UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

Design of a BMI for tetraplegic patients

Dissertation presented by
Matthieu FONTAINE , William CHERMANNE

for obtaining the Master's degree in
Electro-mechanical Engineering

Supervisor(s)
Jean-Didier LEGAT, Pascal DOGUET

Reader(s)
Maxime SCHRAMME, Ludovic MOREAU

Academic year 2017-2018

Abstract

A Brain-Computer Interface (BCI) is a system that translates a user's brain activity into messages or commands for an interactive application. Such a system offers the possibility to take control of a device by thought. Therefore this type of system is totally adapted to help people with physical disabilities. This is the case of tetraplegic patients, who have lost the ability to move their limbs. BCIs represent a relatively recent technology that is experiencing a rapid growth due to recent advances in electronics and signal processing. In this context, this Master Thesis aims at exploring the field of BCI and develop a system that would be able to detect imaginary movements.

This work continues the those done previously by implementing a complete BCI application in real-time on a FPGA, able to collect data and tell if a movement of the right arm, left arm or not movement at all has been done. To achieve this goal, different steps have been completed, each one corresponding to a chapter of this work.

The field of BCI is a complex area of study and we needed to be up to date with the trends and applications up to now. This work thus starts with a state of the art and review of the anatomy of the brain and different brain signals. It also continues with the exploration of the different existing algorithms and techniques for the implementation of such a system. An experimental board received from a previous work is used to collect signals from the brain with a 3D printed helmet where 5 electrodes are fastened on. These electrodes are in contact with the sensory-motor areas of the brain. An algorithm exploration tool is created with `MATLAB` and helps us find the combination of algorithms giving the best classification results, ie. detecting the correct movement. Then a real-time application is developed and proves to be functional by offering good results, with a real-time classification accuracy of 95% in ideal conditions and other remarkable performances.

Acknowledgments

First of all, we would like to thank our Master Thesis supervisors Pascal Doguet and Jean-Didier Legat for giving us the chance to work on this very interesting subject, but also for their support during the entire period of this work. The field of Brain Machine Interface is extremely captivating and we really enjoyed working on it during an entire year.

Many thanks to Maxime Schramme, who has always been available to answer our questions. We really appreciated every time we had the chance to have his help and advices. We are very thankful for his support. We also would like to thank Ludovic Moreau for his support regarding the FPGA programming.

Finally, we would like to thank our respective family for their encouragements during the entire time of our work. They helped us correct our thesis to make it accessible to as much people as possible. A big thank you to all our friends to keep us motivated during the year.

Contents

Introduction **15**

1 A new hope for tetraplegic patients **18**
1.1 What is a Brain Machine Interface ? . . . 18
1.1.1 General description . . . 18
1.1.2 Different categories of BMIs . . . 20
1.2 What is tetraplegia ? . . . 21
1.2.1 What causes tetraplegia ? . . . 21
1.2.2 What are the consequences ? . . . 21
1.2.3 How to treat Tetraplegia ? . . . 22
1.2.4 Brain machine interface for tetraplegic patient ? . . . 22
1.3 A bit of brain anatomy . . . 22
1.3.1 General description . . . 22
1.3.2 The cerebral cortex . . . 23
1.3.3 Motor and sensory areas of the cerebral cortex . . . 24
1.4 Brain electrical signals measured noninvasively . . . 26
1.4.1 Spectrum of brain electrical signals . . . 26
1.4.2 Brain signals . . . 27
1.4.3 Brain signals artifacts . . . 39
1.5 Brain electrical signals measured invasively . . . 43
1.5.1 Local field potentials . . . 43
1.5.2 Single neuron activity . . . 45
1.6 Recording methods . . . 47
1.6.1 Electroencephalogram . . . 47
1.6.2 Electrocorticography . . . 50
1.6.3 Intracortical Neuron Recording . . . 50
1.7 General comparison of BCI systems. . . . 52
1.7.1 Active/Reactive BCI systems. . . . 53
1.7.2 Synchronous/Asynchronous BCI systems. . . . 53
1.7.3 Invasive/Noninvasive BCI systems. . . . 54
1.7.4 Signals types for BCI systems. . . . 54
1.7.5 Acquisition methods for BCI systems. . . . 56
1.8 General specifications and conclusion . . . 56

2 Processing chain **58**
2.1 Preprocessing . . . 58
2.1.1 Frequency filtering . . . 58
2.1.2 Spatial filtering . . . 60
2.2 Feature Extraction . . . 61
2.2.1 Feature properties . . . 62
2.2.2 Temporal methods . . . 62

2.2.3 Spectral methods . . . 62
2.2.4 Time frequency methods . . . 63
2.3 Classification . . . 68
2.3.1 Classification characteristics . . . 68
2.3.2 The classifier . . . 70
2.3.3 Classifiers major problems . . . 70
2.3.4 Linear Classifiers . . . 70
2.3.5 Nonlinear Bayesian classifiers . . . 71
2.3.6 Hidden Markov Model (HMM) . . . 72
2.3.7 Nearest Neighbors classifiers . . . 72
2.3.8 Neural Network . . . 72
2.3.9 Decision trees . . . 75
2.4 Conclusion . . . 76

3 BCI Hardware and some improvements 77
3.1 BCI board . . . 77
3.1.1 The board. . . . 78
3.1.2 Filter . . . 78
3.1.3 ADCs . . . 78
3.1.4 FPGA . . . 79
3.1.5 MCU . . . 79
3.1.6 Buttons, switches and LEDs . . . 79
3.2 Hardware improvements . . . 79
3.3 Number of electrodes . . . 82
3.4 Conclusion and BCI hardware specifications . . . 82

4 Algorithm exploration tool 84
4.1 Timing Specifications of the BCI . . . 85
4.2 Recording protocol for training database generation . . . 85
4.3 Introduction to the FieldTrip toolbox . . . 89
4.3.1 The toolbox . . . 89
4.3.2 The FieldTrip structure . . . 90
4.3.3 FieldTrip functions . . . 90
4.3.4 Conclusion . . . 92
4.4 Application : General Overview . . . 92
4.5 Application : EEG signals acquisition . . . 93
4.6 Application : Frequency and spatial filtering . . . 95
4.7 Application : Feature extraction . . . 98
4.8 Application : Feature classification . . . 99
4.9 Results and methods comparison . . . 100
4.9.1 Conditions and requirements . . . 100
4.9.2 Comparison tables . . . 101
4.10 Conclusion and BCI software specifications . . . 106

5 Real-time application 108
5.1 Specifications . . . 108
5.2 General overview . . . 109
5.3 Verilog implementation . . . 112
5.3.1 Analog-to-Digital conversion . . . 112
5.3.2 Filtering . . . 113
5.3.3 Softcore NIOS II/f . . . 116
5.4 C/C++ Implementation . . . 119

5.4.1 Filtered signals acquisition . . . 119
5.4.2 Downsampling . . . 119
5.4.3 Spatial Filtering . . . 121
5.4.4 Features extraction . . . 125
5.4.5 Feature Translation and Classification . . . 132
5.5 Performances . . . 138
5.5.1 Response time . . . 138
5.5.2 Power management . . . 141
5.6 Conclusion . . . 141

6 Results and discussion **142**
6.1 Performance Metrics . . . 142
6.1.1 Confusion matrix. . . . 143
6.1.2 Accuracy . . . 143
6.1.3 Precision . . . 144
6.1.4 Sensitivity . . . 144
6.1.5 Specificity . . . 144
6.1.6 F1 score . . . 144
6.1.7 Receiver Operating Characteristics . . . 145
6.2 Results and protocol . . . 146
6.3 Results in non-ideal conditions . . . 148
6.3.1 Variation of the BCI user . . . 148
6.3.2 Training database 'age' variation . . . 150
6.3.3 Training database protocol variation . . . 151
6.3.4 Environment variation . . . 155
6.4 Conclusion . . . 155

Conclusion **157**

List of Figures

1 Experimental system developed two years ago. 15

1.1 BMI principle. This corresponds to an alternative path for interactions between the brain (Central nervous system) and the outside world. 19
1.2 BMI Stages. 20
1.3 *Motor homuculus* map that shows brain areas related to different motor and sensory functions. 23
1.4 The cerebral cortex. 23
1.5 Top view of primary motor cortex and its different control areas. 25
1.6 Typical spectrum of brain electrical signals recorded from the scalp. 26
1.7 Cortical oscillations localization areas. 28
1.8 Event-related de-synchronization/re-synchronization wave shapes/patterns for image imagery. 30
1.9 Typical evolution of an event-related potential in case of motor task. 31
1.10 First stage and second stage in ERP : (1) Movement planning occurs before the movement onset and creates an ERS in the γ band. (2) An ERD occurs in both the γ and μ band. The β band is however subject to and ERS during the whole movement. 32
1.11 Third and fourth stage in ERP : (3) Movement ends resulting in a re-synchronization in the μ band while the β band power still increases. (4) Movement has ended since 100 [ms] and the β band power slowly decreases (ERD). Other bands are back in a rest state. 32
1.12 First stage and second stage in ERP for hand-right movement : (1) Movement planning creates an ERS in the γ band in the contralateral hemisphere. (2) An ERD occurs in the both γ and μ band and in both hemispheres. The β band is however subject to and ERS in the contralateral hemisphere during the whole movement. 33
1.13 Third and fourth stage in ERP for hand-right movement : (3) Movement ends resulting in a re-synchronization in the μ band located in the ipsilateral hemisphere while the β band power still increases in the contralateral side. (4) Movement has ended since 100 [ms] and the β band power slowly decreases (ERD) in the contralateral side. Other bands are back in a rest state. 34
1.14 P100 component occurs 100 [ms] after the visual stimulus. 35
1.15 Wave shape of a P300 signals. Notice the presence of P100 signal too. 37
1.16 Spatial localization of the P300a signals is across the frontal and temporal lobes. Spatial localization of the P300b signals is across the central area of frontal lobe and parietal lobe. 37
1.17 Wave shape of a SCP signal. 39
1.18 Comparison between clean a EEG brain signal and an EEG brain signal corrupted by 50 [Hz] signal artifacts due to AC power lines in Europe. 40
1.19 Comparison between a clean EEG brain signal and an EEG brain signal corrupted by EOG artifacts. 41

1.20 Comparison between a clean EEG brain signal and an EEG brain signal corrupted by eye-blink artifacts. . . . 41
1.21 Comparison between a clean EEG brain signal and an EEG brain signal corrupted by EMG artifacts. . . . 42
1.22 Comparison between a clean EEG brain signal and an EEG brain signal corrupted by respiration artifacts. . . . 42
1.23 Local field potentials signal patterns. . . . 44
1.24 Local field potentials signal localization within the motor cortex. . . . 44
1.25 Single neuron firing signals signal when a motor tasks occurs. . . . 46
1.26 EEG acquisition system with differential amplifier and reference eletrcode. . . . 48
1.27 Electrodes placement systems (black dots for the 10-20 system; black and gray dots for the 10-10 system and all electrodes for the 10-5 system). . . . 49
1.28 Three possibilities for the reference electrode location. . . . 49
1.29 Comparison between EEG and ECog recording localization. . . . 50
1.30 Structure of a micro-electrode array. One spike of the array is depicted in (a) and the entire array connected to the substrate in (b). . . . 51
1.31 Comparison between two types of microelectrode arrays. The first is designed based on MEMS fabrication techniques while the second one is designed with micro-wires. . . . 51
1.32 Lumped model of a microelectrode array that records activity of K neurons with $L+1$ electrodes. . . . 52
1.33 General specifications for the BCI. . . . 57

2.1 Four different types of digital filters and their output signals. Respectively, the low-pass filter, high-pass filter, notch filter and bandpass filter. . . . 59
2.2 Method to extract the band power from a signal. Figure taken from [1]. . . . 63
2.3 Time/frequency resolution of wavelet transform. Time/frequency resolution of short-time Fourier transform. . . . 64
2.4 Different wavelet types. Certain wavelets perform better than other under particular circumstances . . . 65
2.5 Shifting and scaling operations of the mother wavelet. . . . 67
2.6 Sub-band decomposition of DWT filterbank; $h[n]$ is the high-pass filter, $g[n]$ the low-pass filter. . . . 68
2.7 Classification scheme with the training and the test set. The classifier is a binary classifier with two possible output classes. . . . 70
2.8 LDA classifier . . . 71
2.9 SVM classifier . . . 71
2.10 Neural Network illustration with 4 layers. . . . 73
2.11 Single-layer perceptron with a unique output perceptron and a sigmoïd function as activation function. Sigmoïd activation function has co-domain between 0 and 1. 74
2.12 Decision tree for an example of weather classification. . . . 76

3.1 General overview of BCI board. . . . 78
3.2 *Ultra Cortex Mark III* . . . 80
3.3 *Ultra Cortex Mark IV* . . . 80
3.4 Comparison between the electrodes placement of the two different helmets. . . . 81
3.5 Two types of electrodes used in this work. . . . 81
3.6 Evolution of the classification rate with respect to the number of electrodes. . . . 82
3.7 *Ultra Cortex Mark IV* helmet with 5 electrodes connected : C3, Cz, C4, Cp1 and Cp2. . . . 83
3.8 Hardware specifications for the BCI . . . 83

4.1 Recording and processing timing specifications of the BCI real-time system. . . . 85
4.2 Event-related potentials for the two same tasks but with different initial conditions. **In red**, the given movement (right-hand here) has already been executed 10 times before the one related to this ERP (right-hand here). Habit effect is present. **In grey**, a different motor task (left-hand here) has been executed before this motor task (left-hand here). No habit effect is present. 87
4.3 The complete protocol is explained in this figure : for a set of 30 acquisition, 15 series of two movements are asked from the patient who trained the system. The user is fed with movement that he/she has to execute 0.5 [s] before the movement recording operation starts. 88
4.4 Protocol used for recording a right-hand movement. The left-hand movement is similar, but for the other arm while the rest state is equivalent to state (1) of this figure for the whole recording period. 89
4.5 Organization of a FieldTrip structure containing EEG data. 90
4.6 Typical pipeline using the FieldTrip functions. 92
4.7 General overview of the exploration tool. 93
4.8 Application tab to simulate or load the EEG data 94
4.9 Loaded EEG signals where a movement occurs after 2 seconds. 95
4.10 Preprocessing panel of the application : preprocessing consists in frequency and spatial filtering options. 96
4.11 Loaded real data with a bandpass filter isolating the μ band. 97
4.12 Loaded real data with a CAR spatial filtering. 98
4.13 Application tab with feature extraction options. 99
4.14 Windowing and framing concept. 99
4.15 Application tab with classification options. 100
4.16 Software specifications for the BCI real-time system. 107

5.1 Complete specifications (general, hardware and software). 108
5.2 Complete specifications (general, hardware and software). 109
5.3 General overview of the real-time processing chain. 110
5.4 Supervised learning process of the feed-forward MLP neural network. 112
5.5 SPI protocol is used for communiaction between ADCs and the FPGA. 112
5.6 Raw electrical signals measured for C3, Cz, C4, Cp1 and Cp2. 113
5.7 Direct-Form implementation of the bandpass filter. 114
5.8 FIR bandpass filter specifications and expected frequency response. 114
5.9 FIR bandpass filter magnitude and phase responses. 115
5.10 Direct-Form implementation of the Notch filter. 115
5.11 IIR Notch filter specifications and expected frequency response. 116
5.12 IIR Notch filter magnitude and phase responses. 116
5.13 Block scheme of the Qsys and Verilog design. 118
5.14 Block scheme implementation of a 2-fold downsampler. 120
5.15 2-fold downsampling of filtered C3 electrode data. 120
5.16 Common spectral pattern block scheme. It requires a training process for obtaining the weight matrix W. The output of CSP spatial filter is a low-variance signal for right-hand movement and a high-variance signal for right-hand movement. 122
5.17 Common spectral pattern results for a right-hand movement of 3 [s]. 124
5.18 Common spectral pattern results for a left-hand movement of 3 [s]. 125

5.19 Detailed block diagram of the features extraction stage. First, each row of the s matrix undergoes a DWT of of 5 levels using a Daubechies window leading to 5 detailed signals (D1 to D5). After this stage, 25 signals are obtained (5 detailed signals for each of the 5 electrodes signals coming from the spatial filter). Four quantities are extracted from each of these 25 signals : The variance, The maximum, The minimum and the Willison amplitude. 127
5.20 Bandwidth of each of 5 decomposition levels of the DWT. 127
5.21 Shape of U structure that contains 5 decomposition signals for each of the 5 s_i signals. 128
5.22 Structure of the discret wavelet transform : (1) wavelet tree, (2) D_{low} decomposition filter and (3) D_{high} decomposition filter. Both impulse responses contain 2 taps. This is a characteristic of 'db1' window. 130
5.23 DWT of a given s_1 coming from of the spatial filter stage. The decomposition performed on 5 levels progressively 'peels off' high frequency content. 131
5.24 2D projection of several entries (23, 81 and 91) of features vectors obtained with the whole processing chain described so far and using the training database of 1500 trials from Chapter 4. 131
5.25 Tansig activation function is defined on $\mathbb{R}$ and has, unlike the sigmoïd function, a co-domain equal to (-1,1). 132
5.26 Organization of the output vector. Each entry is the probability of a given motor task among the 3 motor tasks that can be recognized in this BCI design (right-hand movement, left-hand movement and rest state). 133
5.27 Organization of the MLP neural network used in this BCI design. 133
5.28 Confusion matrix obtained after the training process. The left matrix corresponds to the confusion matrix linked to the training data set ($N_1 = 90\%$ of N), the middle one is obtained with the test data set ($N_2 = 10\%$ of N) and the right one is the total confusion matrix of the whole training database of 1500 trials. 135
5.29 Evolution of the gradient of mean-squared error as a function of the number of iterations. 136
5.30 Evolution of the mean-squared error as a function of the number of iterations. . . 136
5.31 Evolution of the mean-squared error as a function of the number of iterations. . . 137
5.32 Timing sequence of the real-time BCI system 139
5.33 Timing performance of the real-time BCI system 139
5.34 Size partitioning of the total 50k logic elements. 140

