributions (Debian, Fedora, Ubuntu, Android, Slackware among others), and also there are some OS not based on Linux, like FreeBSD and Windows 10 IoT Core (MONK, 2015). In addition, the board has 1GB of SDRAM memory, GPU, 4 USB ports, video and audio interfaces and network adapter. There are 40 General Purpose Input/Output (GPIO) pins in which it is possible to interface with the external devices through some communication interfaces: UART, I^2C , SPI bus and I^2S audio interface.

Due its ease to setup and use, the Raspberry Pi has been widely used by scientist researchers and hobbyists from several different areas. In this work, the Raspberry Pi was being used as processing unit to detect SSVEP brain activity. The board will also, in future works, be used to generate the square waves to blink the LEDs in the stimulation circuit.

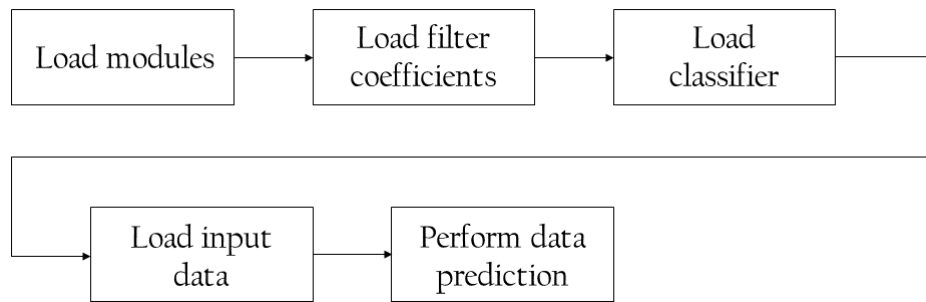
5.6.2 Structure of the Execution Time Evaluation Test

In future, the signal processing module developed in this work will be part of an entire SSVEP-EEG based pattern recognition system. Besides analyzing the model accuracies, an initial study was developed to obtain the execution time of the system. This study aims to analyze the time spent by the pattern recognition system to classify a segment with data recorded for one second. At a sampling rate of 256 Hz, one second of data will result in an array with 256 features. The section of data to be used is from Subject 1 under 5 Hz stimulus.

Initially, the three types of estimators used in this work (KNN, MLP and Random Forest) were fitted to an array with one second of data. One of the `sklearn` capabilities is the ability to save most of the program elements as Python objects. The estimators, after being fitted, can be saved and used in another system. The execution time analysis scripts were evaluated using a Raspberry Pi microcomputer, whose settings are in Section 5.6.1.

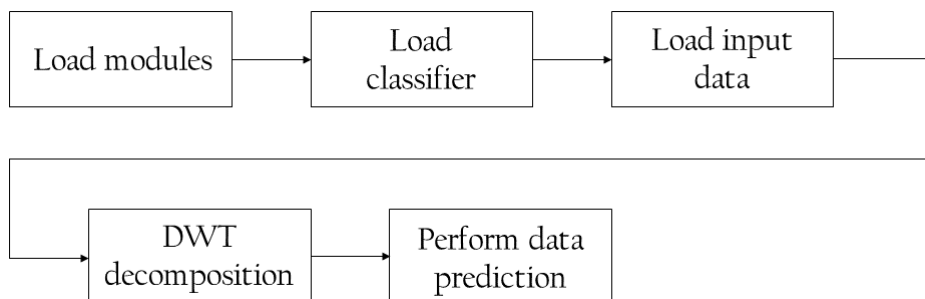
To evaluate the scripts time processing, the standard Python module `timeit` were used. This module provides a simple way to measure time in parts of a Python code. The basic structure of the scripts is given by the block diagrams shown in Figures 33 and 34 for filter bank and DWT approaches, respectively.

Figure 33 – Block diagram for time evaluation using filter bank approach.



Source: from author.

Figure 34 – Block diagram for time evaluation using DWT approach.



Source: from author.

6 Experimental Results and Discussion

In this chapter, the experimental results about the analysis of different pattern recognition techniques for EEG signals are presented. Software modules using Python language were developed, aiming to extract features from the signals, and generate machine learning models to perform the classification. The reliability and robustness of the results were confirmed using a cross validation technique. Furthermore, the execution time of each system was analyzed, aiming to choose the best combination of techniques to compose a real-time BCI pattern recognition system in the future.

6.1 Model Accuracy

After splitting the data to train and test sets, as shown in Section 5.5, initial tests were evaluated in order to find a combination of hyperparameters that resulted in the best possible accuracy. Using the cross validation procedure to evaluate the reliability each model, it were investigated the behavior of the model accuracy as function of the main hyperparameter variations.

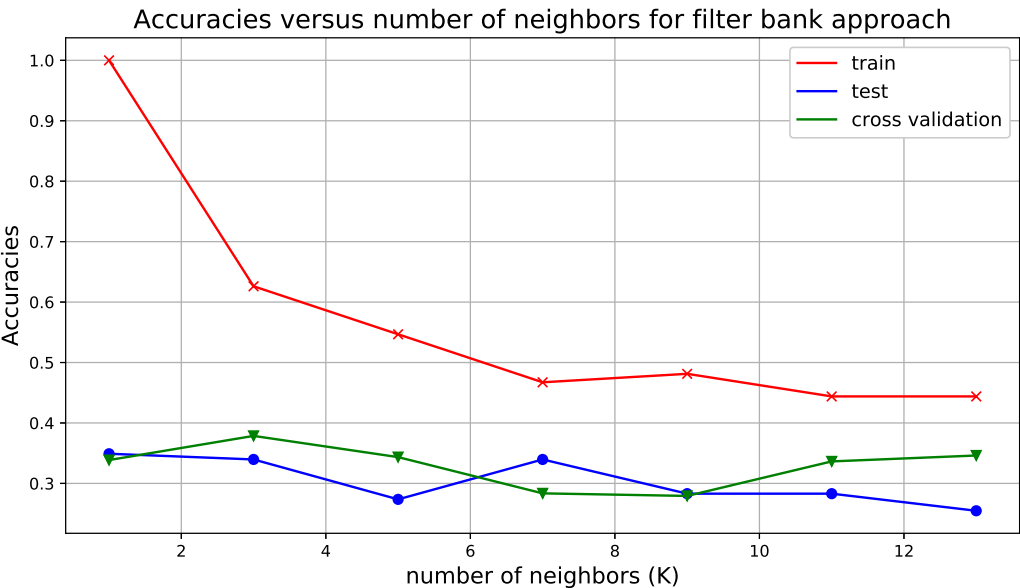
The input vector was submitted to the two different feature extraction methods (filter banks and DWT decomposition). Moreover, the accuracy, robustness and execution time between both were compared. The tests results and its respective Cross Validation scores are shown in the two subsequent sections.

6.1.1 Filter Banks

The input vector was applied to the filter banks in order to extract relevant features. After that, to reduce the dimensionality of the feature vector, a level 100 downsampling process was performed. At the end, the number of features in the feature vector was 77. The model accuracy variation in function of the model complexity to KNN, MLP and RF classifiers are shown in Figures 35, 36 and 37, respectively.

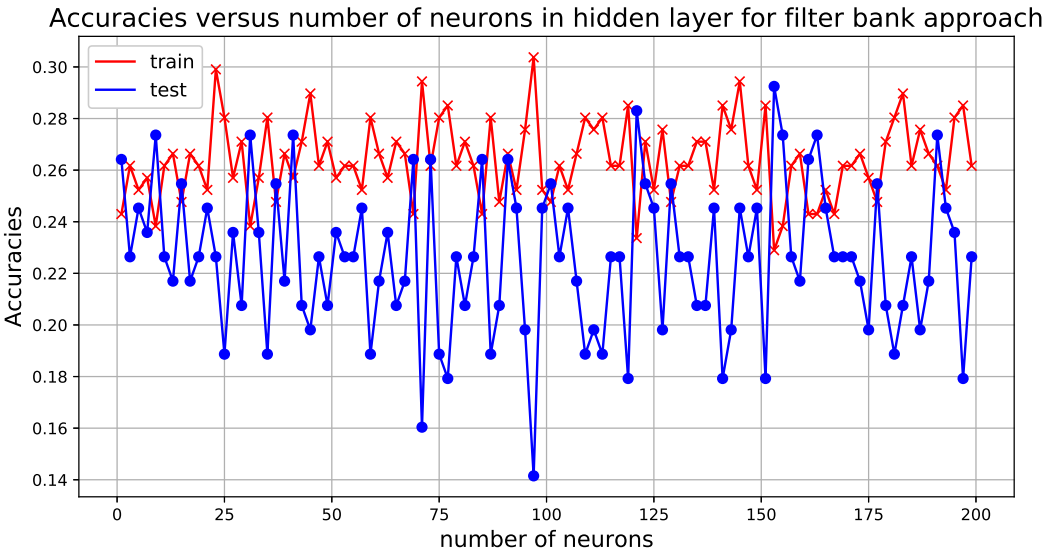
It is possible to see that all three classifiers had low accuracy, which can be explained by the downsampling performed after applying the filter. Useful informations about the frequency contents of the signals were lost, leading these classifiers to have a poor performance. In Figures 35 and 37, it is highly visible the high train accuracies in contrast with poor test accuracies, what indicates a clearly overfitting behavior. In addition, the low accuracies both for training and test in the Multilayer Perceptron shown in Figure 36 indicates an instability in that network. Hence, none of these estimators present generality and reliability, and would not be chosen to be part of a real BCI system.