6.1 Organization of a confusion matrix. Each entry C_{ij} corresponds to the number of trials classified in class i which actually belongs to class j. 143
6.2 Typical classification Receiver Operating Characteristics plot. 145
6.3 Confusion matrix for 5 assessments of the BCI real-time system. 147
6.4 ROC plot for 5 assessment of the BCI real-time system. 148
6.5 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system by a user who has not trained the system. Case 1. 149
6.6 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system by a user who has not trained the system. Case 2. 149
6.7 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system when a 2-weekds old database is used. 151
6.8 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 1. 152
6.9 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 3. 152

6.10 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 1 and assessed based on a sequence of movements similar to protocol 2. 153
6.11 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 3 and assessed based on a sequence of movements similar to protocol 2. 154
6.12 Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired in a given environment and assessed in another environment. 155

List of Tables

1.1 Types of inputs in the somatosensory cortex. . . . 25
1.2 Vectors of information coming from the somatosensory cortex. . . . 26
1.4 Advantages and drawbacks of using cortical oscillations as signals. . . . 28
1.3 Brain activity rhythms. . . . 29
1.5 Event Related Potential patterns. . . . 30
1.6 The different SMR rhythms and their frequency bands. . . . 31
1.7 Advantages and drawbacks of using ERPs as signals. . . . 34
1.8 Two types of VEP. . . . 35
1.9 3 stages of evoked potentials. . . . 35
1.10 Advantages and drawbacks of using VEPs as signals. . . . 36
1.11 Advantages and drawbacks of using P300 as signals. . . . 38
1.12 Different possible patterns of SCP. . . . 38
1.13 Advantages and drawbacks of using SCPs as signals. . . . 39
1.14 Brain electrical signals artifacts. . . . 43
1.15 Advantages and drawbacks of using LFP as signals. . . . 45
1.16 Advantages and drawbacks of using single neuron firing signals as signals. . . . 46
1.17 Advantages and drawbacks of wet and dry electrodes. . . . 49
1.18 Comparison of active/reactive BCI systems. . . . 53
1.19 Comparison of synchronous/asynchronous BCI systems. . . . 54
1.20 Comparison of invasive/noninvasive BCI systems. . . . 54
1.21 Comparison of 7 different brain signals for a BCI design. . . . 55
1.22 Comparison of 3 different acquisition methods for a BCI design. . . . 56

2.1 Comparison of 6 different wavelet types. . . . 66

4.1 Advantages and drawbacks of the different acquisition methods. . . . 86
4.2 Three possible protocol for recording signals in order to generate the training database. . . . 88
4.3 CAR . . . 102
4.4 LAP . . . 103
4.5 CSP . . . 104
4.6 MLP Neural network and filter order . . . 104
4.7 Wavelet Level and parameter to extract . . . 105

5.1 Two different processors. . . . 116
5.2 Comparison of memory usage for acquiring EEG signals with two different sampling rates. . . . 119
5.3 Successive decomposition level of each s_i signal generates 5 detailed signals that have different frequency ranges and bandwidths. Notice that each detailed signal also has a different length : $s1_{D1}$ has a length of 250 samples while $s1_{D5}$ has a length of 16 samples for example. . . . 128
5.4 On-chip RAM memory usage. . . . 141

5.5 Power management of the BCI board. 141

6.1 Summary of the software specifications. 146

List of abreviations

ADC ..... Analog to Digital Conversion

AR ..... Autoregressive

BCI ..... Brain Computer Interface

BMI ..... Brain Machine Interface

BPF ..... Bandpass Filter

CAR ..... Common Average Reference

CMRR ..... Common Mode Rejection Ratio

CNN ..... Convulutionnal Neural Network

CPU ..... Central Processing Unit

CSP ..... Common Spatial Pattern

CWT ..... Continuous Wavelet Transform

DWT ..... Discrete Wavelet Transform

EEG ..... Electroencephalogram

EOG ..... ElectroOcularGraphic

EMG ..... ElectroMyoGraphic

ERD ..... Event Related Desynchronization

ERP ..... Event Related Potential

ERS ..... Event Related Synchronization

FIR ..... Finite Impulse Response

FFT ..... Fast Fourier Transform

FPGA ..... Field Programmable Gate Array

FPR ..... False Positive Rate

HMM ..... Hidden Markov Model

HPF ..... High Pass Filter

ICA ..... Independent Component Analysis

IIR ..... Infinite Impulse Response

kNN ..... k Nearest Neighbors

LDA ..... Linear Discriminant Analysis

LPF ..... Low Pass Filter

LFP ..... Local Field Potential

MCU ..... Microcontroller Unit

MEG ..... Magnetoencephalography

MEP ..... Movement Evoked Potential

MLP ..... Multilayer Perceptron

MSE ..... Mean Squared Error

NN ..... Neural network

PCA ..... Principal Component Analysis

ROC ..... Receiver Operating Characteristics

RNN ..... Recurrent Neural Network

SCP ..... Slow Cortical Potential

SMR ..... Sensory Motor Rythm

SPI ..... Serial Peripheral Interface

SSVEP ..... Steady State Visually Evoked Potentials

SVM ..... Support Vector Machine

TPR ..... True Positive Rate

TVEP ..... Transiant Visually Evoked Potentials

VEP ..... Visually Evoked Potentials

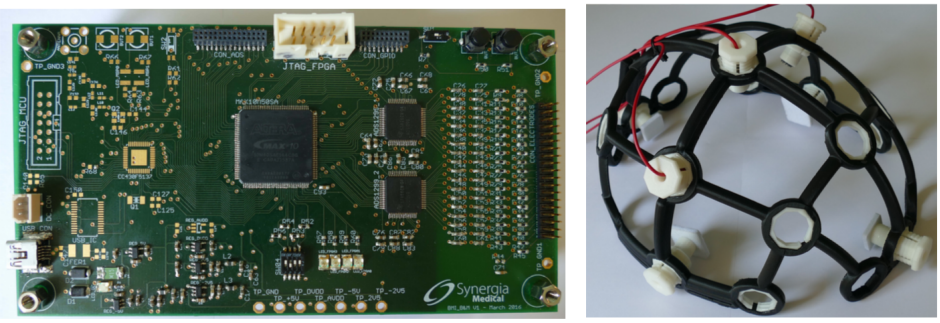
WAMP ..... Willison Amplitude

WT ..... Wavelet Transform

Introduction

Context

This Master Thesis is done in collaboration with the company *Synergia Medical*, based in Mont Saint-Guibert. *Synergia Medical* works in the field of innovative medical implants and aims at developing and commercializing next generation neurostimulation devices [13]. The company would like to explore the possibilities of developing a Brain Machine Interface device allowing tetraplegic patients to retrieve their abilities, i.e. allowing them to move their arms and/or legs by using their brain and thoughts to actuate an exoskeleton. A Brain Machine Interface, or BMI, is an interesting, active and highly interdisciplinary research topic at the interface between medicine, psychology, neurology, rehabilitation engineering, man-machine interaction, real-time electronics, machine learning and signal processing. In this context, a first Master Thesis subject was suggested in 2016 to start developing such a system. At the end of this first Master Thesis [3], an experimental system was developed. This system is made of an experimental board and a 3D printed helmet, holding the electrodes transmitting the neuronal activity to the board. This system is shown in Fig.1



(a) Experimental board.

(b) Helmet *Ultra Cortex Mark III* with the three electrodes C3,Cz,C4.

Figure 1: Experimental system developed two years ago.

In 2017, another group continued with this subject and explored some algorithms and techniques [4]. The group obtained some remarkable results by going further in the software aspects and algorithms implementation. The accuracy of their offline BMI was between 65% and 70%.

Objectives

This year, our work aims at continuing the work done previously and deepening the development of a BMI on the development board. The objectives were established at the beginning of our work and are listed here after:

1. Explore the state of the art about BMI, select the important and useful information for our work.

2. Use the material and hardware components developed in the previous years and try to improve it in different ways.

3. Implement a completely functional BMI application for offline analysis with `MATLAB`, using various algorithms and techniques. Then, explore the different algorithms combination in order to find the most promising results.

4. Implement a completely functional BMI in real-time using the FPGA and its integrated NIOS microprocessor mounted on the board developed in [3]. The BMI has to use a particular combination of algorithms among the ones explored earlier. After having built a strong database for the model training on `MATLAB`, the device should be able to detect when a movement of the arm occurs and distinguish left and right arm movements within a short time delay. This part thus includes

   – Offline classification model training.
   – Online filtering, processing and classification

5. Collect the results from both the offline analysis using `MATLAB` and the online analysis using the FPGA and analyze these results.

Structure

Each objective corresponds to, at least, one chapter of this document.

Chapter 1 is dedicated to the description of a BMI itself and the brain electrical signals that might used for a BMI design. Eventually a broad comparison between different BMI systems is performed in order to determine the general specifications of the BMI real-time system implemented in this work.

Chapter 2 is dedicated to the explanation of the state of the art methods and algorithms used in this field of study.

Chapter 3 focuses on the material changes made and the improvements it brought. Printing the next version of the helmet, *Ultra Cortex Mark IV*, enabled to place more surrounding electrodes in the region of interest. This improvement has brought some new possibilities.

Chapter 4 aims at explaining the different features of the `MATLAB` exploration tool that was created and why it has been useful in our work. It also presents a broad comparison of the different algorithms combinations. The most promising combination is retained for defining the software specifications of the BMI real-time system implemented in this work.

Chapter 5 describes the implementation of a BMI in real-time on the FPGA and its integrated NIOS microprocessor. The entire processing chain, arranged in modules, is explained and its performance is depicted. A particular focus on the time delay is done, as it is the critical point in real-time applications.

Chapter 6, one of the most important of this document, displays the results and performances of the BMI real-time system.

Finally, a conclusion will sum up the work that has been accomplished and open some tracks for further improvements.

Methodology

The methodology used in this Master Thesis consists in five stages :

1. First, the literature has been reviewed in order for the authors to become familiar with the BMI topics.
2. Then, the development of the `MATLAB` tool was done in parallel with the hardware improvements.
3. Next, the real-time BMI system has been programmed using the best processing chain combination found in the second stage and using the new material.
4. Afterwards, the real-time BMI system has been assessed in order to compare its performance with those given by the `MATLAB` tool.
5. Eventually this Master Thesis has been written in order to present the work that has been accomplished.

Results

After a lot of hard work and investment, the authors are proud to say that all objectives are fullfilled and that their work has brought very interesting results. Indeed,

1. We now have a clear and up to date vision of the current and future trends in the field of BMI design.
2. The material taken from the previous years was improved, using a new helmet and more electrodes for brain signals acquisition. This enables new possibilities.
3. The `MATLAB` tool is functional and allows to explore the most common and famous techniques used in the field of BMI design. It also helps to find the best algorithms for each step of the BCI design.
4. The real-time application is working fine, detecting left/right arm movement, and rest position with an accuracy between 75% and 95%. It also exhibits a short time delay between the movement and the detection, which is a critical point for real-time applications.
5. The results obtained through our work are satisfying and build a strong starting point for further improvements.

Chapter 1

A new hope for tetraplegic patients

This Chapter sets the context of our work by defining two important terms : Brain Machine Interface, also called BMI, and tetraplegia. This chapter is organized around seven directions. The first part describes the concept of BMI without giving information about performance of the different types of BMIs . The second part explains tetraplegia and how BMI could potentially help disabled people. Then, the third part gives a quick overview of the brain anatomy and the fourth part extensively describes bio-electric signals acquired from outside the brain, their origin, their influence and their potential use for BMIs. The fifth part describes bio-electric signals acquired from inside the brain and their potential use for BMIs. The sixth part presents different acquisition techniques in order to measure brain bio-electric signals. Eventually, the last part introduces a large comparison of different BMIs configurations. This comparison will influence further choices of design within this work.

1.1 What is a Brain Machine Interface ?

1.1.1 General description

A BMI is a device that measures signals from encephalic activity of its user. These signals are caused by an activity or an intent of activity from the user. By processing these signals, the BCI aims at controlling a computer or an actuator by making a prediction of the user's intention. There exist other definitions in the literature [1] [2]. Anyway, this is a system that allows a human to interact with its surrounding through an actuator controlled by brain signals detected using a huge variety of techniques. Notice that the brain encephalic activity can manifest itself trough four main physiological phenomena.

1. Brain bio-electrical signals created by currents that flow within billions of neuronal cells inside the brain. This is also known as electrophysiological activity and is due to ionic currents within the neurons and between them.

2. Brain magnetic signals generated by charged ions exited within neuron cells. This magnetic field is independent from the electrical field related to electrical potential.

3. Brain metabolic activities trough changes in blood flow within the brain.

4. Brain chemical activity. Numerous chemical reactions occur within the brain and within neurons themselves.

One will exclusively focus on brain electrical activity in this work. Therefore, the BMI designed in this work measures brain bio-electrical signals to predict the user intent. Several other articles [5] [6] deal with the other types of encephalic activity. Notice that, as explained in [12], a BMI can be used for various applications:

1. Replace functions lost due to injury or disease. This application corresponds to the theme of this work.
2. Restore lost functions (for example, stimulation of muscles of a paralyzed person and stimulation of nerves to restore bladder function).
3. Improve functions (for example, in stroke rehabilitation).
4. Enhance functions (for example, detection of stress levels or lapses of attention during demanding tasks).

Notice that a BMI does not require the intervention of peripheral nerves or muscle. It uses information from the central nervous system (CNS) only and is therefore suitable for people who suffer from neurological neuromuscular disorders.

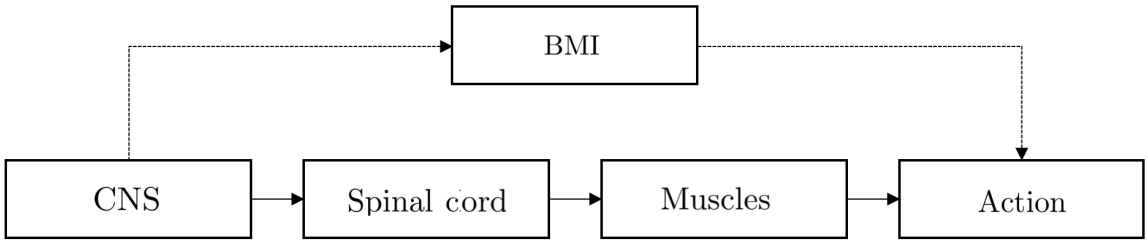


Figure 1.1: BMI principle. This corresponds to an alternative path for interactions between the brain (Central nervous system) and the outside world.

A BMI is usually a closed loop system composed of six main stages : signal recording, signal processing, feature extraction, feature classification, fetaure translation into a command and feedback. Fig.1.2 shows the complete chain constituting a BMI.

Signal recording

This part consists in acquiring raw signal from the brain. Numerous acquisition techniques exist although electroencephalogram still remains the most popular one. Note that several sensors can be placed at different locations on the head for measuring different brain activities.

Preprocessing

This part, also known as signal processing, comes just after the acquisition. The purpose of this stage is to clean up and remove noise and artifacts from measured signals. This stage is usually split into two substages: the frequency filtering and the spatial filtering. The output of this stage contains only the relevant information present in a desired frequency band.

Feature extraction

After the preprocessing stage, only relevant information remain in the signals. However, these signals are usually of large dimension, leading to an expensive classification stage in terms of computational power. Therefore, a dimensional reduction is required. A feature vector is in fact a low dimension vector that contains a sufficient amount of information called 'features' for an efficient classification.

Feature Classification

Once the low dimension feature vector is established, it is classified among different target classes. Each class corresponds to a possible command of the computer and/or the actuator.

Translation into a command

This corresponds to the last but one stage of the processing chain. The feature vector, containing relevant information about the raw signal recorded from the brain, belongs to a class assigned by the classifier. This class corresponds to a given command, which is sent to the actuator.

Feedback

A feedback is then provided to the user to inform him or her about the class that was recognized. Thanks to this feedback, the user may learn to modulate its brain activity for obtaining a better control of the BMI.

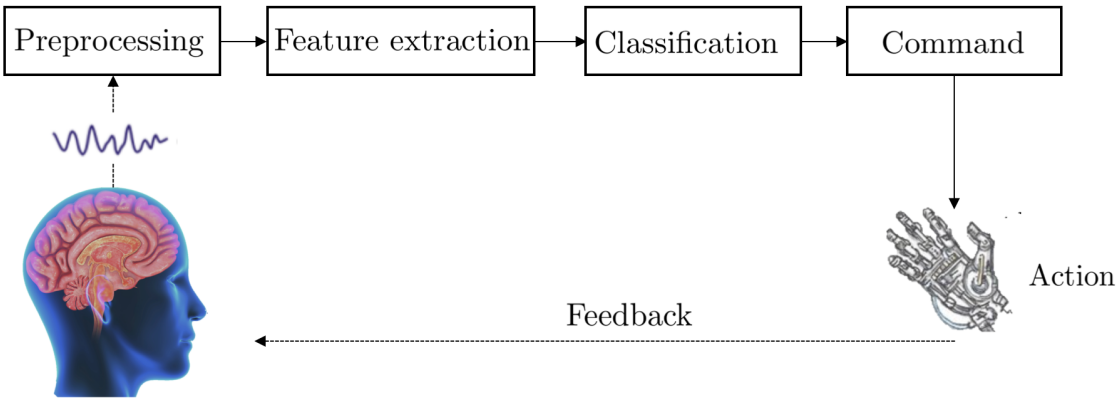


Figure 1.2: BMI Stages.