Figure 35 – KNN accuracy versus number of neighbors for filter banks approach.



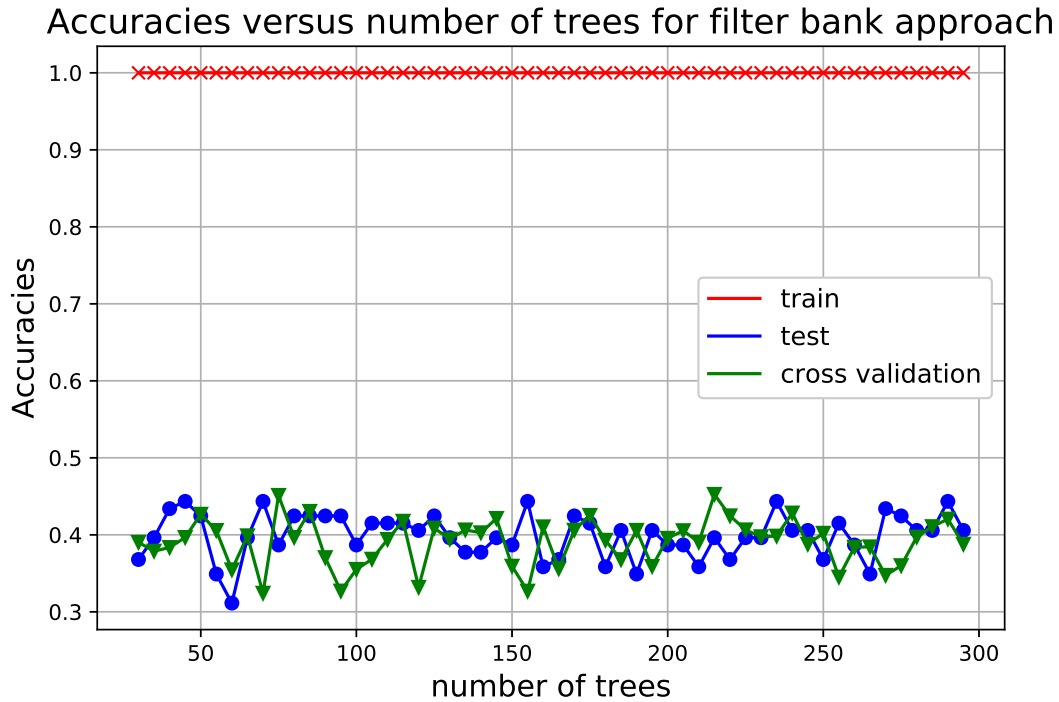
Source: From author

Figure 36 – MLP accuracy versus number of neurons in hidden layer for filter banks approach.



Source: From author

Figure 37 – RF accuracy versus number of trees for filter banks approach.



Source: From author

6.1.2 DWT Coefficients

When DWT were used to extract features, the Approximation coefficients was analyzed in four decomposition levels: 4, 5, 6, 7 and 8. The deeper the level the lower is the number of features, that are shown in Table 4, for each analyzed decomposition level of the DWT, using the db12 as mother wavelet. In this table, it is possible to observe that the number of features in level 8 decomposition (52) is considerably smaller than the number of features used in the filter bank approach with signal downsampling (77).

Table 4 – Number of features in function of DWT decomposition level.

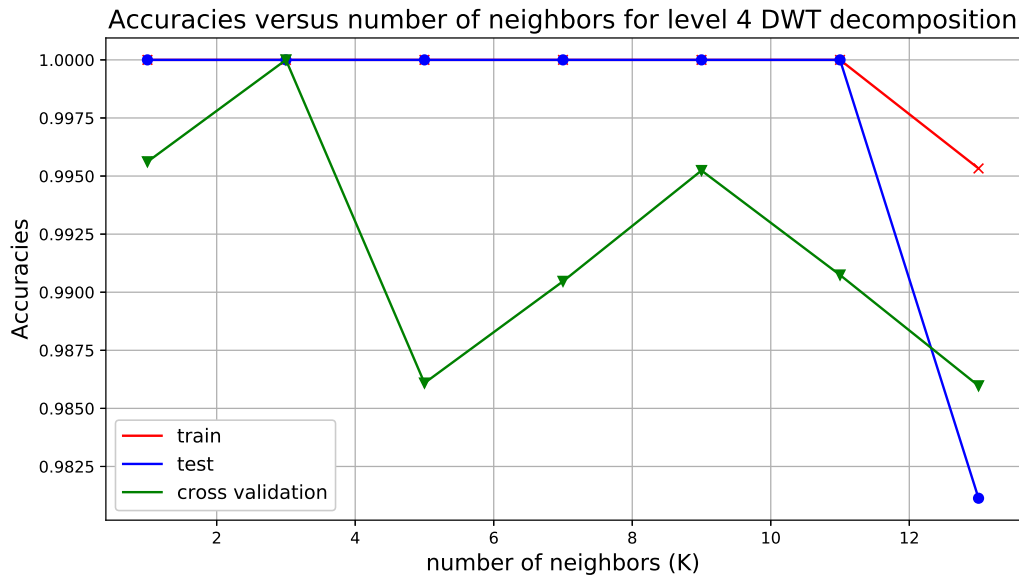
Decomposition level	4	5	6	7	8
number of features	501	262	142	82	52

Source: From author.

For each decomposition level, an analysis of model accuracy based on model complexity was performed. For the KNN classifier, these curves are shown in Figures 38, 39, 40, 41 and 42 for levels 4, 5, 6, 7 and 8, respectively. By analyzing this figures, it is noteworthy that this method is reliable only for low decomposition levels. In higher decomposition levels such as levels 7 and 8, the model reliability drops considerably. This behavior occurs because in lower decomposition levels there are more features to

be analyzed, that is, more relevant information about the signals. It is also visible that, higher the number of neighbors, lower the model reliability. In this case, a high number of neighbors means that the decision of which neighbor the test data will belongs will be calculated through the distance among several points, thus increasing noise susceptibility.

Figure 38 – KNN accuracy versus number of neighbors for level 4 DWT decomposition.

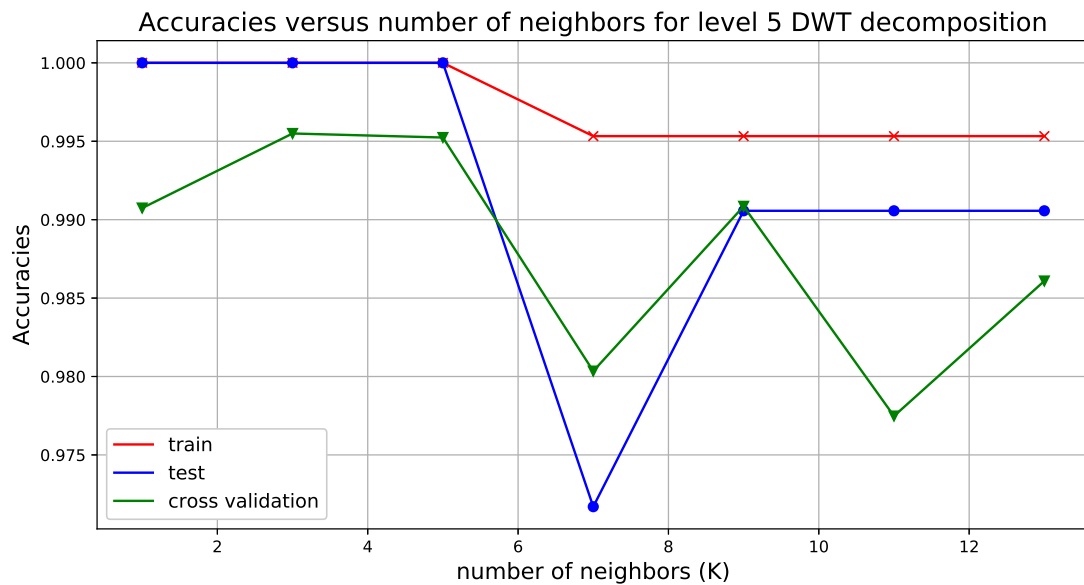


Source: From author

In the case of the MLP classifier, the accuracies were far better than KNN. The accuracies versus number of neurons of the hidden layer for 4, 5, 6, 7 and 8 levels of decomposition are shown in Figures 43, 44, 45, 46 and 47, respectively, and the mean accuracies are presented in Table 5. In this case, high values of train and test accuracies were obtained until level 7 decomposition. Although the model reliability decreases when applying data from the deepest DWT decomposition level, one can note that the known ability of the Multilayer Perceptron to detect patterns, even with a small amount of data, is confirmed. Even with just 82 features in the feature vector (level 7 decomposition) it was possible to achieve test accuracies greater than 98.02%. In this decomposition level, the test accuracy was increased when a greater number of neurons (starting from 75) were applied to model the problem, as can be seen in Figure 46. Also, considering the level 8 decomposition in Figure 47, it is possible to see that the accuracy does not increase even with greater amounts of neurons, what indicates the network's inability to deal with such small amount of features.