1.1.2 Different categories of BMIs

BMI may usually be characterized and classified according to their working principle. Either *active*, *reactive* or *passive*, either *synchronous* or *asynchronous*, either *dependent* or *independent*, either *invasive*, *non-invasive* or *hybrid*. This section describes the different types of BMIs.

Active/Reactive/Passive.

On the one hand, an *active* BMI is a device that requires effort from the user, by performing voluntary mental activity. For example, the user needs to think about a given movement for controlling the actuator. Note that such devices are also defined as endogenous. On the other hand, a *reactive* BMI measures the response of the brain to a given stimulus. These stimuli may be of different nature (sound, image, flashlight). Notice that such devices are also defined as exogenous. Eventually, a *passive* BMI measures the brain activity of the user without any actuator control objectives.

Synchronous/Asycnhronous.

A *synchronous* BMI analyzes the brain activity during a certain period of time and predicts the user's intent based on the signal recorded during this period. All signals that stand outside this recording period are not analyzed. Therefore, the user can send a command to the actuator only during specific time intervals. On the other side, *asynchronous* BMIs are continuously recording and analyzing the brain activity. The user can decide when to start and stop the mental task for controlling the actuator at any time.

Dependent/Independent.

A BMI is considered *independent* if it does not require any motor control to work. In that case, mental activity is only sufficient for predicting user's intention. If motor control is needed, then the BMI is said to be *dependent*. Notice that even an eye movement is considered as a motor task.

Invasive/Non-invasive/Hybrid

This category is related to the nature of recorded signals. Some signals can be acquired directly from the brain, or more precisely from the cortex. In that case, the BMI is *invasive* because it requires a surgical intervention to place sensors in the brain. A *non-invasive* BMI records signals from the head. One can mix invasive and non-invasive sensors to obtain an *hybrid* BMI. Notice also that classifying BMIs in these two categories is slightly restrictive. Indeed there are different levels of invasivity. Some BMIs record signal from the surface of the brain (low invasivity) while others record signals within the brain itself (high invasivity).

1.2 What is tetraplegia ?

Tetraplegia, also called quadriplegia, is a paralysis that affects all four limbs and torso. This paralysis may lead to a loss of sensitivity and/or a loss of mobility. These losses are either partial or total. In most cases, this paralysis is the consequence of an illness or an injury. This form of paralysis is the result of damage located in upper positions in the spinal cord, usually in the cervical spine between C1-C7. The higher the spinal cord is injured, the more extensive the damage will be.

1.2.1 What causes tetraplegia ?

A huge majority of tetraplegia cases are caused by a spinal cord injury. These injuries have several origins [7] :

1. Car accidents (37%)
2. Falls (30%)
3. Violence, including gunshots (28%)
4. Sports and recreational activities (9%)
5. Medical and surgical accidents (4%)

The population of tetraplegic patients grows up by 1200 new cases each year in France, for an estimated total population of 50 000 people in France.

1.2.2 What are the consequences ?

There are two types of consequences: physiological and psychological consequences.

Physiological consequences

The first consequence that one may notice is the loss of mobility of the limbs driven by the spinal cord below the injury location. Several disabled people also suffer of spasticity. This means uncontrolled and unpredictable movements that appear sporadically. One may remark that some difficulties with fertility and sexual function also occur. Eventually, an indirect consequence lies in the additional infections or illnesses related to the lack of movements.

Psychological consequences

From a psychological point of view, one may notice that disabled people are generally subject to an important depression related to their new condition that will not allow them to experiment the same life as they had prior. Several disabled people suffer also of anxiety regarding their future. Eventually, a social isolation occurs for most of people suffering this illness.

1.2.3 How to treat Tetraplegia ?

First, it is important to notice that quadriplegia is not curable with treatment. There is no way to reverse damage of the spinal cord. Nowadays, research is focusing on prostheses and medications for regenerating nerve cells. Meanwhile, current treatments consist in resistance training, physical therapy, occupational therapy or pain medications. These treatments have several purposes :

1. Help patients recovering partial motor functionality.
2. Reduce the patient pain.
3. Improve the long-term prognosis, leading to better life conditions for the patient.

1.2.4 Brain machine interface for tetraplegic patient ?

In the context of quadriplegia, BCI represent a new hope for quadriplegic people. Indeed, as described in Fig.1.1, a BCI can bypass the spinal cord path for controlling an exoskeleton using encephalitic activity. In most quadriplegia cases, only the spinal cord is damaged. This means that the brain is intact and still produces signals similar to non-disabled people. In that case, a BCI can interpret these signals and predict the user intent.

1.3 A bit of brain anatomy

This section describes some important concepts of brain anatomy that are useful for further understanding of bio-electrical signals that exist within the brain.

1.3.1 General description

Since the 19th century, one knows that different parts of the brain are devoted to different functions [1]. One can define a map of the *motor homunculus* which is a distorted representation of the human body based on a neurological map of the areas devoted to different parts of the body, either for motor or sensory functions. Fig.1.3 depicts that map. One can observe that lower limbs are associated to areas close to the junction between the 2 hemispheres while upper limbs areas are located closer to the temporal lobes.

The human brain consists of the two paired cerebral hemispheres. A cortex, also known as *gray matter*, covers both hemisphere. The cortex is characterized by a large number of neurons. Beneath the cortex, other parts present a large density of neurons. These parts are called *sub-cortical areas*. The interface between the cortex and the sub-cortical areas is the *white matter*. This consists of a network of nerve fibers.

Most of BMI research is focused on the cerebral cortex because it is the most accessible part of the brain given its localization just beneath the skull. On top of that, the cerebral cortex is responsible for movements which makes it attractive in BMI design for disabled patients. The next section will therefore exclusively focus on the cerebral cortex. More information about sub-cortical parts can be obtained in the literature [8] [9].

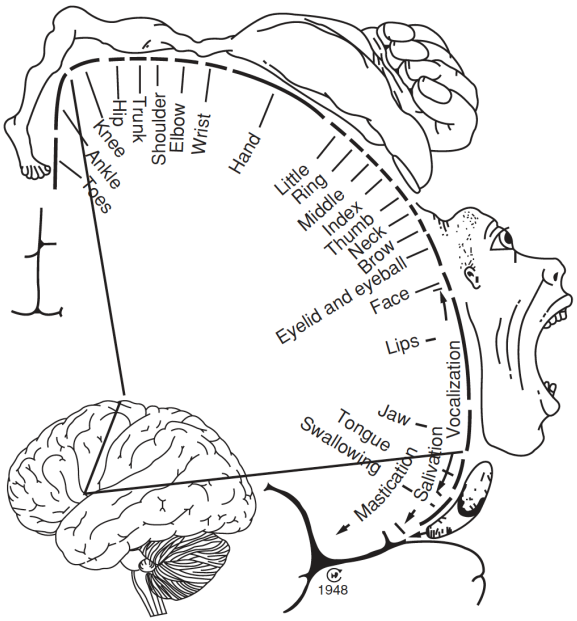


Figure 1.3: *Motor homuculus* map that shows brain areas related to different motor and sensory functions.

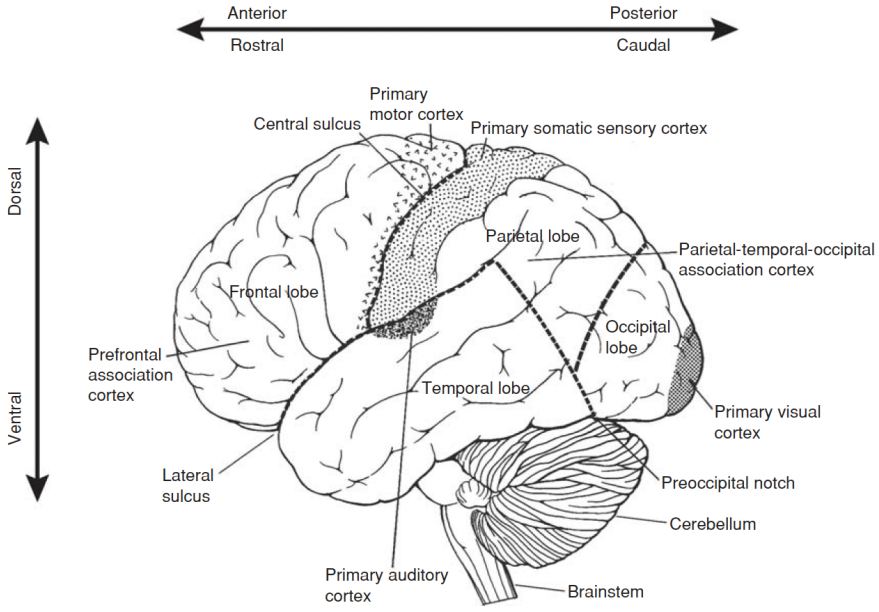


Figure 1.4: The cerebral cortex.

1.3.2 The cerebral cortex

The cerebral cortex, as shown in Fig.1.4, is made of four lobes : frontal, parietal, temporal and occipital. It is highly convoluted in order to have a large cortical volume while maintaining its thickness. On top of these lobes, there are several motor areas. Each area is useful for specific BMI applications.

1.3.2.1 Frontal lobe

It consists of the frontal and prefrontal areas. It is involved in higher-order cognitive function including complex reasoning, judgment and decision making. It is mainly involved in motor functions and is, for that reason, the most important for BMIs that aims at recognizing motor tasks.

1.3.2.2 Parietal lobe

It includes the primary somatosensory cortex and, in the posterior area, the multimodal association cortex that receives inputs from somatosensory, visual and auditory cortex. It also manages sensation, handwriting and body position.

1.3.2.3 Temporal lobe

It is located on both sides of the brain, close to the ears. This is why it is critical for auditory signal processing but it is also involved in higher-level visual processing and memory.

1.3.2.4 Occipital lobe

It is located at the posterior area of the brain and consists mainly of visual areas. It is made of numerous regions, and these have different specializations for different visual tasks, such as visuo-spatial processing, motion perception or color recognition.

1.3.3 Motor and sensory areas of the cerebral cortex

In case of a BMI design for tetraplegic patients, only motor and sensory parts of the brain interest the authors because they are involved in broken bio-electrical paths between the brain and the muscles but are still active even if the spinal cord is cut off. This section gives an introduction on principal sensory-motors areas of the brain.

1.3.3.1 Primary motor cortex

It has a close relation to movement control. It is possible to stimulate muscles directly form this area of the brain with an electrical stimulus because there exist several direct connections between this area and the motoneurons of the spinal cord. The primary motor cortex is organized in different sub-areas, where each of them is involved in the control of a particular part of the body. This organization can be seen in Fig.1.3. Legs and feet are close to the junction between hemispheres, also called the medial wall. Upper limbs such as arms and hands are progressively located more laterally within each hemisphere. Eventually, face muscles are located close to temporal lobes. Fig.1.5 depicts this organization. Notice also that some parts of body require more cortical areas because they are more complex muscles structures. For example, the face muscles occupies a larger cortical area than legs muscles. Muscles size is not related with associated cortical areas.

Input types	Tactile	Proprioceptive
Definition	The sensitive touch, also known as tactile sensation, is related to skin receptors located at different positions beneath the skin. They express the sensation of touching a surface with the skin. There exists different types of sensors with different sensitivities (maintained contact or punctual contact).	Proprioceptive sensors are related to the awareness of one's limbs position even if this person could not see them. For example, even with the eyes closed, one knows what is his/her arm position. Theses sensors are required for movement planning and guidance.

Table 1.1: Types of inputs in the somatosensory cortex.

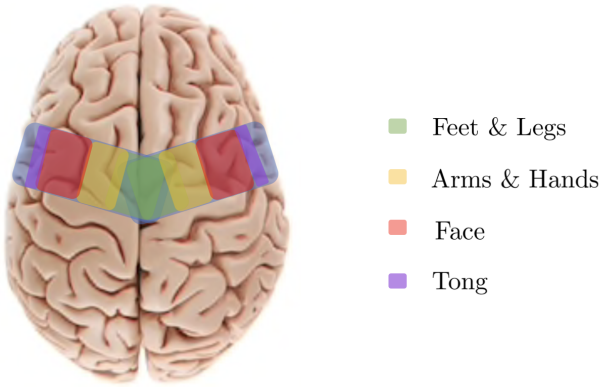


Figure 1.5: Top view of primary motor cortex and its different control areas.

Bearing that map in mind is important for the BMI design. Indeed, based on the knowledge of limbs position in the brain, one may position the acquisition system accurately depending on the brain activity of which limbs has to be recorded.

1.3.3.2 Premotor cortex

This part of the motor cerebral cortex is also located in the frontal lobe. It is anterior to the primary motor cortex. The functions of the premotor cortex are diverse but are not fully understood so far. It has a direct connection to the spinal cord (in the same way as the primary motor cortex) and may therefore stimulate muscles directly trough an electrical stimulus. Some researches [9] show that it is involved in movement planning and in the spatial guidance of movement.

1.3.3.3 Somatosensory cortex

This part is crucial for movement control because it processes all sensation information about different limbs of the body (temperature, touch sensation, limbs position in space). It is located in the most anterior part of the parietal lobe. It receives two types of inputs as depicted in table 1.1. Notice that all sensory information reach the somatosensory cortex through the thalamus.

1.3.3.4 Posterior parietal cortex

This part of the cerebral cortex is also involved in sensory function. Current research [10] finds out that this part is a multimodal association cortex. It means that it processes, associates

Visual streams	Dorsal	Ventral
Function	This path is related to the object-location (e.g. "where is the object ?"). It is also called the vision-for-action path.	This path is related to the object-recognition (e.g. "what is the object ?"). It is also called the vision-for-perception path.

Table 1.2: Vectors of information coming from the somatosensory cortex.

and combines data from different inputs (visual, auditory and somatosensory inputs). By combining these sensory inputs, the posterior parietal cortex creates a mapping of the limbs with respect of the external world. This mapping is used for movement coordination and guidance.

Notice that vision and proprioception coming from the somatosensory cortex are the two most important vectors of information for movement planning and guidance.

1.4 Brain electrical signals measured noninvasively

As a reminder, a BMI creates a communication and control system that bypasses the classic muscular output channels of the brain. To accurately predict the user intention, one must record brain signals associated with the planned movements. But what type of signal is useful for which application ? The purpose of this section is to describe first the typical spectrum of electroencephalogram electrical signals acquired from outside of the brain when a subject is at rest. Then, based on that spectrum, relevant **noninvasive** brain signals for a BMI application are presented and discussed. Eventually, artifacts that corrupt brain signals measurements when acquired from the scalp are described.

1.4.1 Spectrum of brain electrical signals

Considering a human being with no disability, one may establish the typical spectrum of its brain electrical signals in an aware state at rest, acquired from the scalp. Notice that this spectrum is established based on a real record of the brain electrical signals from the scalp of one of the authors using electroencephalogram technique. Fig.1.6 depicts that spectrum.

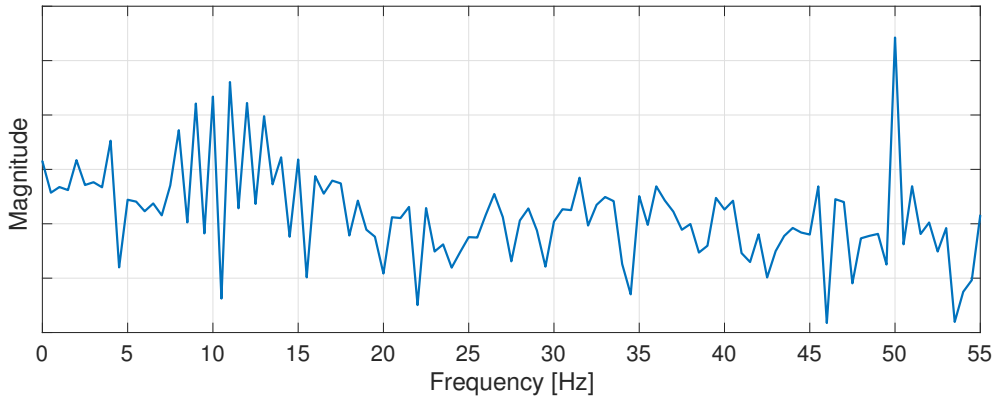


Figure 1.6: Typical spectrum of brain electrical signals recorded from the scalp.

One may observe several properties of the raw spectrum :

1. There is a continuous component at 0 [Hz] and low frequency components up to 4 [Hz]. These low frequency bins are associated with electromyographic effects (muscles contractions), eyes blinks and eye movements (electrooculargraphic effects). Several particular brain signals related to sleep occupy low frequencies too.

2. Frequency bins between 7 and 13 [Hz] present larger power than others. These frequency bins are associated with movements and reasoning (see later). Notice that the frequency bin with largest power lies generally between 10 and 12 [Hz].

3. From 13 [Hz] to 40 [Hz], a rule of thumb indicates (see [11]) that the power of these frequency bins decreases as $1/f$, with f being the frequency.

4. Frequency bins larger than 18-20 [Hz] are associated with complex mental cognition or intense focus.

5. There exists several artifacts in the recorded spectrum. The 50 [Hz] component is one of them and is caused by surrounding power lines. Eye blinking is also considered as an artifact in most BMI applications.

1.4.2 Brain signals

The last paragraphs illustrate the typical spectrum of brain activity. The question that arises now is : How does that spectrum vary when the patient thinks about a concept or an action ? One will study only the influence of an action, and particularly of a motor task in this work. This can be justified by the authors intention of designing a BMI for people who have lost their sensorymotor functions. Another work could study the variation of that spectrum when the patient thinks about a mathematical computation, for example. Five different signal types will be studied : Cortical oscillations, Event-related potentials, Visual evoked potentials, P300 signals and Slow Cortical Potentials.