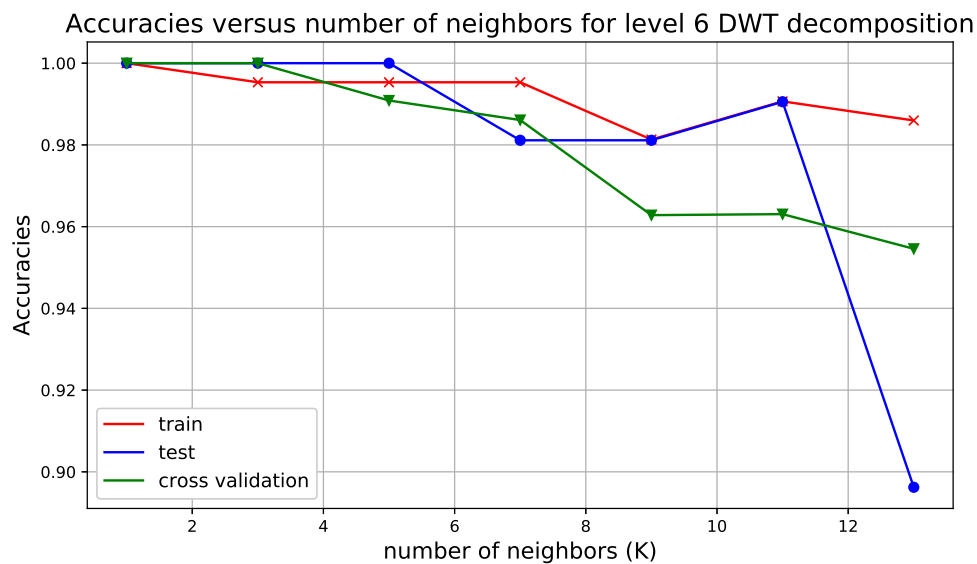
In the same way, graphs regarding the tests with the Random Forest classifiers are shown in Figures 48, 49, 50, 51 and 52, respectively. The mean accuracies using this type of classifier are shown in Table 6. The Random Forest classifier performed

Figure 39 – KNN accuracy versus number of neighbors for level 5 DWT decomposition.



Source: From author

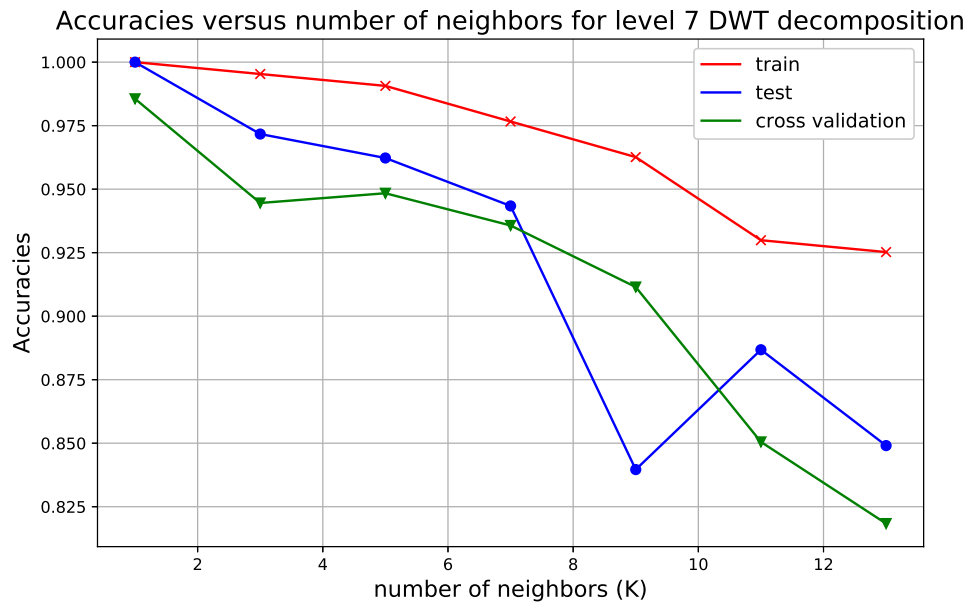
Figure 40 – KNN accuracy versus number of neighbors for level 6 DWT decomposition.



Source: From author

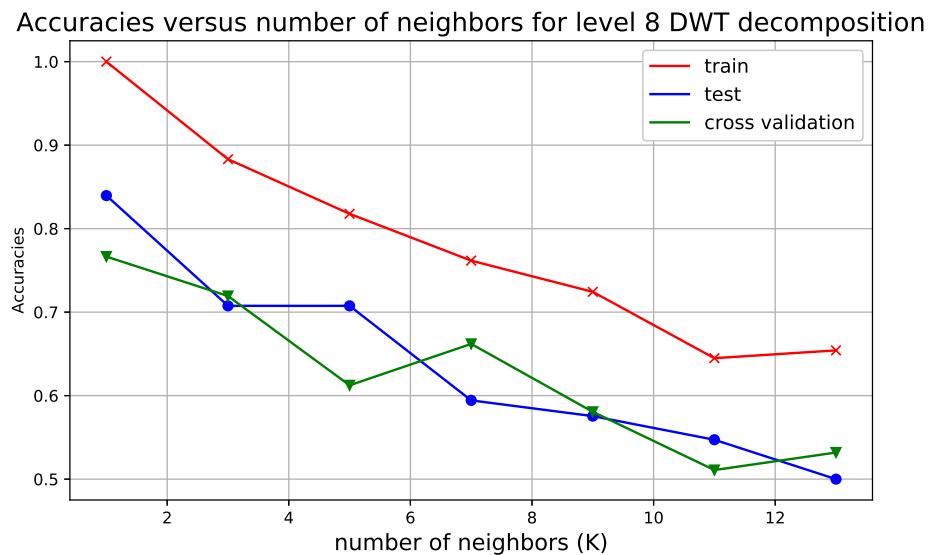
the task with better accuracy than Multilayer Perceptron. In this case, similar to the other methods, the accuracies had lower values in level 8 decomposition when compared to lower decomposition levels. However, the Random Forest classifier obtained the best accuracy of the three classifiers when dealing with the least amount of data. While in

Figure 41 – KNN accuracy versus number of neighbors for level 7 DWT decomposition.



Source: From author

Figure 42 – KNN accuracy versus number of neighbors for level 8 DWT decomposition.



Source: From author

level 8 decomposition the test accuracies of KNN and MLP reached numbers below 40% and 87.34%, respectively, the correspondent test accuracy reached by the RF classifier was 92.85%. It is also observable that the RF classifier obtained the greatest accuracy rates

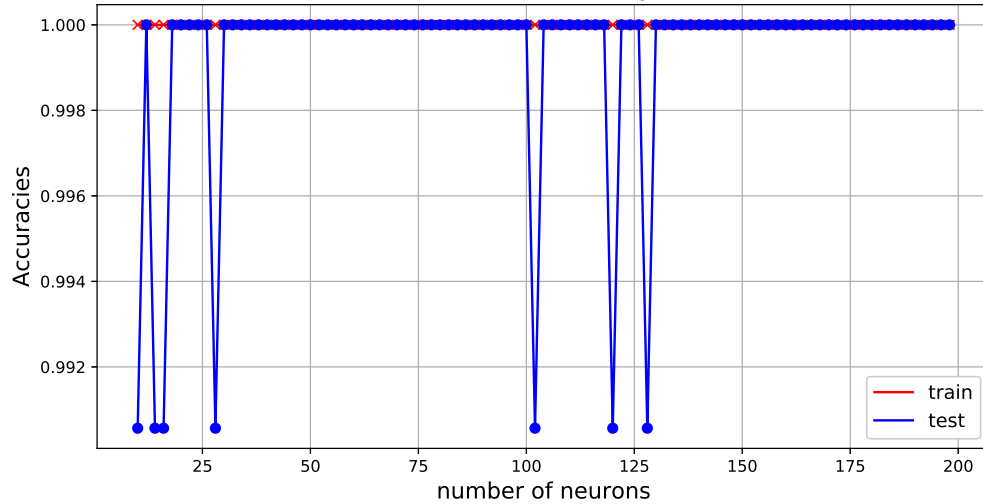
Table 5 – Mean accuracies for different decomposition levels using MLP classifier.

level	4	5	6	7	8
train (%)	100	100	100	99.99	95.70
test (%)	99.93	99.65	99.19	98.02	87.34

Source: From author.

Figure 43 – MLP accuracy versus number of neurons in hidden layer for level 4 DWT decomposition.

Accuracies versus number of neurons in hidden layer for level 4 DWT decomposition.



Source: From author

with a simplest model, that is, with less computational power. It is possible to observe that the accuracies of the RF classifier never drops below 92%, while the other classifiers have, with its simplest models, accuracies below 50%. This confirms the robustness and reliability of the Random Forest classifier.

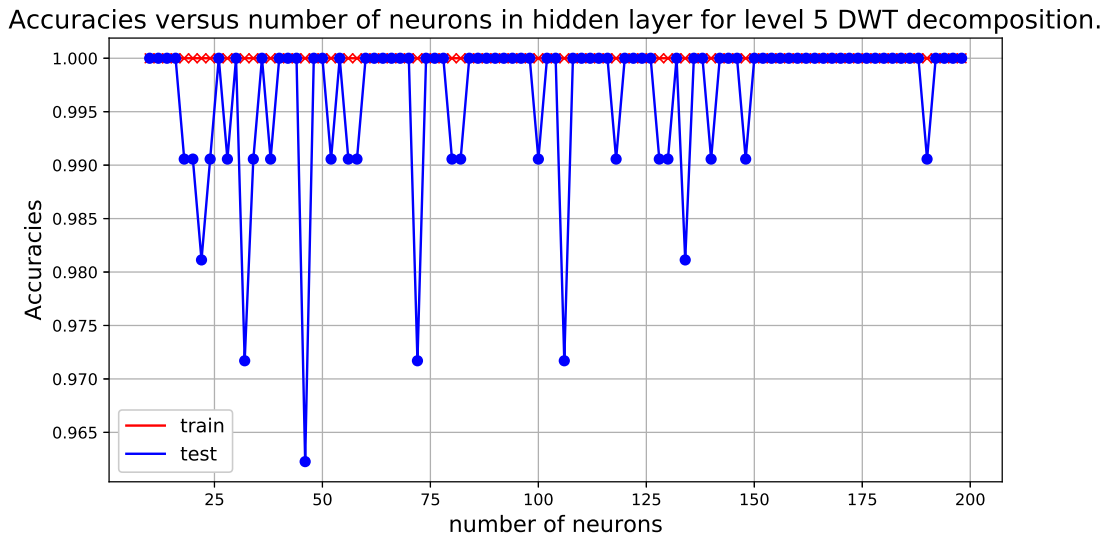
Table 6 – Mean accuracies for different decomposition levels using Random Forest classifier.

level	4	5	6	7	8
train (%)	100	100	100	100	100
test (%)	99.91	99.91	99.56	98.64	92.85

Source: From author.

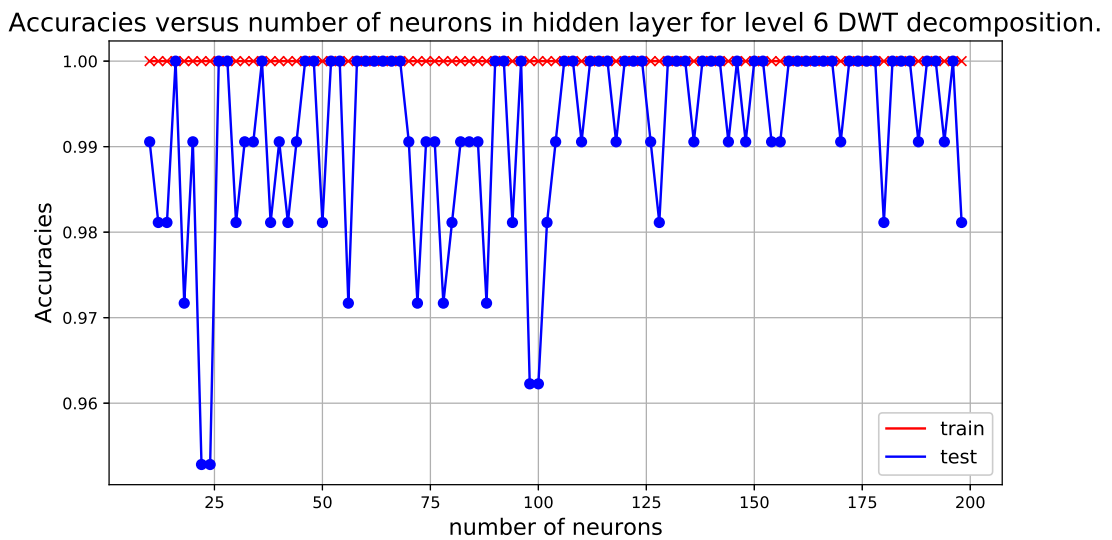
All of the classifiers had acceptable accuracy using DWT coefficients as features. By analyzing all of these graphs, it is possible to observe that extracting features from the signal using the DWT is far better than the filter bank approach. All the classifiers had an acceptable accuracy in lower decomposition levels, and the Random Forest classifier reached the greatest rates at higher decomposition levels. These high accuracies occur due to the DWT computation, that separates the signal components with higher correlation

Figure 44 – MLP accuracy versus number of neurons in hidden layer for level 5 DWT decomposition.



Source: From author

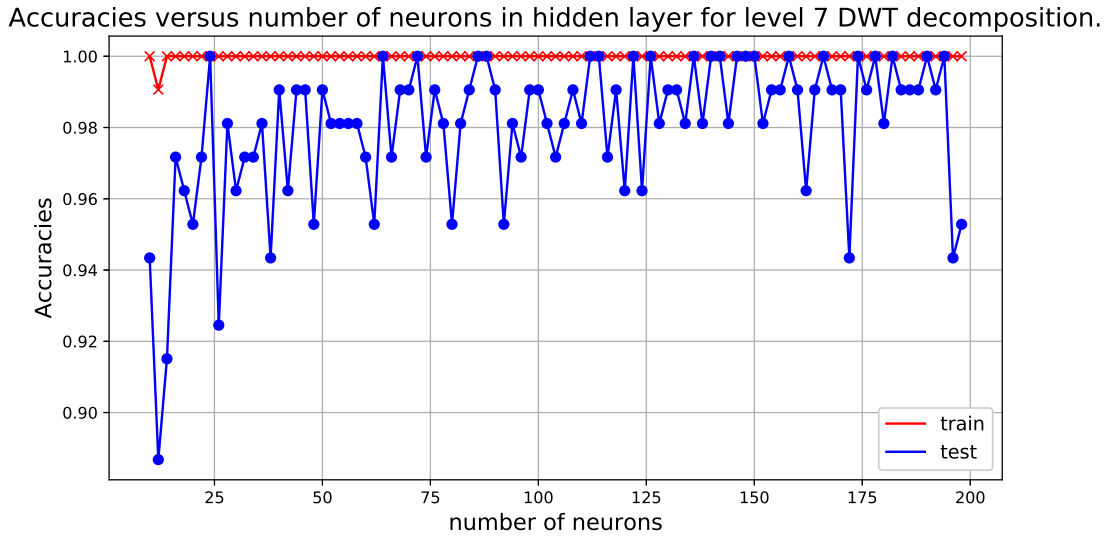
Figure 45 – MLP accuracy versus number of neurons in hidden layer for level 6 DWT decomposition.