1.4.2.1 Cortical oscillations

Brain activity is reflected in a large variety of oscillating signals, also known as rhythms. Each rhythm can be distinguished by its frequency range within the brain activity spectrum depicted in Fig.1.6. Moreover, these cortical oscillations are influenced by scalp location, and correlations with ongoing brain activity such as event-related potential (described in the next paragraph) [36] [37]. Given that patients can learn to control such signals by a training process, they are interesting for a BCI design purpose. One will analyze cortical oscillations into three directions. First, the different rhythms are presented, then their occurrence period and their spatial localization.

Cortical oscillations are characterized by different rhythms. Table 1.3 depicts these rhythms.

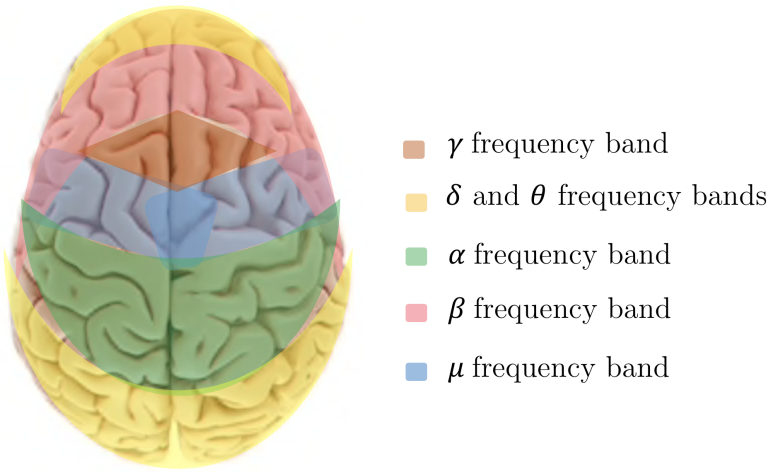


Figure 1.7: Cortical oscillations localization areas.

All these rhythms have different occurrence period. δ signals occur during the entire sleep period and continuously for babies. Regarding θ signals, they occur during the entire period of daydreaming. On the other hand, μ and β rhythms devoted to motor tasks are well localized in time and occur when a motor task occurs and last for a few seconds after the motor task end. Eventually, γ-rhythm signals occur continuously. Regarding the localization, one may record signals from each frequency band everywhere on the scalp. However these frequency bands have favorite areas, i.e. areas where the amplitude of these frequency bands is larger. This is depicted in Fig.1.7. One may notice that θ and δ frequency bands are localized both in the most anterior and posterior parts of the brain. Hopefully, these two frequency bands are not useful for BCI design purpose. μ and β bands are localized in the center of the brain, corresponding to the top of the scalp. Their proximity allows to record them together. To summarize, cortical oscillations are characterized by numerous different signals, which are characterized by several frequency bands and are related to several brain activities. Furthermore, patients can learn to control these signals, which is very interesting for a BCI design purpose. Using cortical oscillations as signals to be recorded for a BCI design has advantages and drawbacks, as listed in Table 1.4.

Advantages (+)	**Drawbacks (-)**
1. Except the recording material, no additional material is needed because this is used for **active** BCIs where the user performs an action 2. Can be used for both **actual** and **imagined** movements 3.Patients can learn to control some of these rhythms.	1. Signal-to-noise ratio of is poor due to the noninvasive acquisition. 2. Bandwidth is quite large (1 [Hz] to 60 [Hz]) but is not entirely usable for a BMI purpose, especially the δ and θ rhythms. One need to select some frequency bands.

Table 1.4: Advantages and drawbacks of using cortical oscillations as signals.

1.4.2.2 Event-related potentials

Event-related potentials (ERP) is a particular brain activity linked to the occurrence of a particular event (e.g. a hand movement). This event influences the electrical potential of the

Frequency bands names	**Frequency bands ranges**	**Comments**
δ	0-3.5 [Hz]	High amplitude but slowest wave. It is observed in adult sleeps and in babies. This rhythm is barely controllable and hence has few interests for BCI design.
θ	3.5-7.5 [Hz]	This is related to daydreaming or inefficiency. It represents a kind of border between awake state and sleep. Notice that this rhythm has a low amplitude in normal patients [37]. Not interesting for BCI design purpose.
α	7.5-12 [Hz]	This rhythm is related to relaxation or eye closing.
μ	8-12 [Hz]	This rhythm is related to motor tasks. **Notice that its frequency range overlaps with alpha range.** Highly interesting for BCI design purpose.
β	13-30 [Hz]	This rhythm is related to motor task, as well as the μ rhythm, but also to complex mental reasoning for solving mathematics problems for example. Interesting for BCI design purpose.
γ	31-60 [Hz]	It mainly reflects the mechanism of consciousness, although a part of the γ range is devoted to motor tasks. Slightly interesting for BCI design purpose.

Table 1.3: Brain activity rhythms.

scalp by an amplitude modulation on a particular frequency band during a certain time period and at particular spatial localization. One will analyze ERP following its four properties: the particular signal patterns, the frequency bands of these signals, the occurrence period and the spatial localization of these signals.

These ERPs are characterized by two patterns/signal wave shapes, as listed Table 1.5. In Fig.1.8, one can observe the two patterns. To summarize, an ERS corresponds to an increase of power in given frequency bins while an ERD corresponds to a decrease of power in given frequency bins.

ERP patterns	**De-synchronization**	**Re-synchronization**
Definition	This represents a de-synchronization in electrical oscillating signals that causes a decrease of power in a given frequency band. That frequency band depends on the event. This is also known as event-related de-synchronization (ERD).	This represents a re-synchronization in electrical oscillating signals that causes an increase of power in a given frequency band. That frequency band depends on the event. This is also known as event-related re-synchronization (ERS).

Table 1.5: Event Related Potential patterns.

By analyzing Fig.1.8 and looking in the literature [14] [15], one may notice that a typical ERD/ERS leads to a decrease/increase of power up to 20% with respect to the typical power of the brain electrical signals in rest state in case of motor imagery. In case of real motor tasks the power variations can reach up to 80%.

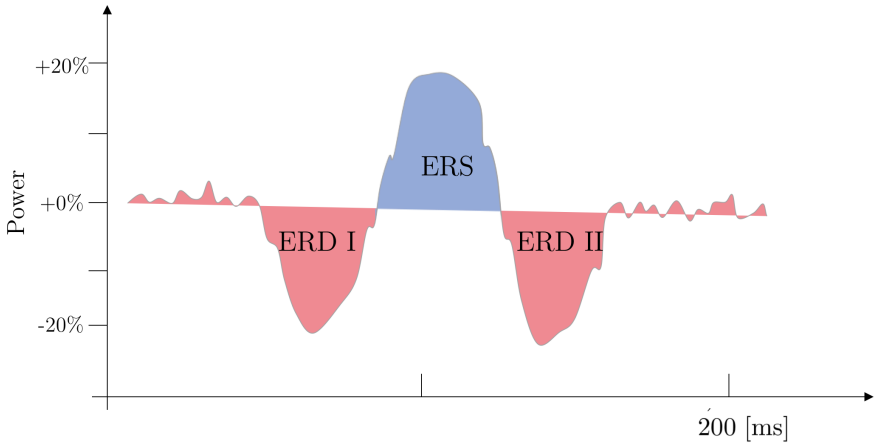


Figure 1.8: Event-related de-synchronization/re-synchronization wave shapes/patterns for image imagery.

Once the patterns are known, one needs to characterize the particular frequency ranges that are affected by these ERD/ERS. Notice that one will focus on **sensory-motor frequency bands** only, also known as **sensory-motor rhythm** (SMR). Such a choice is justified by the use of SMR exclusively for designing a Brain Machine Interface for patients who have lost their sensory-motor functions. The following table 1.6 summarizes the four main event-related rhythms/frequency bands that are involved during a motor task, i.e. **the SMR rhythms.**

Frequency bands names	Frequency bands ranges	Comments
Lower μ	8-10 [Hz]	The lower μ band is associated with the occurrence of any motor task (not topographically related).
Higher μ	10-13 [Hz]	The higher μ band is topographically related to a particular movement (i.e. such a rhythm manifests itself in a particular area for a given motor task. This area is devoted to this given motor task.)
β	13-30 [Hz]	A re-synchronization occurs in that band after a movement onset.
γ	38-42 [Hz]	A re-synchronization occurs in that band before a movement onset (during the movement planning).

Table 1.6: The different SMR rhythms and their frequency bands.

One may draw several conclusion based on the sensory-motor frequency bands :

1. Frequency bands are adjacent. This will require selective filters for processing different bands.
2. One may use the lower μ band for detecting a motor task but this is not sufficient for identifying a given movement because this band is involved in all motor tasks.
3. One may use a sampling frequency of at least 84 [Hz] for capturing all motor frequency bands while respecting Shannon theorem.

Once the patterns ERD/ERS and the frequency bands are known, one may characterize event-related potentials for a motor task by its occurrence time. First, one need to notice that the potential will vary before, during and after the motor task. Moreover this variation involves both patterns (ERD and ERS) and several frequency bands. Indeed a typical motor task gives the time evolution depicted in Fig 1.9. One can observe that the γ band is involved first in case of a motor task corresponding to the movement planning. Indeed an event-related synchronization occurs first around 40 [Hz]. This means that there is an increase in power for frequency bins around 40 [Hz].

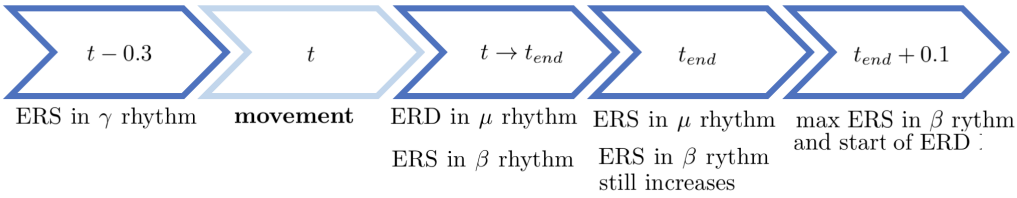


Figure 1.9: Typical evolution of an event-related potential in case of motor task.

In terms of spectrum, the movement planning, manifested by an ERS in the γ band, is depicted in Fig.1.10 (left figure). When the movement starts, an ERD occurs in both the γ and μ band.

The β band is however subject to and ERS. This is depicted in Fig.1.10 (right figure). When the movement ends, the μ band starts a re-synchronization while the β band is still under synchronization. This is depicted in Fig.1.11 (left figure). Eventually, the β band reaches its maximum power and starts to de-synchronize. This is depicted in Fig.1.11 (right figure).

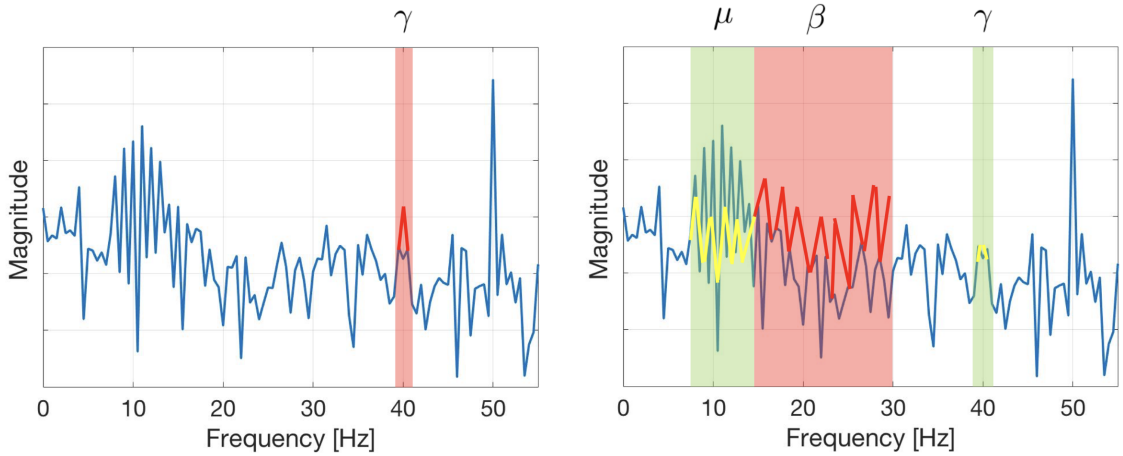


Figure 1.10: First stage and second stage in ERP : (1) Movement planning occurs before the movement onset and creates an ERS in the γ band. (2) An ERD occurs in both the γ and μ band. The β band is however subject to and ERS during the whole movement.

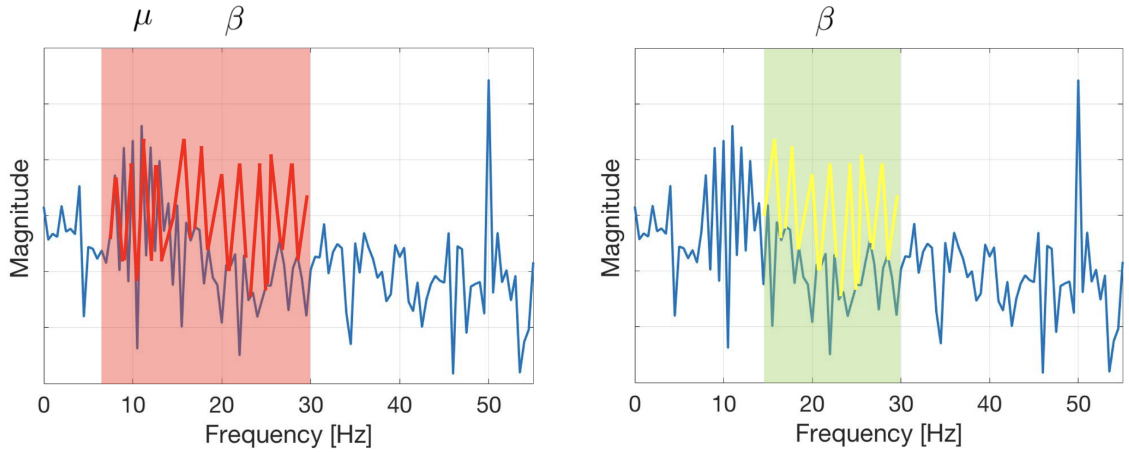


Figure 1.11: Third and fourth stage in ERP : (3) Movement ends resulting in a re-synchronization in the μ band while the β band power still increases. (4) Movement has ended since 100 [ms] and the β band power slowly decreases (ERD). Other bands are back in a rest state.

So far, most characteristics of an ERP are defined : its wave shape, its frequency bands and its different time occurrences around the event. However these ERPs also occur in different locations. Let's analyze the spatial properties of event-related potentials [16] [17] [18]. For that purpose one will use an example based on a right-hand movement. Please notice that a left-hand movement would produce the exact symmetric situation with respect to the *longitudinal fissure* (junction between the two hemispheres).

A right-hand movement is planned : An ERS occurs in the γ band in the **contralateral** side of the brains (i.e. the hemisphere opposite to the movement), in the area of the motor

cortex devoted to hand movements. This is depicted in Figure 1.12. When the movement starts, an ERD occurs in the γ band in the contralateral side of the brain over the motor cortex. An other ERD occurs in the μ band, in the **ipsilateral** hemisphere (i.e. the same hemisphere as the movement). The β band is however subject to and ERS in the contralateral side. Notice that this ERS impacts a larger area than the ERS in the γ band [18]. This is depicted in Fig.1.12.

Next the μ band goes back to its rest potential in the ipsilateral hemisphere. This is depicted in Fig.1.13. Regarding the contralateral side, one must notice that β signals still increases when the movement ends. It eventually reaches a maximum value (close to +20% of power) about 100 [ms] after the movement offset. These signals are more and more diffused in the contralateral hemisphere. The maximum area occupied by such signals occurs about 100 [ms] after the movement offset too. This is depicted in Fig.1.13.

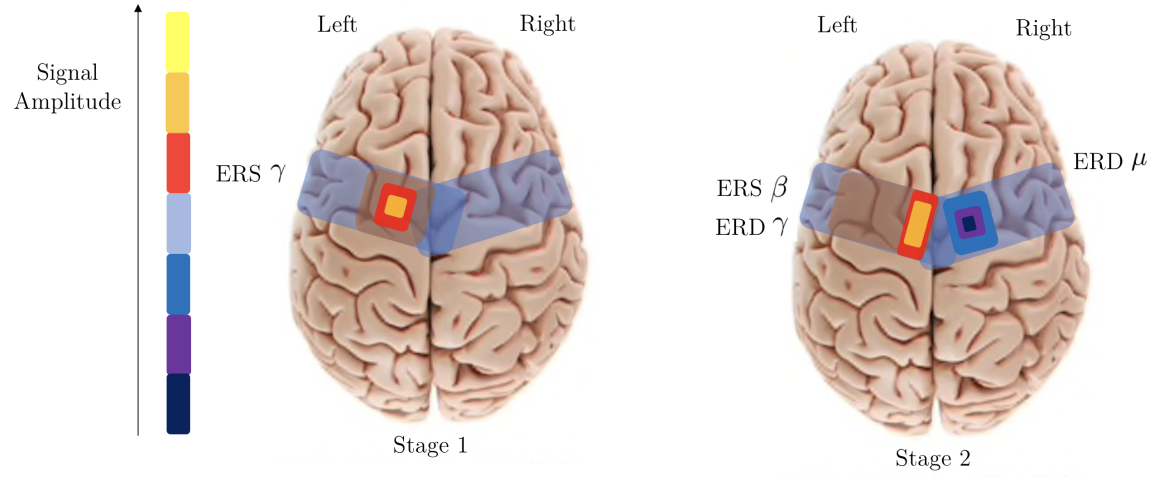


Figure 1.12: First stage and second stage in ERP for hand-right movement : (1) Movement planning creates an ERS in the γ band in the contralateral hemisphere. (2) An ERD occurs in the both γ and μ band and in both hemispheres. The β band is however subject to and ERS in the contralateral hemisphere during the whole movement.

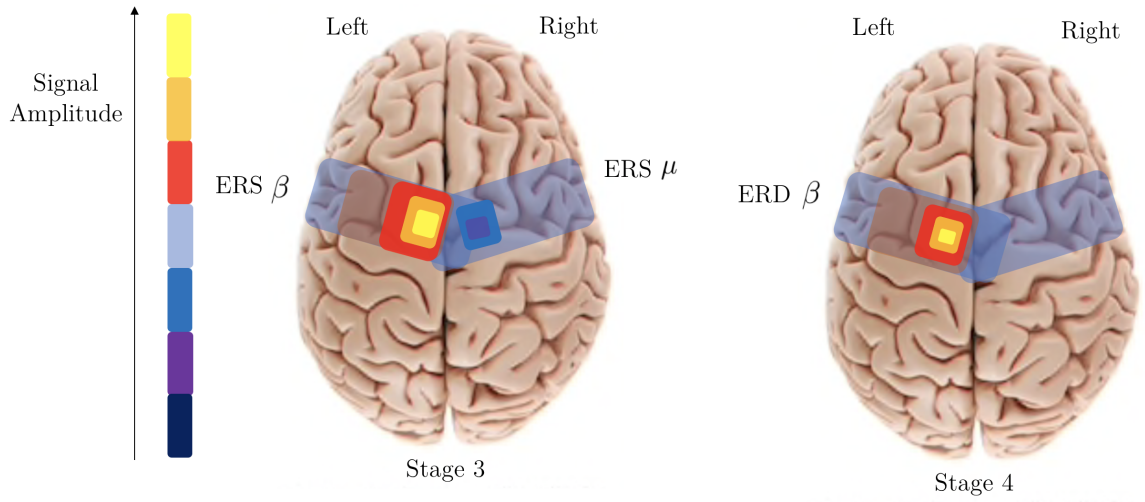


Figure 1.13: Third and fourth stage in ERP for hand-right movement : (3) Movement ends resulting in a re-synchronization in the μ band located in the ipsilateral hemisphere while the β band power still increases in the contralateral side. (4) Movement has ended since 100 [ms] and the β band power slowly decreases (ERD) in the contralateral side. Other bands are back in a rest state.

To summarize, event-related potentials, or ERPs, correspond to a modification of the brain activity when an event occurs. This event is assumed to be a motor-task in this work. ERPs have been presented here following 4 properties : the signal wave shape, the frequency bands of the signal, the occurrence period of the signal and the spatial localization of the signal. Using ERPs as signals to be recorded for a BMI design has advantages and drawbacks, as listed in Table 1.7.

Advantages (+)	**Drawbacks (-)**
1. Except the recording material, no additional material is needed because this is used for **active** BMIs where the users performs/thinks an action 2. Can be used for both **actual** and **imagined** movements 3. These signals are well-localized in time. There is no influence of a movement a long time after the end of that movement.	1. Signal-to-noise ratio of ERP is poor, especially when it's acquired form outside the brain.

Table 1.7: Advantages and drawbacks of using ERPs as signals.

1.4.2.3 Visual-evoked potentials.

A visually-evoked potential (VEP) reflects the visual information-processing mechanism of the brain. A visual stimulus is presented to a subject and the brain activity in reaction to that visual stimulus is recorded. VEPs are sometimes seen as a particular cases of ERPs where the event is external to the subject. It has been extensively studied and it is known that VEPs comprise at least three successive components that occur after the visual stimulus. These components vary according to the type of stimulus. This section will present the two possible types of visual

stimuli and typical components that occur after these stimuli.

One need to distinguish two types of VEPs. These two types are described in Table 1.8.

Transient VEP	**Steady-state VEP**
Also known as TVEP, transient VEPs occur for low stimulation rate. In other words, the frequency of the stimulation signal is low (below 6 [Hz]). In that case, evoked signals are transient signals. Analysis of such signals is generally done in time domain given its transient nature.	Also known as SSVEP, steady-state VEPs occurs for high stimulation rate. The frequency of the stimulation signal is larger than 6 [Hz]. Therefore, evoked potentials remain continuously (thus similarly as a steady state signal). Notice also that the continuous nature of the SSVEP requires the analysis to be done in the frequency domain.

Table 1.8: Two types of VEP.

After visual stimuli, evoked potentials occur on the cerebral cortex of the subject. These signals follow a particular pattern that can be decomposed in 3 stages, as explained in Table 1.9.

Stages	**Signal type**	**Occurrence time**	**Localization**
1st	Decrease in potential	70 [ms] after the visual stimulus	Primary visual cortex
2nd	Increase in potential	100 [ms] after the visual stimulus	Primary visual cortex
3rd	Back to initial potential	140 [ms] after the visual stimulus	Primary visual cortex

Table 1.9: 3 stages of evoked potentials.

Notice that this particular pattern is valid for SSVEP too. But in that case, the third stage occurs when the visual stimulus stops. The potential increase that happens after 100 [ms] is sometimes called the P100 component. Fig.1.14 depicts the successive stages that happen in TVEP.

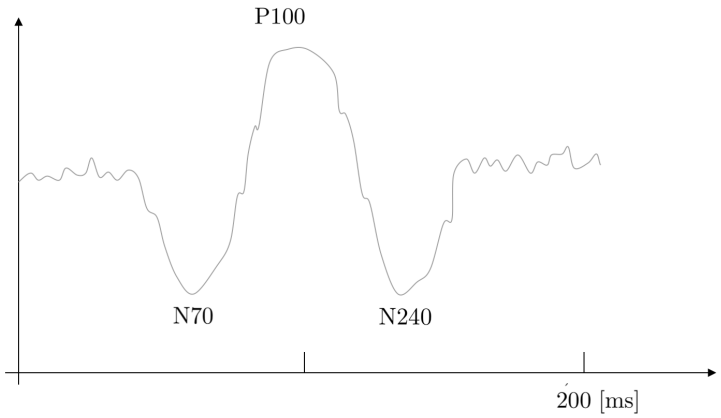


Figure 1.14: P100 component occurs 100 [ms] after the visual stimulus.

Notice that VEPs depend mainly on the properties of the visual stimulus. One may use this property for designing a BCI. For example, the user can produce different VEPs by looking at

different visual stimuli, each stimulus with different properties. These stimuli are, for example, flash lights with different rates. Evoked potentials will vary and will be measured to determine which flash light the user is looking at. For that reason, this kind of BMI is assumed to be dependent because it requires the eyes to move. To summarize, evoked potentials correspond to a special case of ERP where the stimulus comes from the external world. The visual stimulus creates a well-defined pattern where a peak of amplitude occurs in the recorded signal after 100 [ms]. This is called the P100. Using VEPs as signals to be recorded for a BMI design has advantages and drawbacks, as listed in Table 1.10.

Advantages (+)	**Drawbacks (-)**
1. The P100 component has a large amplitude compared to ERP signals and is thus recognizable. 2. Suitable for quadriplegic patients because it requires eye movements only. 3. These signals are well-localized in time. There is no influence of a movement a long time after the end of that movement. 4. Does not require much training from the user.	1. Material required for such a BMI. For each new BMI command, a new stimulation source is required. 2. Uncomfortable for the patient who is constantly subject to stimuli. 3. This leads to the design of dependent BMI.

Table 1.10: Advantages and drawbacks of using VEPs as signals.

1.4.2.4 P300 signals

In order to observe a P300 signal, one must have several repeating stimuli that are occuring continuously. Notice also that these stimuli may be of different natures : auditory, somatosenrory or visual. A P300 signal occurs when a patient is focusing on a given stimulus that occurs among other stimuli. In that case, the fact of focusing on a particular stimulus creates a potential increase 300 [ms] after its onset. The most used identification paradigm is called the **oddball** paradigm. Within this paradigm, a low-probability target item occurs simultaneously wit a set of high-probability non-target item [19]. The user needs to categorize each item. Only the attended item generates a P300 signal. This section describes important properties of P300 signals : its wave pattern, its occurrence time and eventually its location.

The wave shape of a P300 signal is characterized by a positive (this is the reason of P in P300) variation of electrical potential, i.e. an increases of potential. Its amplitude may vary depending on the intensity of the brain activity. Fig.1.15 depicts the wave shape of a P300 signal.

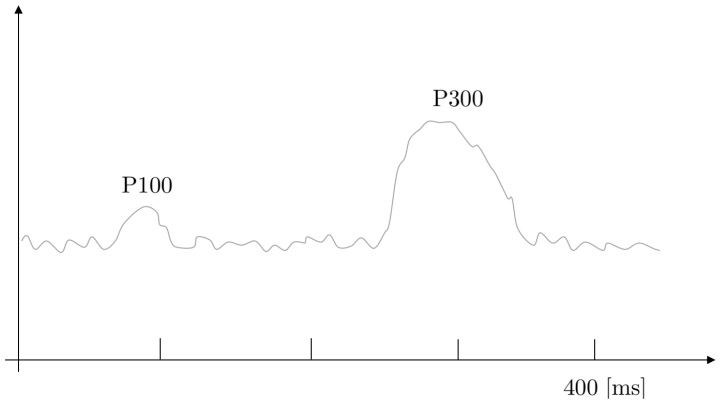


Figure 1.15: Wave shape of a P300 signals. Notice the presence of P100 signal too.

The occurrence period of the P300 signal is around 300 [ms] after the stimulus. However, one must distinguish two slightly different occurrence periods : P300a and P300b. The case of P300a occurs when the subject is under a new stimulus and its occurrence period is a faster than 300 [ms]. It is generally between 220 and 290 [ms]. It is related to the surprise effect. On the other hand, P300b signals occur when the subject faces a known stimulus that requires an action from the subject and therefore a memory effort. This type of P300 signal occurs generally between 310 and 380 [ms].

The spatial localization of P300 signals also depends on the type of P300 signals. Fig.1.16 depicts the spatial localization of both P300a and P300b. P300a signals appear mainly in the frontal and often in temporal lobes. P300b signals are localized in the posterior part of the brain, mainly across the brain central area and parietal lobe.

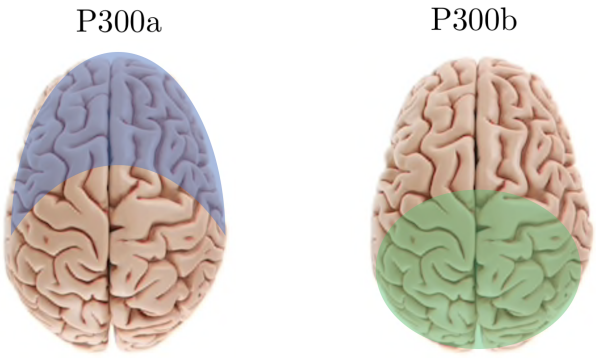


Figure 1.16: Spatial localization of the P300a signals is across the frontal and temporal lobes. Spatial localization of the P300b signals is across the central area of frontal lobe and parietal lobe.

To summarize, P300 signals may be seen as particular case of ERP signals where the patient focuses on one out of several external stimuli. That particular stimulus creates a positive variation of brain electrical signals after 300 [ms]. This positive variation reflects the state of stimuli evaluation and/or categorization. One need to distinguish P300a and P300b signals. Using P300 potentials as signals to be recorded for a BMI design has advantages and drawbacks, as listed in Table 1.11.

Advantages (+)	**Drawbacks (-)**
1. The P300 component has a large amplitude compared to ERP signals and is thus recognizable. 2. These signals are well-localized in time. There is no influence of a movement a long time after the end of that movement. 3. Does not require much training from the user. 4. Signals are well localized in space.	1. Material required for such a BMI. One need several sources of different stimuli (auditory, somatosensory, visual). 2. Uncomfortable for the patient who is constantly subject to stimuli.

Table 1.11: Advantages and drawbacks of using P300 as signals.

1.4.2.5 Slow cortical potentials

Slow cortical potentials, or SCPs, are the signals that correspond to the slowest varying electrical signals that can be recorded from outside the brain. They correspond to slow voltage changes generated by the cerebral cortex. For example, these voltage shifts may have a duration of about 5 to 10 [s]. These signals are not related to a particular motor task but rather to a phenomenon called *habituation effect* that leads to a voltage drift when a patient experiments any kind of situation during a long period such as moving a arm during a long time, keeping on thinking during a while or even experimenting pain during a few minutes [20] [21]. One will analyze SCPs according to its properties : the wave pattern, the occurrence period and time evolution and finally, the spatial localization of such signals.
The wave shape of slow cortical potentials (SCP) is characterized by two possible patterns, as described in Table 1.12.

Positive SCP	**Negative SCP**
1. A positive voltage shift occurs, i.e. there is an increase of potential. 2. Associated with mental preparation. 3. Reflects the increase in excitation of cortical neurons.	1. A negative voltage shift occurs, i.e. there is a decrease of potential. 2. Associated with mental inhibition. 3. Reflects the decrease in excitation of cortical neurons.

Table 1.12: Different possible patterns of SCP.

Notice that studies [22] have extensively proved that patients can learn to control SCP signals. This is done through a conditioning protocol. The patient can then use this ability to operate a BCI system. Notice that the amplitude of the voltage shift associated with the SCP is larger than for other signals such as ERPs or VEPs. This propety is quite interesting because it leads to an easier detection.

Fig.1.17 depicts the two wave patterns of slow cortical potentials. One may also observe that the voltage shift is observable between 3 and 10 [s]. This is the typical duration of a voltage shift related to a SCP. This makes this kind of signals much slower to detect than other signals presented earlier in this work. Notice also that some SCP signals can last for more than a minute [22].

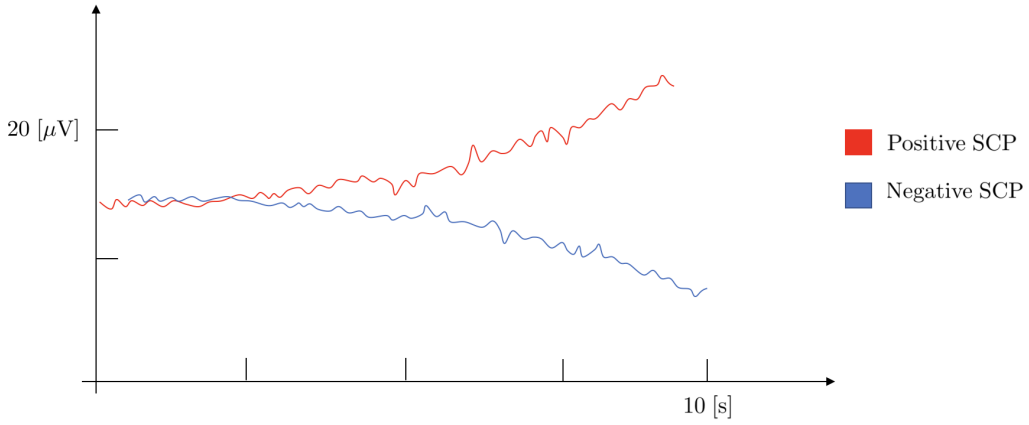


Figure 1.17: Wave shape of a SCP signal.

One can measure SCPs across the frontal lobe and the anterior parts of temporal lobes [22]. To summarize, SCPs are slow voltage drifts that occur when a situation is experimented by a user for a while. This situation could be for example the fact of rocking one's arm during 30 [s]. Notice that SCPs can appear superimposed to other bio-electrical signals and may have a much larger amplitude than other brain signals acquired from the scalp. Such signals are more commonly known as *habituation signals* and are critical for BMI design because they might cause a bias, which could alter or improve the performance of a BMI system. More details in chapter 4 and 5. Using SCP as signals to be recorded for a BMI design has advantages and drawbacks, as listed in Table 1.13.

Advantages (+)	**Drawbacks (-)**
1. Except the recording material, no additional material is needed because this is used for**active** BMIs where the users performs/thinks an action. 2. Can be used for both **actual** and **imagined** movements. 3. larger amplitude with respect to signals presented so far, leading to a better SNR. Signals are well localized in space.	1. There is only one drawback for SCP, but it is important : signals are not localized in time. They sometimes may occur a few minutes after an event. These signals are therefore hardly ever user in real-time BCI systems. 2. SCPs signals can not be controlled and their presence often modifies the performance of a BMI. Indeed when using a BMI, that has not been designed for acquiring SCPs signals, for a while could cause a bias that degrades the performance.

Table 1.13: Advantages and drawbacks of using SCPs as signals.

1.4.3 Brain signals artifacts

So far, one has presented different bio-electrical signals produced by the brain activity in several circumstances. However, in any case, the brain electrical activity has a weak amplitude with an order of magnitude in [μV]. Therefore, recording such weak signals is a difficult task because these signals are always corrupted by artifacts and noise. These artifacts generally have larger amplitude than brain electrical signals and can be of several natures. They represent the biggest

challenge when designing a non-invasive BCI. Sometimes, these artifacts are easily eliminated given their frequency content but in other cases, they occupy the same frequency band as signals of interest. This section aims at describing principal artifacts and gives some hints to eliminate them.

1.4.3.1 Environmental noise

This represents the external noise caused by AC power lines, large lighting arrays or electronic devices in the neighborhood. In Europe, AC lines have an oscillation frequency of 50 [Hz]. Introducing a Notch filter that removes this 50 [Hz] component is one possible solution when recording brain electrical activity. More generally speaking, these noise sources operate at high frequency (WiFi and other modem protocols). Therefore, lowpass filtering operations help to eliminate these artifacts. Other solutions exist but require much more equipment (Faraday cage for recording brain activity). Fig.1.18 depicts a comparison between a clean EEG brain signal and an EEG brain signal corrupted by 50 [Hz] noise.