Source: From author

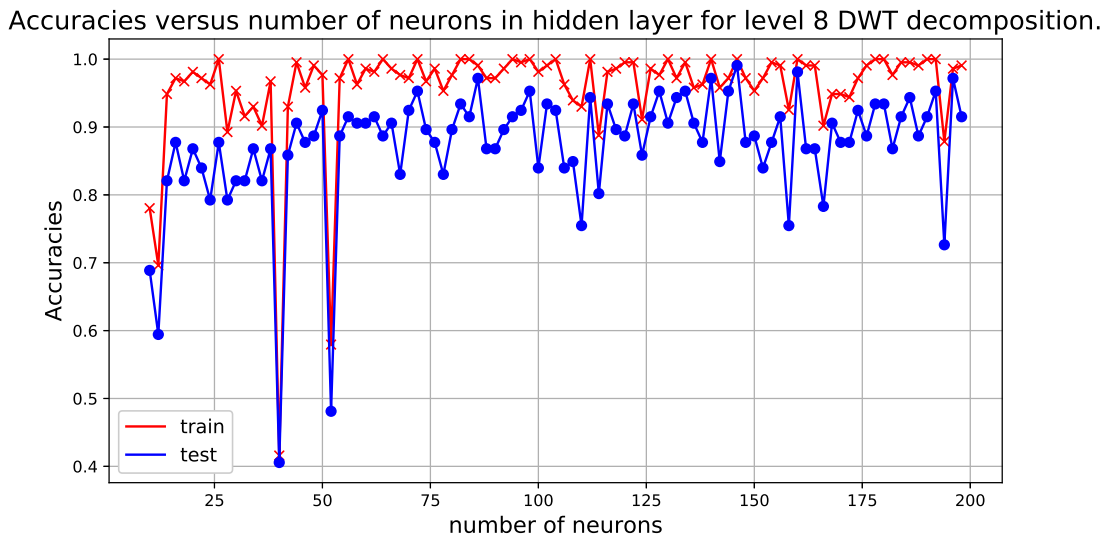
with the mother wavelet. This way, the loss of significant samples is smaller than the conventional downsampling process.

Figure 46 – MLP accuracy versus number of neurons in hidden layer for level 7 DWT decomposition.



Source: From author

Figure 47 – MLP accuracy versus number of neurons in hidden layer for level 8 DWT decomposition.

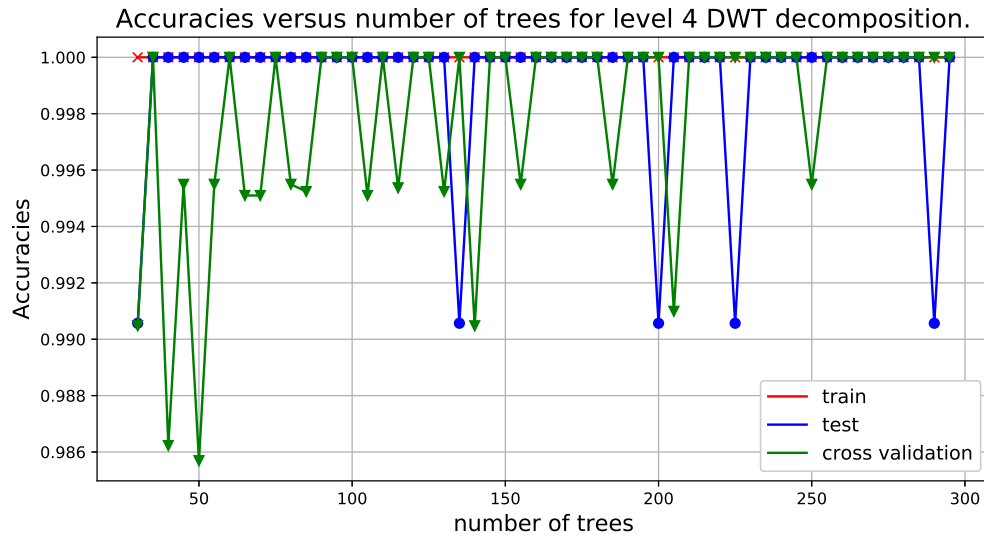


Source: From author

6.2 Execution Time

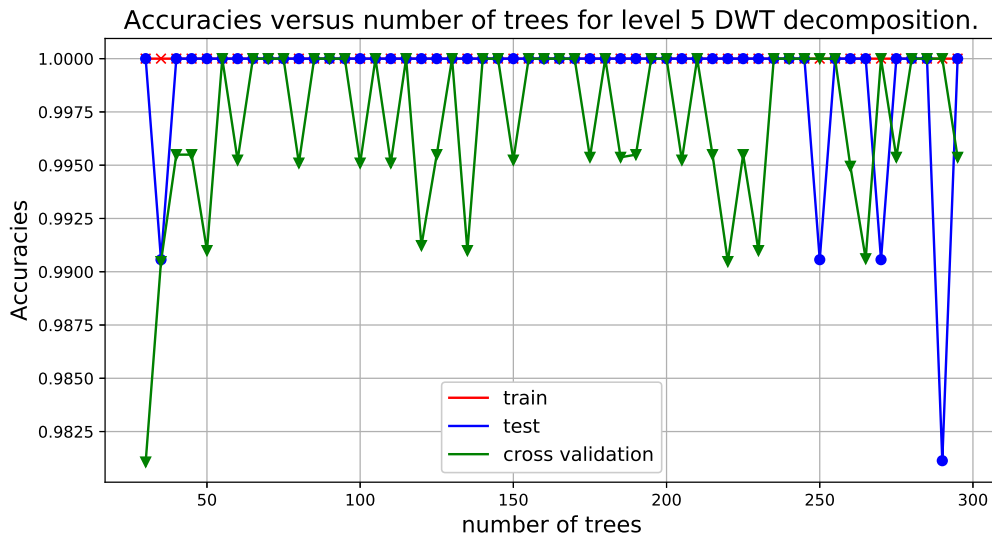
At this point one can observe that using DWT decomposition to extract features is the best choice for this work, due to its high accuracy, even when dealing with a lower amount of data. Also, in order to be correctly applied in real-time applications, a pattern

Figure 48 – RF accuracy versus number of trees for level 4 DWT decomposition.



Source: From author

Figure 49 – RF accuracy versus number of trees for level 5 DWT decomposition.

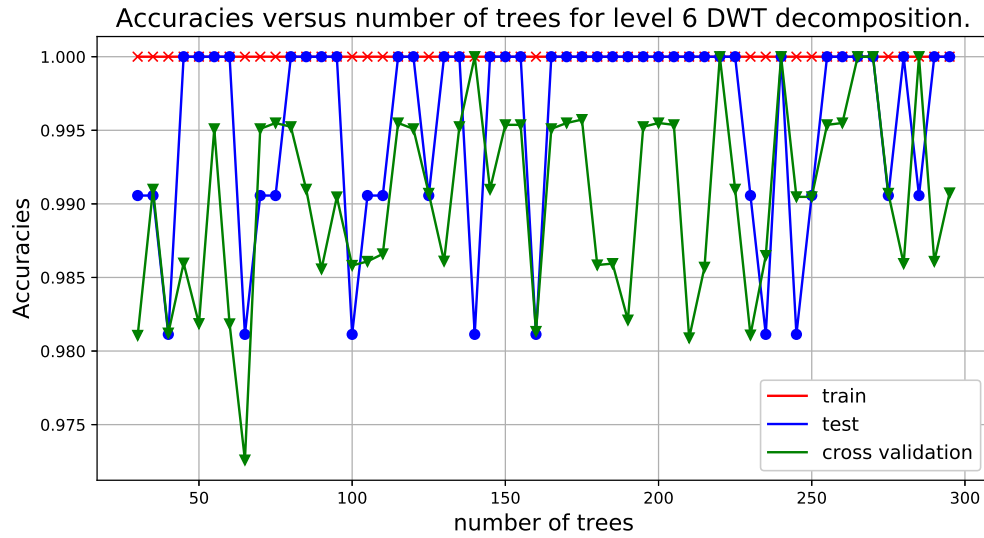


Source: From author

recognition system should have high accuracy rates and, at the same time, the lowest processing time as possible.