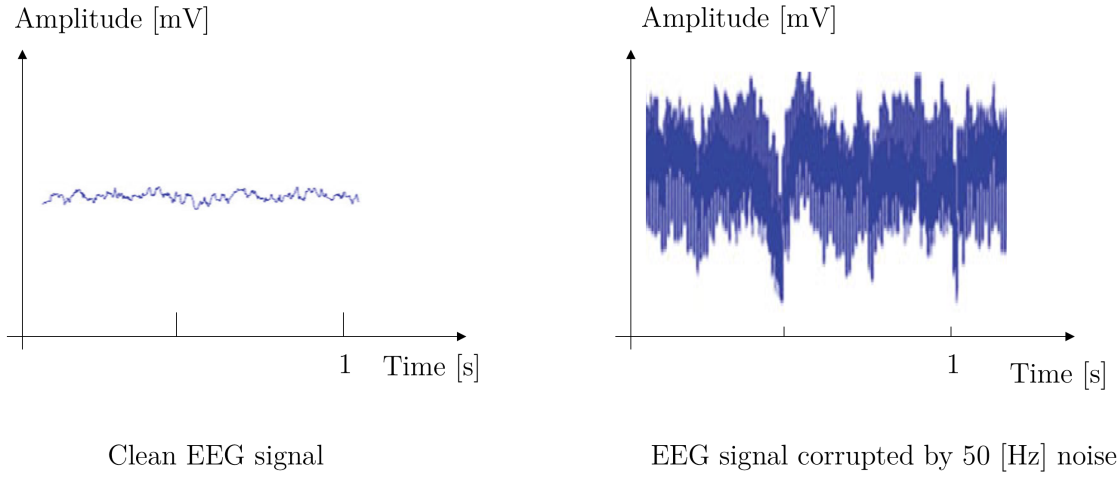
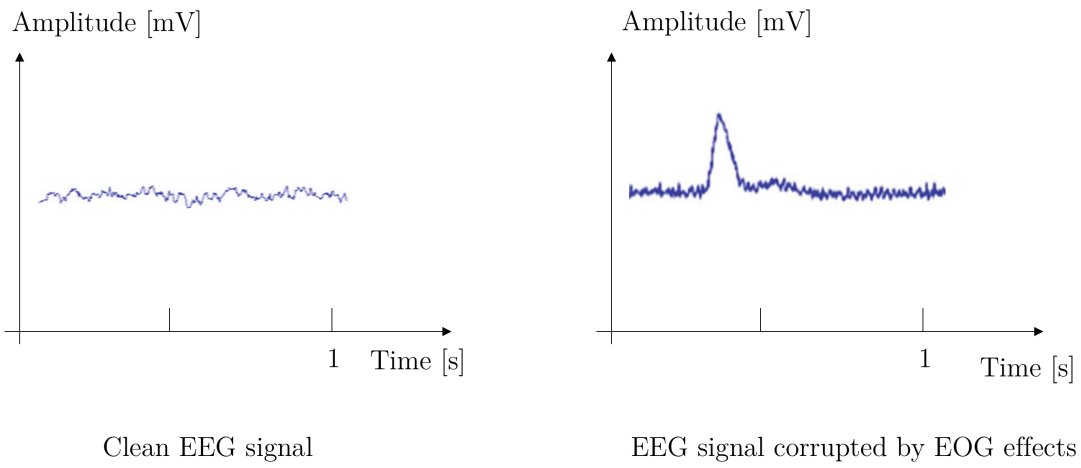


Figure 1.18: Comparison between clean a EEG brain signal and an EEG brain signal corrupted by 50 [Hz] signal artifacts due to AC power lines in Europe.

1.4.3.2 Electrooculargraphic activity

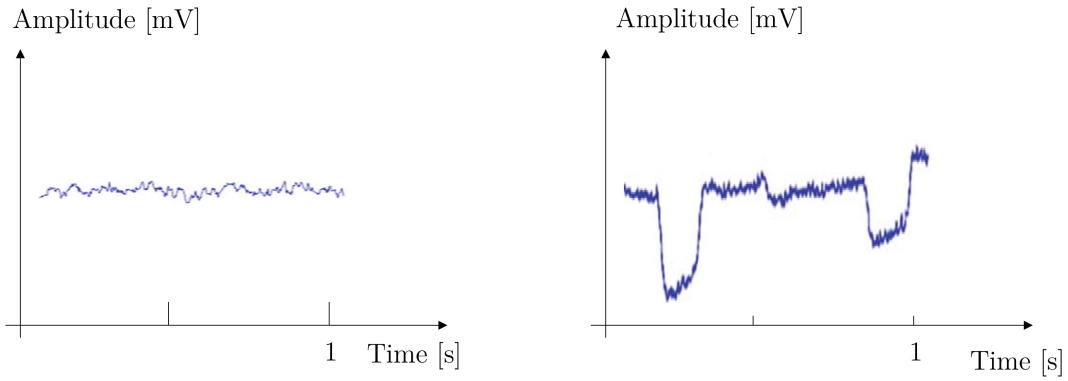
This artifact, also known as EOG effects, is related to the eye movements when one stares at something or even during an eye-blink. Hopefully, these electrical signals have a rather low frequency content, between 0 and 3 [Hz]. Therefore a highpass filter can eliminate these effects. Fig.1.19 and 1.20 depict a comparison between clean a EEG brain signal and an EEG brain signal corrupted by EOG artifacts and eye-blinks.



Clean EEG signal

EEG signal corrupted by EOG effects

Figure 1.19: Comparison between a clean EEG brain signal and an EEG brain signal corrupted by EOG artifacts.



Clean EEG signal

EEG signal corrupted by 2 eye-blinks

Figure 1.20: Comparison between a clean EEG brain signal and an EEG brain signal corrupted by eye-blink artifacts.

1.4.3.3 Electromyographic activity

This artifact, also known as EMG effect, is related to muscles contractions and relaxation that create an electrical signal that can be measured on the scalp. This artifact is the hardest to eliminate because of two reasons :

1. EMG signals occupy the same bandwidth as brain electrical activity.
2. EMG signals recorded form the scalp have a larger amplitude than brain electrical activity, (generally an order 2 to 6 depending on the muscles types [23]).

Removing this artifact requires advanced techniques such as spatial filters. Indeed, these filters aim at identifying where the different sources of electrical signals are localized and allows to remove those who are not localized on the scalp. Notice that a simple frequency filter is not sufficient in this case because of the EMG bandwidth which goes from 4 [Hz] to 60-80 [Hz] [24]. This bandwidth overlaps with θ, μ, β, γ frequency bands. Fig.1.21 depicts a comparison between a clean EEG brain signal and an EEG brain signal corrupted by EMG artifacts.

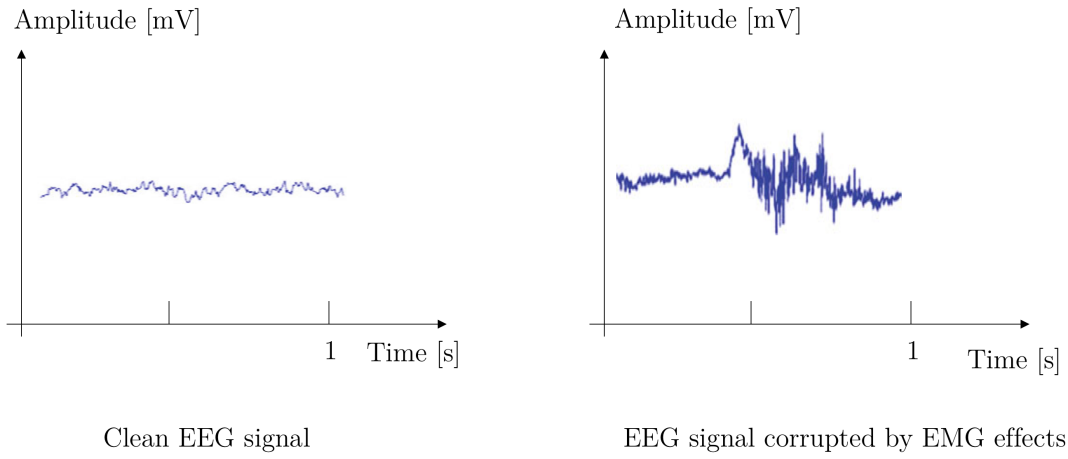


Figure 1.21: Comparison between a clean EEG brain signal and an EEG brain signal corrupted by EMG artifacts.

1.4.3.4 Respiration

Respiration represents also an artifact that corrupts brain electrical signals. Indeed, thorax movements generate low frequency electrical signals. Hopefully, these signals have a low frequency content between 0.5 [Hz] and 2 [Hz]. Therefore a simple lowpass filter can remove this artifact. Fig.1.22 depicts a comparison between a clean EEG brain signal and an EEG brain signal corrupted by respiration artifacts.

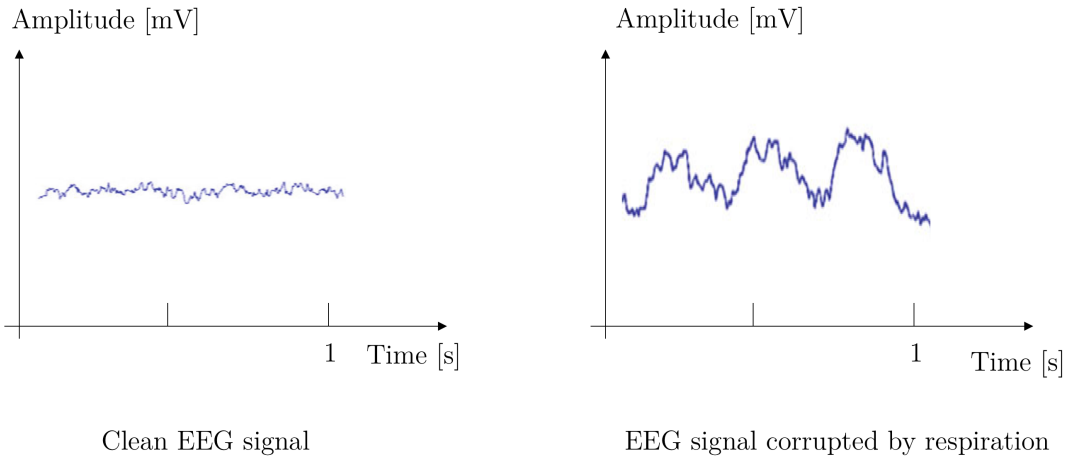


Figure 1.22: Comparison between a clean EEG brain signal and an EEG brain signal corrupted by respiration artifacts.

1.4.3.5 Summary

To summarize, brain electrical signals are corrupted by two categories of artifacts : external perturbations (environmental noise related AC power lines and electronic devices) and internal perturbations (EOG, eye-blinks, EMG and respiration). Regarding environmental noise, one may eliminate it using specific recording conditions. Internal artifacts are, however, always presents to some extent, cannot be controlled or voluntary removed and are typically much more prominent

on the scalp than the macroscopic cerebral potentials. Both perturbations lead to a bad signal to noise ratio (SNR) when acquiring signals from the scalp. Table 1.14 summarizes these artifacts.

Artifacts	**Type**	**Frequency range**	**Amplitude**	**Comments**
Power lines noise.	External.	50 [Hz] up to several [Ghz].	10 to 100 times larger than brain activity signals.	Elimination by highpass filters or isolation of the recording system.
Eye-blinks and EOG.	Internal.	0-3 [Hz].	up to 5 times larger than brain activity signals.	Elimination by lowpass filters.
EMG.	Internal.	4-80 [Hz].	up to 6 times larger than brain activity signals.	Elimination by spatial filters.
Respiration.	Internal.	0.5-2 [Hz].	up to 5 times larger than brain activity signals.	Elimination by lowpass filters.

Table 1.14: Brain electrical signals artifacts.

1.5 Brain electrical signals measured invasively

Beside noninvasive signals acquired form outside of the brain, signals acquired within the cerebral cortex (i.e. beneath the skull) can be used for designing BCI. Roughly speaking, these signals generally present a much better SNR compared to signals acquired on the scalp but require a surgical process in order to be functional. This section aims at introducing principal electrical signals that can be acquired from inside the brain : Local field potentials and Single neuron activity. Notice that this is only an introduction and more details can be found in [25] [26] [27] [28].

1.5.1 Local field potentials

Local field potentials (LFPs) reflect synaptic and neuronal activity from the areas close to the acquisition system. This paragraph first describes LFP waveforms. Then, the occurrence period and eventually the spatial localization are presented.

LPFs present a particular and complex wave shape prior to a movement onset called the movement-evoked potential (MEP). This MEP has two positive (P1 and P2) and two negative (N1 and N2) peaks. When a new movement starts, the positive and negatives are modulated in amplitude. Notice that no one has proved yet that these LFPs exist in disabled people. One may also analyze LPF in frequency domain. Prior to the movement onset, a change in synapse synchronization leads to a successive increase and decrease in power for several frequency bins between 25 and 90 [Hz]. Fig.1.23 depicts the LFP waveforms.

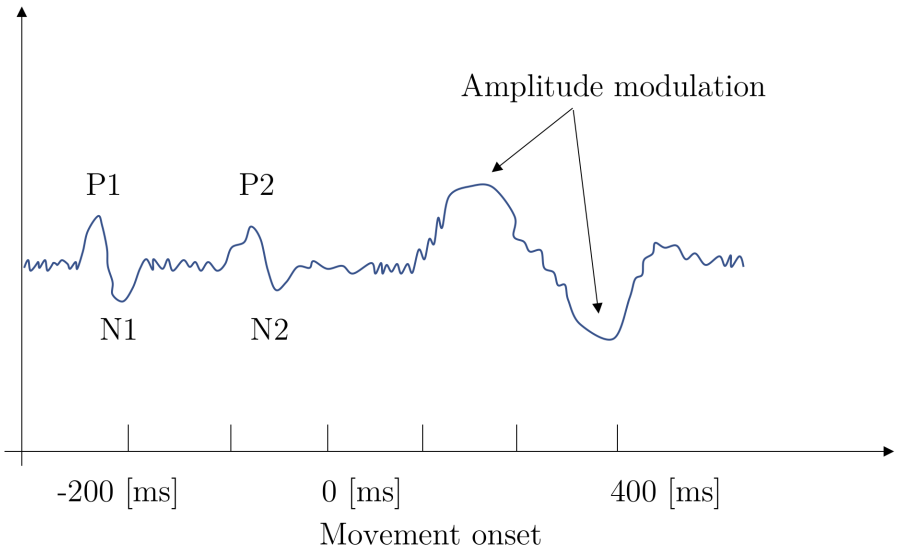


Figure 1.23: Local field potentials signal patterns.

The occurrence period of LFP negative and positive peaks is related to synaptic and neuronal synchronization and synchronization when a motor task occurs. four components occurs 200 [ms] prior to the movement onset (P1, P2, N1 and N2) while two additional components, modulated in amplitude, occur after the movement onset and lasts for 400 [ms]. This is shown in Fig.1.23.

LFP peaks occur in the whole motor cortex. However, it is more pronounced in the primary and supplementary motor areas of cortex. Fig.1.24 depicts the two areas where LFP peaks are more distinguishable.

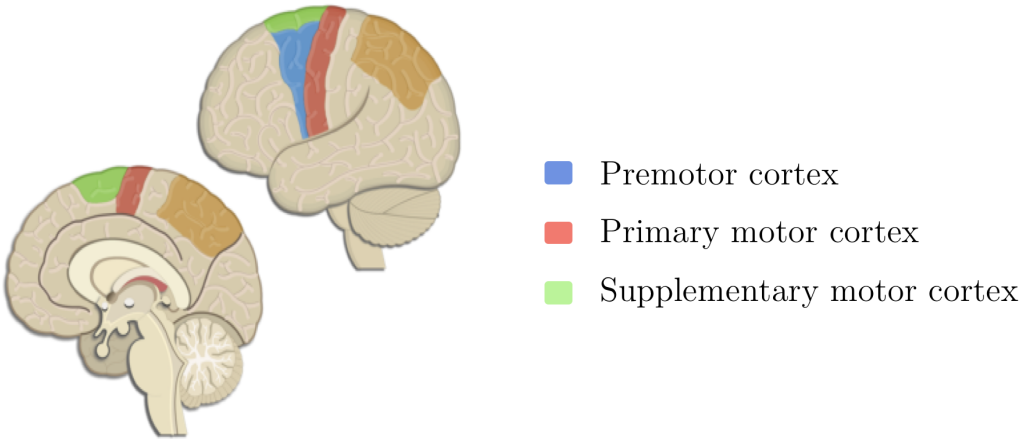


Figure 1.24: Local field potentials signal localization within the motor cortex.

To summarize, LFP peaks in sensory-motor areas in the cortex are interesting when designing a BCI. Such signals exhibit complex wave patterns that are easily recognizable and localized in time across the movement onset. Using LFP peaks as signals to be recorded for a BCI design has advantages and drawbacks, as listed in Table 1.15.

Advantages (+)	**Drawbacks (-)**
1. Excellent SNR. Signals acquired from within the brain. 2. Good spatial resolution. It concerns only a few millions neurons which is much lower than ERPs of VEPs. 3. Wave shape easy to identify.	1 Require an invasive method for acquisition. 2. One has not proved yet presence of such signals for quadriplegic patients.

Table 1.15: Advantages and drawbacks of using LFP as signals.

1.5.2 Single neuron activity

Single neuron activity is about the voltage drop and current that one can measure trough a unique neurons. When subject to an event, neuronal activity increases or decreases (depending on the activity) by modifying the **firing rate**. This is defined as the rate of charge and discharge of a neuron. Several researches have proved that firing rates of individual cortical neurons could be operantly conditioned. This means that the user can learn to control the activity of a single neuron on purpose, after a conditioning protocol. This section briefly describes the behavior of single neurons related to motor tasks. First, single neuron firing signal wave patterns for motor tasks is presented, then the occurrence period of firing signals is explained and eventually the spatial localization of firing activity.