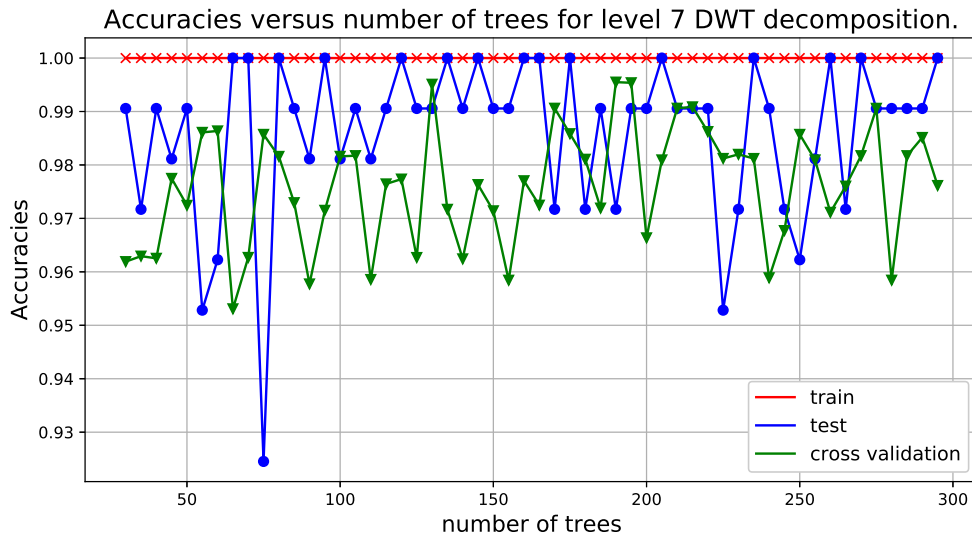
In this section, in order to choose the best classifier to the system, the time spent to perform the feature extraction and classification steps in the embedded computer Raspberry Pi version 2 were analyzed for the two feature extraction methods (filter banks and DWT decomposition).

Figure 50 – RF accuracy versus number of trees for level 6 DWT decomposition.



Source: From author

Figure 51 – RF accuracy versus number of trees for level 7 DWT decomposition.

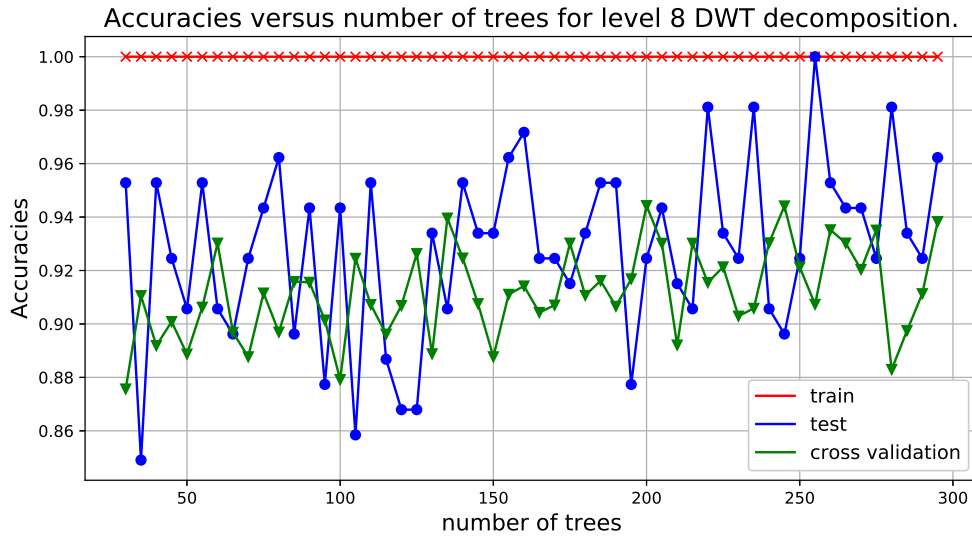


Source: From author

6.2.1 Filter Banks

The time spent to execute the scripts using the filter bank approach and different classifiers are shown in Table 7. Using this approach, Multilayer Perceptron was the algorithm that took the longest time to process the signal, taking 2.13 seconds to perform a classification. In this instance, the fastest algorithm was Random Forest, taking 0.111 seconds. However, no classifier obtained an acceptable accuracy score using the filter bank

Figure 52 – RF accuracy versus number of trees for level 8 DWT decomposition.



Source: From author

approach. Therefore, when using filter bank method, none of these classifiers would be useful in this case.

Table 7 – Execution time using filter banks.

Classifier	K-Nearest Neighbors	Multilayer Perceptron	Random Forest
time (seconds)	1.08	2.13	0.111

Source: From author.

6.2.2 DWT coefficients

Besides having a far better accuracy, it is noticeable that the processing time by the DWT approach is much lower than for the filter bank method. This occurs because, although it is a type of filter bank, the DWT computation is more efficient than the filter banks used in the other case. The wavelet lengths are much smaller than the length of the multi-band filter bank, which leads to a faster convolution between the signal and the wavelet. This behavior can be confirmed through the analysis of Table 8.

After analyzing the execution times using DWT approach, one can confirm that the most time-efficient classifier is the Multilayer Perceptron. By using this algorithm, it is possible to perform DWT decomposition and classification more than 5 times faster than KNN, and over than 20 times faster than Random Forest.

Regarding the obtained results, the feature extraction methodology and the classifier were chosen aiming to optimize the trade-off between processing time and accuracy of

Table 8 – Execution time using DWT (seconds).

level	4	5	6	7	8
KNN	0.297	0.248	0.223	0.207	0.193
MLP	0.0456	0.0487	0.0471	0.0464	0.0471
RF	1.03	1.04	1.04	1.04	1.04

Source: From author.

classification. When comparing the classifiers using DWT as feature extraction, the highest classification rate were from the Multilayer Perceptron and the Random Forest with decomposition levels above 8. Furthermore, in all cases is possible to note that, when the signals are decomposed until the maximum level using DWT, although the features are more relevant than those in filter bank approach, there are significant losses in meaningful information about the original signal. Based in the models accuracies, both Multilayer Perceptron and Random Forest are reliable to be embedded in a real-time BCI system. However, by analyzing Table 8 one can note that the time spent by the Multilayer Perceptron to achieve the task was considerably smaller than the others. In that case, the most ideal pattern recognition system to be used in the SSVEP-EEG BCI is the Multilayer Perceptron, studied in this work, using a feature extraction step based in the Approximation coefficients from DWT decompositions. At the same time, depending on the time requirements of the application, the Random Forest classifier can also be applied.

7 Conclusion

The work presented here shows, in details, the study and implementation of the main components in pattern recognition systems. These components were applied to processing EEG signals under visual stimulation, also known as SSVEP-EEG. Studies and implementations about the main preprocessing, feature extraction and classification techniques were developed.

Using only Open Source software tools, such as Python programming language, software modules were developed in order to perform SSVEP-EEG signal processing, as well as to determine the correct classification. As input data, previously recorded signals from five volunteers under visual stimulus of four different frequencies were used. Functions to easily manipulate and arrange data were generated, allowing the analysis of the SSVEP-EEG data in a more flexible manner.

Based on the results obtained after testing two feature extraction methods (filter banks and DWTs), it is possible to conclude that, in this case, the use of the naive approach using filter banks and periodograms produces very low accuracies, even with carefully choosing filters and windows to take care of aliasing. In addition, the time spent by this method to perform the tasks were far greater than the other approaches. Using this approach, useful information about the signal were lost, leading the classification systems to overfitting, in the case of KNN and Random Forest, and instability, in the case of MLP.

By its turn, the approach based on DWTs resulted in better accuracy rates and considerably lower processing times. Using the coefficient generated by the deepest DWT decomposition level using db12 wavelet, were obtained accuracies of 87.34% and 92.85% using, respectively, MLP and Random Forest classifiers. The K-Nearest Neighbors, although accurate in lower decomposition levels, was not able to guarantee its reliability to deeper levels. The Random Forest classifier obtained the highest accuracy within all of the classifiers. However, the time spent by Multilayer Perceptron to achieve the task was considerably lower, what makes this the chosen classifier. It is important to note that there is room for improvements in the developed system, since just some of the hyperparameters were tested in the classifiers: number of neighbors, number of neurons in hidden layer and number of trees. In the same way, there is the possibility to perform other changes, such as the mother wavelet, wavelet family, classification algorithm and even other feature extraction methods.

The neuroengineering is an attractive research field, with increasing funding and a promising future. This is a preliminary and pioneer study within the Laboratory of

Electronic Instrumentation and Biomedical Engineering. Starting from this work, some future works are proposed, such as:

- Development of a reliable and low cost EEG acquisition system;
- Study of new stimulation techniques to generate evoked potentials, like multi-stimulus SSVEP, P300 potentials, auditory or somatosensory evoked potentials;
- Study and development of new techniques for EEG signal processing and detecting, as well as new paradigms for BCI implementations;
- Development of a real-Time device control by acquiring and decoding EEG signals from the user;
- Contributing with the scientific computation community around the world, developing and providing Open Source tools and techniques.

Bibliography

- ABRAHAM, A. et al. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, Frontiers, Lausanne, v. 8, n. 14, p. 1–10, 2014. 45
- ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT, 2014. 42
- BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. *Neurociências: desvendando o sistema nervoso*. 3. ed. [S.l.]: Lippincott Williams & Wilkins, 2007. 17
- BIRBAUMER, N. et al. The thought translation device (ttd) for completely paralyzed patients. *IEEE Transactions on rehabilitation Engineering*, Piscataway, v. 8, n. 2, p. 190–193, 2000. 31
- BISHOP, C. M. *Pattern recognition and Machine Learning*. Heidelberg: Springer, 2006. 35, 36, 37, 41, 42
- BREIMAN, L. Random forests. *Machine learning*, New York, v. 45, n. 1, p. 5–32, 2001. 52
- BRONZINO, J. D. *Biomedical engineering handbook*. [S.l.]: CRC, 1999. v. 2. 23, 25, 26, 40
- BYRNE, J.; DAFNY, N. *Neuroscience Online: an electronic textbook for the neurosciences*. [S.l.: s.n.], 1997. 17, 18, 19
- CAGLAYAN, O. *A Portable Embedded SSVEP BCI System: emBCI*. 2014. 57 f. Dissertação (Master in Electrical Engineering) — Institute of Science and Engineering, Galatasaray University, Galatasaray, 2014. 21
- CHANG, M. H. et al. Eliciting dual-frequency ssvep using a hybrid ssvep-p300 bci. *Journal of neuroscience methods*, Amsterdam, v. 258, p. 104–113, 2016. 34
- CHAPIN, J. K. et al. Real-time control of a robot arm using simultaneously recorded neurons innervate the motor cortex. *Nature neuroscience*, New York, v. 2, n. 7, p. 664–670, 1999. 19, 22
- CHAUDHARY, U.; BIRBAUMER, N.; RAMOS-MURGUIALDAY, A. Brain–computer interfaces in the completely locked-in state and chronic stroke. *Progress in brain research*, Amsterdam, v. 228, p. 131–161, 2016. 30
- CREEL, D. *Visually evoked potentials*. [S.l.]: University of Utah Health Sciences Center, 2012. 32
- DARLING, E. M.; GUILAK, F. A neural network model for cell classification based on single-cell biomechanical properties. *Tissue Engineering Part A*, New York, v. 14, n. 9, p. 1507–1515, 2008. 52
- DIEZ, P. F. et al. Commanding a robotic wheelchair with a high-frequency steady-state visual evoked potential based brain-computer interface. *Medical engineering & physics*, Amsterdam, v. 35, n. 8, p. 1155–1164, 2013. 34
- DIEZ, P. F. et al. Asynchronous bci control using high-frequency ssvep. *Journal of neuroengineering and rehabilitation*, London, v. 8, n. 1, p. 39, 2011. 47, 48

- DONCHIN, E.; SPENCER, K. M.; WIJESINGHE, R. The mental prosthesis: assessing the speed of a p300-based brain-computer interface. *IEEE Transactions on rehabilitation engineering*, Piscataway, v. 8, n. 2, p. 174–179, 2000. 31
- DONGARE, H.; SHAH, S. Eye gaze tracking and eyes off the road detection for traffic safety on raspberry pi. *Eye*, Thorofare, v. 4, n. 6, 2016. 66
- ESLING, P.; AGON, C. Time-series data mining. *ACM Computing Surveys (CSUR)*, New York, v. 45, n. 1, p. 12, 2012. 45
- FANELLI, G. et al. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, New York, v. 101, n. 3, p. 437–458, 2013. 53
- FARWELL, L. A.; DONCHIN, E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, New York, v. 70, n. 6, p. 510–523, 1988. 30
- FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *The elements of statistical learning*. [S.l.]: New York, 2001. v. 1. 53
- FRUITET, J.; MCFARLAND, D. J.; WOLPAW, J. R. A comparison of regression techniques for a two-dimensional sensorimotor rhythm-based brain-computer interface. *Journal of Neural engineering*, Bristol, v. 7, n. 1, p. 016003, 2010. 38
- FUGAL, D. L. *Conceptual wavelets in digital signal processing*. San Diego: Space and Signals technical publishing, 2009. 48, 49, 62
- FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers*, Piscataway, v. 100, n. 7, p. 750–753, 1975. 50
- GAO, R. X.; YAN, R. *Wavelets: theory and applications for manufacturing*. [S.l.]: Springer Science & Business Media, 2011. 47, 62
- GARCIA, V.; DEBREUVE, E.; BARLAUD, M. Fast k nearest neighbor search using gpu. In: IEEE COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2008. *Proceedings...* Anchorage: IEEE, 2008. p. 1–6. 50, 51
- GNANAYUTHAM, P.; BLOOR, C.; COCKTON, G. Discrete acceleration and personalised tiling as brain body interface paradigms for neurorehabilitation. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2005. *Proceedings...* [S.l.]: ACM, 2005. p. 261–270. 14
- GOLLEE, H. et al. An SSVEP-based brain-computer interface for the control of functional electrical stimulation. *IEEE Transactions on Biomedical Engineering*, Piscataway, v. 57, n. 8, p. 1847–1855, 2010. 34
- GRAIMANN, B.; ALLISON, B. Z.; PFURTSCHELLER, G. *Brain-computer interfaces: Revolutionizing human-computer interaction*. [S.l.]: Springer Science & Business Media, 2010. 21, 24, 28, 29, 30, 32, 33
- GÜNEYSU, A.; AKIN, H. L. An SSVEP based BCI to control a humanoid robot by using portable EEG device. In: ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY (EMBC), 35., 2013. *Proceedings...* Osaka: IEEE, 2013. p. 6905–6908. 19

- GUYTON, A. C.; HALL, J. E. *Tratado de fisiologia médica*. [S.l.]: Elsevier Brasil, 2006. 19, 20, 29
- HAKEN, H. *Principles of brain functioning: a synergetic approach to brain activity, behavior and cognition*. [S.l.]: Springer Science & Business Media, 2013. v. 67. 23
- HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, 2009. v. 3. 51, 52
- HOFFMANN, U. et al. An efficient p300-based brain–computer interface for disabled subjects. *Journal of Neuroscience methods*, Amsterdam, v. 167, n. 1, p. 115–125, 2008. 31
- HUSTER, R. J. et al. Brain–computer interfaces for eeg neurofeedback: Peculiarities and solutions. *International journal of psychophysiology*, Amsterdam, v. 91, n. 1, p. 36–45, 2014. 30
- IBRAHIM, B.; SHERWANI, F. Brain computer interface based functional electrical stimulation: An outline. In: FUNCTIONAL ELECTRICAL STIMULATION SOCIETY ANNUAL CONFERENCE (IFESS), 19., 2014. *Proceedings...* [S.l.]: IEEE, 2014. p. 1–5. 14, 34
- INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Censo demográfico 2010*. Brasília, 2010. 14
- ISCAN, Z.; DOKUR, Z.; DEMIRALP, T. Classification of electroencephalogram signals with combined time and frequency features. *Expert Systems with Applications*, Kidlington, v. 38, n. 8, p. 10499–10505, 2011. 51
- IVANCEVIC, V. G.; IVANCEVIC, T. T. Brain and classical neural networks. *Dordrecht*, Springer, 2010. 48, 49
- JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, Piscataway, v. 22, n. 1, p. 4–37, 2000. 36
- JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: a tutorial. *Computer*, Piscataway, v. 29, n. 3, p. 31–44, 1996. 51
- KANDEL, E. et al. *Principles of neural science*. New York: McGraw-hill, 2014. v. 4. 17, 19, 20, 25
- KAPPENMAN, E. S.; LUCK, S. J. The effects of electrode impedance on data quality and statistical significance in erp recordings. *Psychophysiology*, Hoboken, v. 47, n. 5, p. 888–904, 2010. 38
- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1995. *Proceedings...* Stanford, 1995. p. 1137–1145. 38
- KOŁODZIEJ, M.; MAJKOWSKI, A.; RAK, R. J. A new method of spatial filters design for brain-computer interface based on steady state visually evoked potentials. In: INTERNATIONAL CONFERENCE ON INTELLIGENT DATA ACQUISITION AND ADVANCED COMPUTING SYSTEMS: TECHNOLOGY AND APPLICATIONS (IDAACS), 8., 2015, Warsaw. *Proceedings...* Piscataway: IEEE, 2015. v. 2, n. 697-700. 27, 47, 48, 56, 58