Firing signals of motor cortex neurons are linked to muscles activity. Neuronal firing rates may depend on movement velocity, acceleration, torque, and/or direction. Generally speaking, the neuron first undergoes an increase in voltage (charge) and then a release of this charge (discharge). Let's analyze this situation : the right-hand of a patient is raised. This will lead to an increases of potential (before firing) for single neurons in the contralateral side of the brain (left hemisphere for right-hand movement) while a desynchronization (firing starts leading to a flow of current and thus a decrease in voltage) occurs in the ipsilateral side of the brain. (right hemisphere for right-hand movement) This is depicted in Fig.1.25.

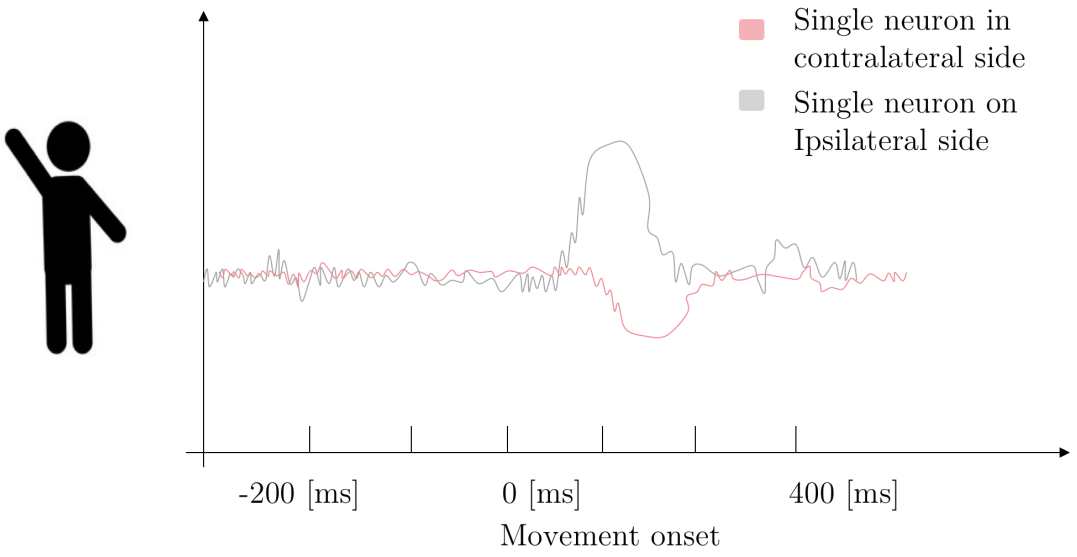


Figure 1.25: Single neuron firing signals signal when a motor tasks occurs.

Firing signals generally occurs between 50 [ms] and 300 [ms] after the movement onset. Notice that that the ipsilateral signal is generally phased out by 20 to 50 [ms] with respect to the contralateral signal. Timing properties are depicted in Fig.1.25.

The motor cortex is the main localization area for single neurons which undergoes firing activity in case of motor task. Notice that for given movement, there is always at least one pair of neurons that activates. One neuron is localized in the contralateral (opposite) side of the brain with respect to the movement and experiments a positive variation of potential. On the ipsilateral side of the brain with respect to the given movement, a negative variation occurs and has generally a lower amplitude.

To summarize, single neuron firing signals in sensory-motor areas in the cortex are of interest for a BCI design purpose. Such signals exhibit an antagonist behavior regarding both hemispheres. Using single neuron firing signals as signals to be recorded for a BCI design has advantages and drawbacks. This is depicted in Table 1.16.

Advantages (+)	**Drawbacks (-)**
1. Excellent SNR. Signals acquired from within the brain 2. Good spatial resolution. It concerns only a few neurons. 3. Antagonist wave patterns on both hemispheres whihc lead to a pattern easy to identify.	1 Require an invasive method for acquisition.

Table 1.16: Advantages and drawbacks of using single neuron firing signals as signals.

1.6 Recording methods

One will discuss different signals acquisition methods for measuring and recording brain electrical signals, invasively or not. Each method will be introduced and the different brain electrical signals that can be acquired with this particular method will be enumerated. As a,reminder, this work focuses on brain electrical signals for designing a BCI. Therefore only electrical signals recording methods are presented. This sections presents three different acquisition methods : the electroencephalogram (EEG), electrocorticography (ECoG) and intracortical neuron recording.

1.6.1 Electroencephalogram

Using an EEG recording method from the outside the brain is the most practical and used method for measuring brain activity. One can record the voltage from the scalp (i.e the surface of the brain) with respect to the potential of another part of the body, which is generally the ear. The voltage is recorded thanks to a set of electrodes. Moreover EEG is inexpensive and convenient. Due to the growing interest for BCI, electrodes have been developed commercially into portable devices. This paragraph first introduces the working principle of EEG, then the electrodes arrangement and eventually the reference localization.

An EEG system is made of at least three electrodes : a ground electrode, also known as reference electrode and two electrodes that monitor the brain activity. Fig.1.26 (left figure) depicts the structure of the acquisition device. The voltage difference between the electrodes is amplified by a differential amplifier with a gain A. This stage is followed by a lowpass filter that aims at removing high frequency noise and avoid spectral folding when digitization will be performed. Notice that the patient must be isolated from the ground voltage of the power supply. The voltage $V_2(t) - V_1(t)$ difference measured on the scalp is amplified. Notice that this voltage difference is generated by both brain current source (as described in [29]) and biological and environmental artifacts. In Fig.1.26 (right figure), the potentials measured on the brain at location 1 and 2 of the scalp are given :

$$V_{1,tot}(t) = V_1(t) + V_{CM}(t) \qquad V_{2,tot}(t) = V_2(t) + V_{CM}(t) \tag{1.1}$$

$V_{CM}(t)$ is the common-mode potential, mainly caused by external artifacts (power lines and other electric devices) and internal artifacts. The purpose of the amplifier is to eliminate the common-mode potential and to amplify the voltage difference of the two electrodes. Ideally, the output of the amplifier should be $E(t) = A(V_1(t) - V_2(t))$ with A is the total gain of the different amplifying stages. Notice that a reference electrode is used as ground voltage but also helps to prevent the amplifier from drifting away and helps to better reject the common-mode potential. The contact between the electrode and the scalp has to be ensured with a contact impedance less than $10[k\Omega]$. This sometimes requires to use a conductive gel or paste to decrease the resistivity [30] (Ferree et al, 2001). Notice that we need to distinguish *wet electrodes* and *dry electrodes*. The wet electrodes need a low contact impedance between them and the scalp. Therefore, a conductive paste is required. The second category is directly in contact with the scalp without requiring a conductive gel. The advantages and drawbacks of each type of electrode are listed in Table 1.17.

The disposal of the electrodes depends on what type of brain signals you want to record. For recording P300a signals, one will generally dispose electrodes all over the frontal and temporal lobes while for recording event-related potentials for motor tasks, one will place the electrodes across the sensory-motor cortex. There exist international disposal conventions known as 10-20, 10-10 or 10-5 placements systems. Fig.1.27 depicts the different electrodes placement systems. The 10–20 montage is indicated by the 21 electrodes shown as black circles. The 10–10 structure

consists of the 21 electrodes of the 10–20 montage (black circles) with 53 other electrodes shown in gray. This corresponds to a total of 74 electrodes. Eventually the 10-5 system is made of the electrodes of the 10-10 system and 68 additional electrodes (open circles). This corresponds to the most complete and homogeneous brain coverage.

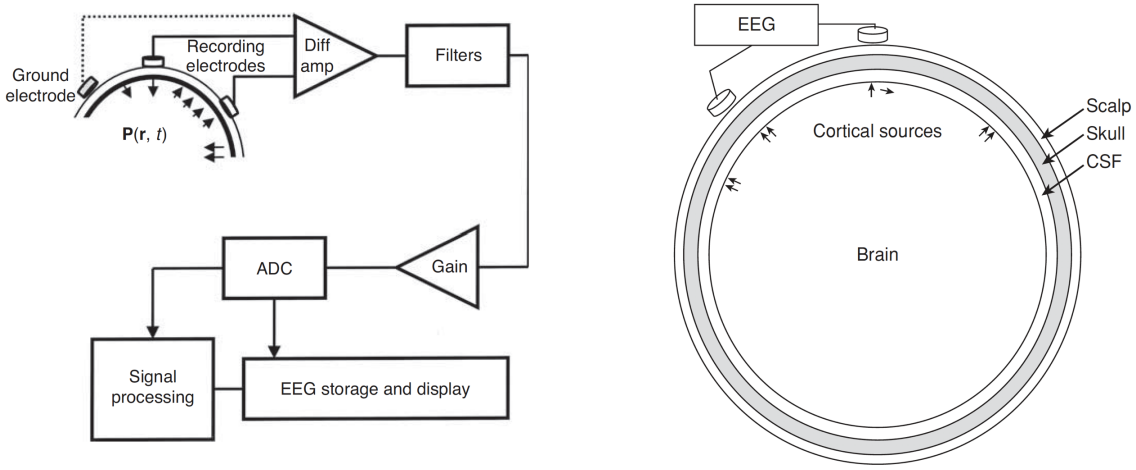


Figure 1.26: EEG acquisition system with differential amplifier and reference eletrcode.

EEG signals are bipolar signals. It means that there is no way for recording monopolar EEG signals. We always need a pair of electrodes. One of the electrode corresponds to the recording electrode while the other corresponds to the reference electrode. The choice of the reference electrode and its position is crucial because in most EEG practice, the potentials measured for all the other electrodes are recorded with respect to one reference electrode. Different common reference electrode locations are used : the neck, the ear or even on the head itself. One will discuss briefly the three most used reference electrode locations : the closely spaced electrodes, the linked ear and the common average reference. Fig.1.28 depicts the three cases.

The closely spaced electrodes is a system where the the reference electrode is surrounded by the measurements electrodes in a small area. This configuration allows to measure accurately very localized source but can not be used for recording a large area of the brain. When the two electrodes get closer together, the measurement of the potential and the gradient of the potential along the scalp surface between the electrodes (i.e. the voltage) gets better.

The linked ear is the most popular recording configuration. In this case, the reference electrode is connected to one of the ear. The measured voltage is in that case expressed with respect to the ear electrical potential. The major advantage of this technique is the possibility to measure the potential of the entire head with respect to the ear. It means that it performs well for any measurement electrode location on the head.

The common average reference system generalizes the linked-ear system. In this case, the reference electrode is located anywhere on the head. The reference electrode position becomes a parameter that one may play on for improving the measurements depending on the application.

To summarize, EEG consists in acquiring a potential on several parts of the scalp using electrodes, which can be either dry or wet (i.e. using a conductive past). In order to do such an acquisition, a reference electrode must be chosen. EEG acquisition systems represent the best compromise in terms of accuracy, reliability, usage simplicity and cost.

	Advantages (+)	**Drawbacks (-)**	**Materials**
Wet electrodes	1.Low sensitivity to environmental noise 2. Lifespan is longer than dry electrodes.	1. Need of a conductive paste/gel which is very inconvenient for practical use. 2. The paste must be changed after each use or during a long use	Yin (Sn), silver/ silver-chloride (Ag/AgCl), gold (Au), or platinum (Pt).
Dry electrodes	1. Ease of use and convenience for the user 2. Can be used for hours in a row without contact problems	1. More sensitive to artifacts than wet electrodes .	Inert materials (Pt, Au, Fe, AgCl) or even insulators.

Table 1.17: Advantages and drawbacks of wet and dry electrodes.

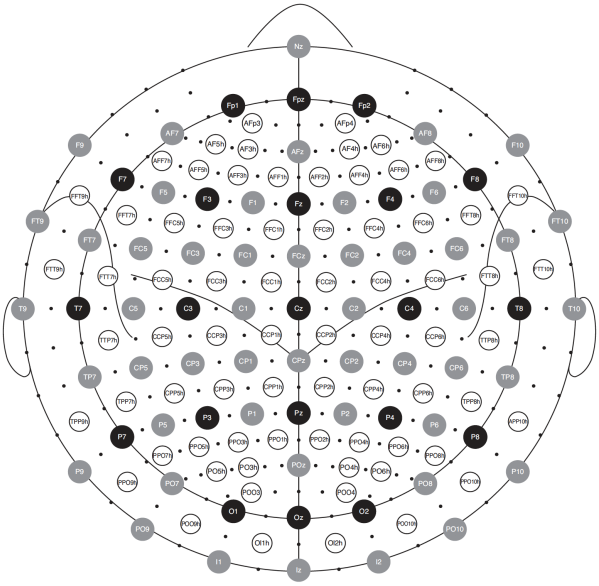


Figure 1.27: Electrodes placement systems (black dots for the 10-20 system; black and gray dots for the 10-10 system and all electrodes for the 10-5 system).

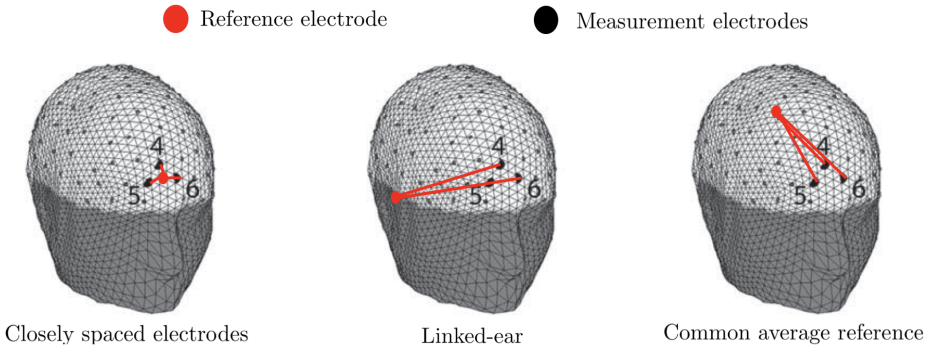


Figure 1.28: Three possibilities for the reference electrode location.

1.6.2 Electrocorticography

Electrocorticography (ECoG) is an other method for recording the brain electrical signals. The main difference with EEG is the invasiveness of this method given that it records directly from the surface of the brain and not from the scalp. The main advantage of recording directly from the brain is the higher spatial resolution, the broader bandwidth, the higher amplitude of measured signals and the lower sensitivity to artifacts such as electromyographic (EMG) signals and electrooculargraphic (EOG) signal. The price to pay is a surgical intervention for positioning the sensors. This paragraph will explain (1) the structure and working principle of ECoG. Figure 1.29 depicts a comparison between EEG and ECog recording localization and electrodes.

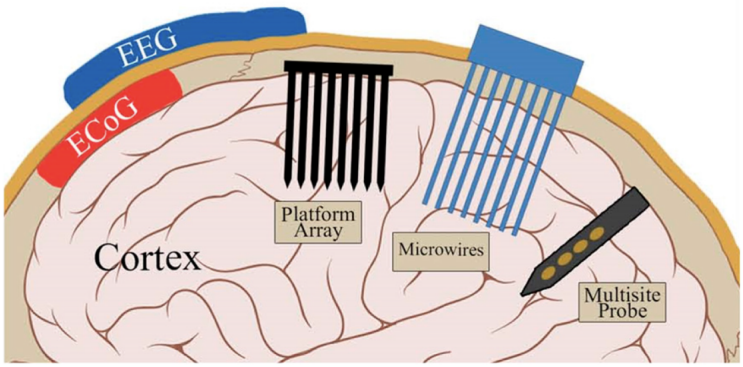


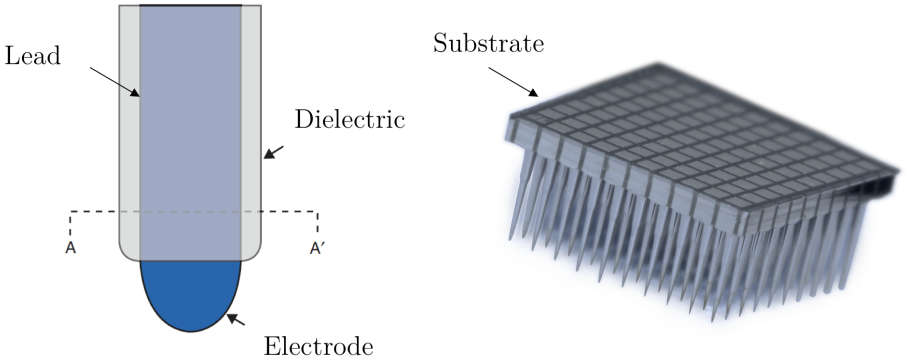
Figure 1.29: Comparison between EEG and ECog recording localization.

ECoG systems have a working principle close to the EEG principle except that electrodes are placed on the surface of the brain. They can record local field potentials (LFPs) and the activity of localized sets of neurons and synapses thanks to their high spatial resolution. One may find more details in [31] [32] [33].

1.6.3 Intracortical Neuron Recording

Beneath the skull, the motor cortex is made of gray matter as described earlier in this chapter. The activity of neurons within this gray matter is responsible for motor tasks. Therefore, instead of recording the potential at the surface of the motor cortex, one may record directly the activity of neurons trough a microelectrode array implemented inside the brain. This paragraph aims at presenting such a technology and is organized as follows : first, a general overview of microelectrode array is presented and then the working principle of such an array is briefly introduced.

Such electrodes are designed to penetrate into the cerebral cortex and be positioned in close proximity to the targeted neuronal population (movement-related areas of the cortex). Hence such arrays have a different design compared to EEG electrodes. The structure is generally made of three components : an array of spikes electrode where bio-currents transduction is done. A set of leads which connect the spikes array to an electronic interface. Notice that each lead is electrically insulated from the brain by a dielectric. Each spike is connected on a substrate that is ensures the mechanical cohesion of the system. Notice that the dielectric may also play the role of substrate. Fig.1.30 depicts the structure of such an array.