- KÜBLER, A. et al. Patients with als can use sensorimotor rhythms to operate a brain-computer interface. *Neurology*, Philadelphia, v. 64, n. 10, p. 1775–1777, 2005. 30
- KWAK, N.-S.; MÜLLER, K.-R.; LEE, S.-W. A lower limb exoskeleton control system based on steady state visual evoked potentials. *Journal of neural engineering*, Bristol, v. 12, n. 5, p. 056009, 2015. 51
- LANCIONI, G. et al. *Assistive technology: interventions for individuals with severe/profound and multiple disabilities*. [S.l.]: Springer Science & Business Media, 2012. 14
- LEITE, S. N. d. C. *Contribuições ao desenvolvimento de interfaces cérebro-computador baseadas em potenciais evocados visualmente em regime estacionário*. 2016. 181 f. Tese (PhD in Electrical Engineering) — Universidade Estadual de Campinas, Campinas, 2016. 44
- LIU, Q. et al. Recent development of signal processing algorithms for ssvep-based brain computer interfaces. *Journal of Medical and Biological Engineering*, Heidelberg, v. 34, n. 4, p. 299–309, 2014. 47, 48
- MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, Piscataway, v. 11, n. 7, p. 674–693, 1989. 48
- MILTON, J. S.; ARNOLD, J. C. *Introduction to probability and statistics: principles and applications for engineering and the computing sciences*. [S.l.]: McGraw-Hill, 2002. 41
- MITCHELL, T. M. et al. *Machine learning*. WCB. [S.l.]: Boston, 1997. 35, 36
- MONK, S. *Programming the Raspberry Pi: getting started with python*. [S.l.]: McGraw Hill Professional, 2015. 67
- MULLER-PUTZ, G. R.; PFURTSCHELLER, G. Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on Biomedical Engineering*, Piscataway, v. 55, n. 1, p. 361–364, 2008. 34
- MÜLLER, S. M. T.; BASTOS-FILHO, T. o. F.; SARCINELLI-FILHO, M. Using a SSVEP-BCI to command a robotic wheelchair. In: INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, 20., 2011, Gdansk. *Proceedings...* [S.l.]: IEEE, 2011. p. 957–962. 34
- MUNOZ, A. *Machine Learning and Optimization*. 2014. Available in: <https://www.cims.nyu.edu/munoz/files/ml_optimization.pdf>. Access in: 02 Mar. 2016. 35
- MURGUIALDAY, A. R. et al. Haptic brain-computer-interfaces (BCI) in stroke. *Biomedical Engineering China*, Abingdon, v. 51, n. Supplement, p. M–139, 2013. 30
- NICOLAS-ALONSO, L. F.; GOMEZ-GIL, J. Brain computer interfaces, a review. *Sensors*, Basel, v. 12, n. 2, p. 1211–1279, 2012. 21, 22, 24, 30, 31, 32
- NIJBOER, F. et al. A P300-based brain-computer interface for people with amyotrophic lateral sclerosis. *Clinical neurophysiology*, Amsterdam, v. 119, n. 8, p. 1909–1916, 2008. 31

- NUMPY, D. *Numpy Official Website*. 2016. Available in: <https://www.numpy.org>. Accessed in: 03 Feb. 2017. 56
- OCAK, H. Automatic detection of epileptic seizures in eeg using discrete wavelet transform and approximate entropy. *Expert Systems with Applications*, Kidlington, v. 36, n. 2, p. 2027–2036, 2009. 47, 49
- OIKONOMOU, V. P. et al. *Comparative evaluation of state-of-the-art algorithms for SSVEP-based BCIs*. 2016. Available in: <https://arxiv.org/abs/1602.00904>. Access in: 03 Mar. 2016. 15, 39, 44, 47, 48, 50, 51
- OPPENHEIM, A. V.; SCHAFER, R. W.; BACK, H. R. *Discrete time signal processing*. São Paulo: [s.n.], 2015. 40, 42, 43, 44, 45, 46, 47
- PEDREGOSA, F. et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, Cambridge, v. 12, p. 2825–2830, 2011. 51, 56
- PFURTSCHELLER, G. et al. Current trends in GRAZ brain-computer interface (BCI) research. *IEEE Transactions on Rehabilitation Engineering*, Piscataway, v. 8, n. 2, p. 216–219, 2000. 30
- PINTO, M. A. da S. *Estudo do potencial evocado visual em regime permanente baseado em LED para interface cérebro máquina*. 2011. 190 f. Tese (PhD in Electrical Engineering) — Universidade Federal de Minas Gerais, Belo Horizonte, 2011. 29, 32, 53
- PYBYTES, D. *PyBytes Official Website*. 2017. Available in: <https://wavelets.pybytes.com/wavelet/db12/>. Access in: 03 Mar. 2017. 63
- PYWAVELETS, D. *pyWavelets Official Website*. 2017. Available in: <https://pywavelets.readthedocs.io/en/latest/>. Access in: 03 Mar. 2017. 56
- SCIPY, D. *Scipy Official Website*. 2017. Available in <https://www.scipy.org/about.html>. Accessed in: 03 Mar. 2017. 56
- SELLERS, E. W.; DONCHIN, E. A P300-based brain-computer interface: initial tests by als patients. *Clinical neurophysiology*, Shannon, v. 117, n. 3, p. 538–548, 2006. 31
- SILVA, R. L. *Desenvolvimento de uma interface homem-máquina aplicada a uma cadeira de rodas robótica por meio de PDA*. 2007. 146 f. Dissertação (Master in Electrical Engineering) — Universidade Federal do Espírito Santo, Vitória, 2007. 34
- STEYRL, D. et al. Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier. 2016. *Biomedical Engineering/Biomedizinische Technik*, Berlin, v. 61, n. 1, p. 77–86, 2016. 52
- SUGIARTO, I.; ALLISON, B.; GRÄSER, A. Optimization strategy for ssvep-based bci in spelling program application. In: INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND TECHNOLOGY, 2009. *Proceedings...* Cairo: IEEE, 2009. p. 223–226. 34
- TAN, D.; NIJHOLT, A. *Brain-computer interfaces and human-computer interaction*. [S.l.]: Springer, 2010. 22, 23

- TANNUS, A. M. *Controle de uma cadeira de rodas utilizando potencial evocado visual*. 2014. 92 p. Dissertação (Master in Electrical Engineering) — Universidade Federal de Minas Gerais, Belo Horizonte, 2014. 14, 27, 34
- TELLO, R. J. M. G. et al. Comparison of the influence of stimuli color on steady-state visual evoked potentials. *Research on Biomedical Engineering*, London, v. 31, n. 3, p. 218–231, 2015. 33
- TEPLAN, M. et al. Fundamentals of eeg measurement. *Measurement science review*, Warsaw, v. 2, n. 2, p. 1–11, 2002. 40
- THORNTON, C. et al. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 19., 2013. *Proceedings...* Chicago: ACM, 2013. 36
- VOLOSYAK, I. et al. Brain–computer interface using water-based electrodes. *Journal of neural engineering*, Bristol, v. 7, n. 6, p. 066007, dec 2010. 15
- WANG, Y. et al. Brain-computer interfaces based on visual evoked potentials. *IEEE Engineering in medicine and biology magazine*, Piscataway, v. 27, n. 5, 2008. 33
- WEBSTER, J. G.; HENDEE, W. R. *Encyclopedia of medical devices and instrumentation*. [S.l.]: AIP, 2006. v. 3. 25, 26, 27, 39
- WOISKI, E. R. Data analysis in python: anonymized features and imbalanced data target. In: *Probabilistic prognostics and health management of energy systems*. [S.l.]: Springer, 2017. p. 169–188. 35
- WOLPAW, J. R. et al. Brain-computer interface technology: a review of the first international meeting. *Transactions on rehabilitation engineering*, Piscataway, v. 8, n. 2, p. 164–173, 2000. 15
- WOLPAW, J. R. et al. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, Shannon, v. 113, n. 6, p. 767–791, 2002. 15, 29, 30, 31
- WOOD, G. et al. On the need to better specify the concept of “control” in brain-computer-interfaces/neurofeedback research. *Frontiers in systems neuroscience*, Lausanne, v. 8, p. 171, 2014. 30
- WORLD HEALTH ORGANIZATION. *World report on disability*. Switzerland, 2011. 350 p. 14
- XU, M. et al. A hybrid bci speller paradigm combining p300 potential and the ssvep blocking feature. *Journal of neural engineering*, Bristol, v. 10, n. 2, p. 026001, 2013. 34
- YI, W.-J. et al. Design flow of wearable heart monitoring and fall detection system using wireless intelligent personal communication node. In: INTERNATIONAL CONFERENCE ON ELECTRO/INFORMATION TECHNOLOGY (EIT), 2015, Dekalb. *Proceedings...* Piscataway: IEEE, 2015. p. 314–319. 66
- YIN, E. et al. A novel hybrid bci speller based on the incorporation of ssvep into the p300 paradigm. *Journal of neural engineering*, Institute of Physics, Bristol, v. 10, n. 2, p. 026012, 2013. 31