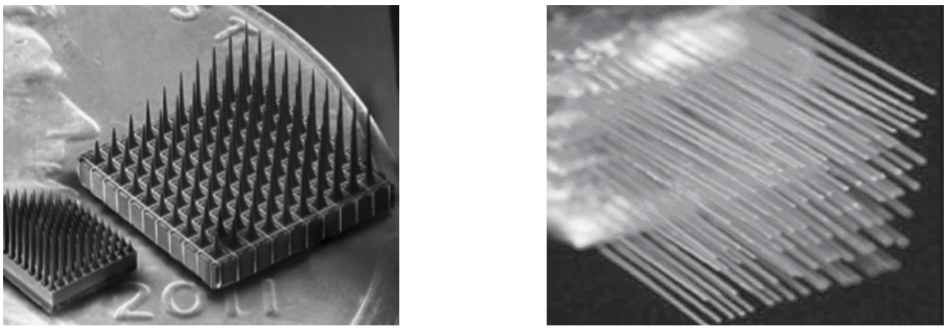


(a) Spike

(b) Microelectrode Array

Figure 1.30: Structure of a micro-electrode array. One spike of the array is depicted in (a) and the entire array connected to the substrate in (b).

Notice that there exist numerous microelectrode array categories depending on the fabrication technologies or the package. For example, microelectrode arrays are produced either using bundles of microwires or MEMS fabrication processes. Fig.1.31 depicts a comparison between these two types of microelectrode arrays.



(a) MEMS-based microelectrode array

(b) Microwires-based microelectrode array

Figure 1.31: Comparison between two types of microelectrode arrays. The first is designed based on MEMS fabrication techniques while the second one is designed with micro-wires.

The biopotential at the spikes surface is related to currents flowing trough neurons and synaptic potential among neurons bundles. Notice that brain tissues between the neurons is considered as a homogeneous volume conductor but not necessarily isotropic. Therefore the biopential in the vicinity of the spike location is transduced to an electronic current thanks to two phenomena [34] [35] :

1. Capacitive displacement current related to the intrinsic capacitance of the interface between brain tissues and the spike.

2. Chemical current resulting from oxidation-reduction reaction at the electrode surface.

These two sources of current consist in the measure of both single-neurons activity and local field potentials while using intracortical measurement devices. A lumped-model of the intracortical neuron recording has been established in [**?**]. This model represents the recording process of K neurons using a microelectrode array of L electrodes with an additional electrode used as reference electrode and localized in the surroundings. Each channel of the microelectrode array consists in 3 blocks in series :

1. Bipoential sensing which represents the total potential contained in the recording volume at the electrode site.

2. The neural tissue impedance which represents the impedance of the tissues in the vicinity of the electrode site.

3. The tissue-electrode interface impedance which represents the capacitive impedance between the electrode and the surrounding tissues.

This model can be used for simulating intracortical measurements. More details can be found in [**?**]. Fig.1.32 depicts this lumped-model.

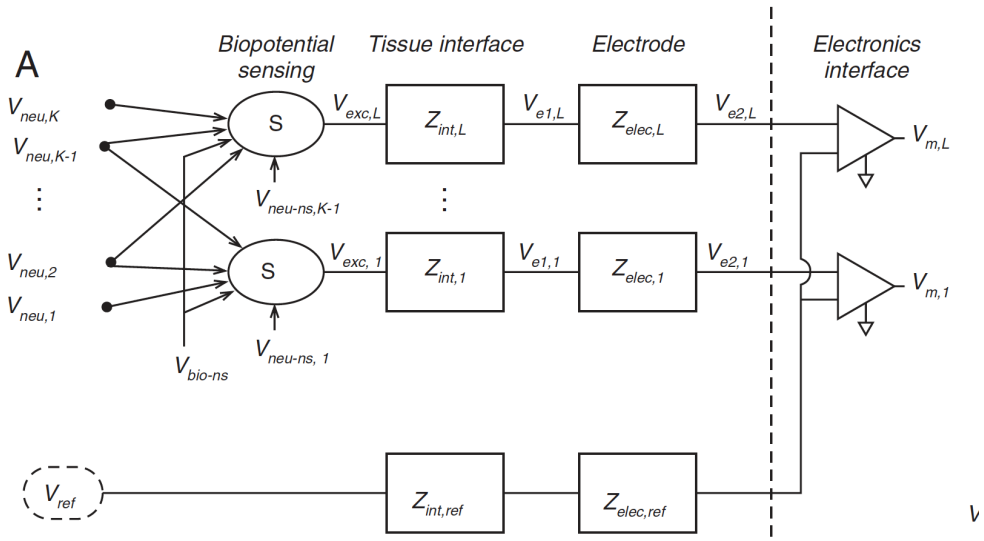


Figure 1.32: Lumped model of a microelectrode array that records activity of K neurons with $L+1$ electrodes.

To summarize, intracortical neurons recording is a technique that consists in acquiring brain electrical signals directly in the brain tissues by inserting microelectrodes spikes that penetrate within the tissues. Such an acquisition system can be used for recording local field potentials or single neurons activity.

1.7 General comparison of BCI systems.

After having presented a general overview of BCI systems using brain electrical signals, one will rank these systems based on their properties and select the most appropriate combination in order to design a BCI in this work.

1.7.1 Active/Reactive BCI systems.

The first criterion used for comparing BCI systems is related to the nature of signals used as inputs. *Active* BCI systems require the user to modulate its brain activity by active thinking to produce signals used by the BCI to predict the user's intent. The user generally learns to control its brain signals to improve the capability of the BCI system. On the other hand, *Reactive* BCI systems measure the reaction of the user to an external stimulus. Table 1.18 depicts a comparison between active and reactive systems. **Regarding this table, we will choose an active BCI in this work**. This is justified by the fact that the authors desire to design a portable BCI that may operate without bothering the user with the need of external stimuli.

1.7.2 Synchronous/Asynchronous BCI systems.

The second criterion used for comparing BCI systems is related to the detection and prediction timing. On the one hand, an *asynchronous* BCI records and analyzes continuously brain electrical signals, regardless of the user's activity. Once it detects a user's intent, it produces a prediction. On the other hand, a *synchronous* BCI system operates in two stages. It first records brain signals during a specific time windows. Then it processes recorded signals and predict the user's state. Table 1.19 depicts a comparison between synchronous and asynchronous systems. **Regarding this table, we will choose a synchronous BCI in this work**. This can be justified by the simplicity of design while maintaining good performance if the acquisition window and processing period are both shorts.

Type	**Advantages (+)**	**Drawbacks (-)**
Active BCI	**1.** Few required recording material. No need for system that generates external stimuli. **2.** User operates the system whenever. No need to wait for an external stimuli. **3.** User can learn to modulate its own signals.	**1.** Much more challenging system to be designed. **2.** Requires an extensive training process. **3.** Requires complex algorithm to identify accurately the user's intent.
Reactive BCI	**1.** Accurate prediction of the user's intent with simple algorithms. **2.** Short training process is required. **3.** User does not need to concentrate to control the BIC system.	**1.** Require an stimuli generator. **2.** The user's sensory capabilities should not be damage in order for some BCI systems to work (sensitive stimuli). **3.** System is not necessarily portable because of the stimuli generator. **4.** The user may be bothered to be continuously subject to external stimuli.

Table 1.18: Comparison of active/reactive BCI systems.

Type	Advantages (+)	Drawbacks (-)
Synchronous BCI	**1.** Processing chain and signal acquisition are executed sequentially leading to a simple design. **2.** A prediction of the user's intent is given periodically. **3.** Can be implemented on chips with low computational power and/or low memory.	**1.** The system predicts user's intent based on signals that have been recorded a few seconds before. The system has thus a slow time response. **2.** Require the user to plan/execute a motor task during an entire acquisition window.
Asynchronous BCI	**1.** Fast prediction of user's intent. **2.** Continuous acquisition and prediction leading to a natural behavior of the system.	**1.** Require a parallel programming with several treads and thus large computational power and/or memory. **2.** More difficult to train because a training protocol is not continuous.

Table 1.19: Comparison of synchronous/asynchronous BCI systems.

1.7.3 Invasive/Noninvasive BCI systems.

The third criterion for comparing BCI systems is related to the invasivness. As reminder, an *nvasive* BCI system records brain electrical signals directly from the cerebral motor cortex while a *noninvasive* BCI system acquires signals from the scalp/head. Table 1.20 depicts a comparison between invasive and noninvasive systems. **Regarding this table, we will choose a noninvasive BCI in this work**. One cannot afford to use invasive technology while designing a BCI for a commercial purpose.

Type	Advantages (+)	Drawbacks (-)
Invasive BCI	**1.** Excellent signal quality with large signal-to-noise ratio, good temporal and spatial resolution. **2.** Such BCIs generally exhibit better performance.	**1.** Require a surgical intervention for placing sensors and recording material. **2.** Difficulties for long-term usage. **3.** Hardware can not be easily updated or modified. **4.** Such technologies are not usable for commercial purpose.
Noninvasive BCI	**1.** Acquisition system is positioned on the head by anyone in a few minutes. **2.** Acquisition system can be modified and/or updated in a straightforward manner.	**1.** Acquired signals are subject to numerous artifacts when acquired from outside the brain leading to a poor SNR.

Table 1.20: Comparison of invasive/noninvasive BCI systems.

1.7.4 Signals types for BCI systems.

The fourth criterion for comparing BCI systems is related to the type of electrical signals used for the BCI design. One has presented seven different electrical signals : Cortical oscillations, event-related potentials, visual-evoked potentials, P300 signals, slow cortical potentials, local

Signal Type	**Temporal localization**	**Spatial localization**	**Invasive or not ?**	**External stimuli**	**SNR**
Cortical oscillations	++	+	No	No	- -
Event-related potentials	++	++	No	No	-
Visual-evoked potentials	++	+	No	Yes	+
P300 signals	++	+	No	Yes	+
Slow cortical potentials	- -	+	No	No	+
Local field potentials	++	+	Yes (low)	No	++
Single Neuron Recording	++	++	Yes (high)	No	++

Table 1.21: Comparison of 7 different brain signals for a BCI design.

field potentials and single neuron activity. Table 1.21 depicts a general comparison of each signal in terms of several criteria.

1. Temporal localization of a signal refers to its occurrence period with respect to the event that triggers it. if the signal appears immediately prior or immediately after the event and lasts less than 1-2 [s], its temporal localization is considered as good and it is denoted with a '++' score. On the other hand, if the signal lasts for a while after the event that triggers it, it is denoted wit a '- -' score.

2. Spatial localization of a signal refers to its location within the brain. A signal that occurs in a small and well-defined area is denoted by with a '++' score. Otherwise if the signal can occurs in the whole cerebral cortex, it is denoted by with a '- - ' score.

3. Invasiveness. Does the signal require an invasive acquisition method to be measured. One will precise the degree of invasiveness when needed.

4. External stimuli. Does the signal require an external stimuli to be measured.

5. Signal-to-Noise Ratio of a signal. It measures the power of desired signal with respect to the power of artifacts signals. A SNR of 5 [dB] is considered as excellent and is denoted by '++' while a SNR of -10 [dB] is denoted as '- -'.

Regarding Table 1.21, **we will choose event-related potentials as information signals for predicting user's intent in this BCI design**. One may justify this choice for two reasons : First, ERPs allows to design noninvasive and active BCI as specified earlier in this section. Then, ERPs have better spatial localization properties than cortical oscillations and better temporal localization than slow cortical potentials.

Method	**Invasiveness**	**Temporal resolution**	**Spatial resolution**
EEG	No	0.05 [s]	10 [mm]
ECoG	Yes (low)	0.005 [s]	5 [mm]
Intracortical array	Yes (high)	0.005 [s]	1 [mm]

Table 1.22: Comparison of 3 different acquisition methods for a BCI design.

1.7.5 Acquisition methods for BCI systems.

The last criterion for comparing BCI systems is related to the acquisition method of electrical signals used for the BCI design. One has presented three different acquisition methods : (1) EEG, (2) ECoG and (3) Intracortical array. Table 1.22 depicts a general comparison of each acquisition method in terms of several criteria. Notice that methods presented in this work only record electrophysiological brain activity.

1. Invasiveness. This corresponds to the first characterization that indicates whether the method acquires electrical signals from inside of the brain or not. One will precise the degree of invasiveness when needed.

2. Temporal resolution. This characterizes the accuracy in terms of timing that one may expect form an acquisition method. This is related to the sampling rate than one method can reach without measuring noise only.

3. Spatial resolution. This characterizes the spatial accuracy that one may expect from an acquisition method. Another way for interpreting this quantity is as follows : An acquisition method has a spatial resolution of d [m] method able to distinguish at most two electrical brain sources separated by d [m].

Regarding table 1.22, one will choose EEG as acquisition method because this is the only noninvasive recording method for brain electrical signal.

1.8 General specifications and conclusion

To summarize, this chapter briefly introduced the concept of BCI. It explained the interest in BCI for quadriplegic persons. Furthermore, brain electrical signals that are used in most applications in the literature are introduced. These signals are classified into two main categories: signals acquired from outside of the brain and signals acquired from within the cerebral cortex. Each type of signal has pros and cons. One may choose a trade-off between signals with precise temporal localization, precise spatial localization and high SNR. Besides these signals, acquisition methods were presented. Electroencephalogram is the best alternative for noninvasive measurements in terms of ease of use, accuracy and cost. Eventually, a large comparison between different BCI configurations was performed. This comparison helped the authors defining the general specifications of the BCI in this work. These specifications are enumerated in Fig.1.33.

1. The BCI system will be noninvasive. It will acquire electrical signals from the scalp.
2. The BCI system will use a synchronous working principle.
3. The BCI system will be active. It will record the user's intention and will not record the user's reaction to external stimuli.
4. The BCI system record event-related potentials related to SMR rhythms (μ band essentially).
5. The BCI system will use EEG as acquisition method.
5. The BCI system is able to recognize three motor tasks : (1) a right-hand and arm movement, called in the rest of this work **right-hand movement**, (2) a left-hand and arm movement, called in the rest of this work **left-hand movement** and (3) no movement (the user is at rest) called in the rest of this work **rest state**.

Figure 1.33: General specifications for the BCI.

Chapter 2

Processing chain

In this chapter, the processing chain showed in Fig.1.2 is detailed. As reminder a BCI consists in a sequence of operations that processes raw brain signals, taken as inputs, and produces a prediction of the user's intent as output. More specifically, the processing chain can be decomposed in 3 mains stages : the preprocessing (itself decomposed in two main sub-stages : frequency filtering and spatial filtering), the feature extraction and feature classification. These stages are explained within this chapter by briefly presenting the various algorithms.

The chapter is organized as follows : First the prepossessing operations are presented. Next relevant information must be extracted from preprocessed signals. This is realized by the feature extraction stage, which is explained here. Eventually, the feature classification stage, which consists in assigning a class to the feature vector, is described.

This chapter aims at being as exhaustive as possible by presenting a review of the numerous techniques that exist and comparing them. Notie, neverthelessess, that many more details can be found in [1] [3] [4].

2.1 Preprocessing

The first step of the signal processing chain is called the preprocessing step. This step enhances the signal acquired from the electrodes by eliminating known interferences or irrelevant information. This step, also called signal conditioning in [1], can include different procedures such as:

1. Frequency-range prefiltering.
2. Spatial filtering.
3. Artifact removal.
4. Data decimation and normalization.

We will focus on the two first procedures, frequency filtering and spatial filtering, that are the ones used and implemented in almost every BCI.

2.1.1 Frequency filtering

Frequency filtering is the first important procedure in the processing chain as it is used to eliminate frequencies that lie outside the frequency range of the brain activity most relevant for the application. Digital filters are keys to signal processing. They are used to modify the signal and its frequency content in order to isolate some frequencies in a specific range. Digital filters

will thus attenuate some frequencies and preserve others. The generic equation giving the filter's output $y[n]$ at any given time is the weighted sum of the current and the past input samples:

$$y[n] = \sum_{l=0}^{M} b_1 x[n-l] - \sum_{l=1}^{N} a_1 y[n-l] \tag{2.1}$$

where $x[n-l]$ is the l^{th} past input sample, $y[n-l]$ is the l^{th} past output sample, b_1 and a_1 are scalar weights for each input and output sample, respectively, and M and N are the number of input and output sample weights employed by the filter, respectively. The simplest example of a digital filter is the uniform moving average filter. The filter output at any time is equal to the sum of the N past samples each multiplied by $1/N$, which is equivalent to taking the average of the N past samples. The digital filters can be divided into 4 applications, as shown in Fig.2.1:

1. Low-pass filter
2. High-pass filter
3. Notch filter
4. Bandpass filter

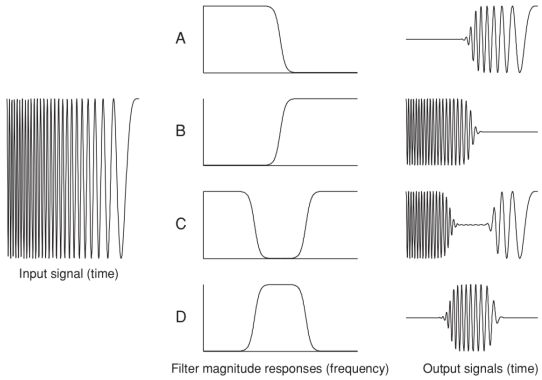


Figure 2.1: Four different types of digital filters and their output signals. Respectively, the low-pass filter, high-pass filter, notch filter and bandpass filter.

In the field of BMI, it is very common to apply a 0.5-40 Hz bandpass filter at the beginning of the chain to isolate the relevant brain activity corresponding to the application. Narrower bandpass filters can also be used to isolate more specific frequency bands showed in Fig.1.6. Another typically used filter is a 50/60 Hz notch filter to eliminate the power line interference that is caused by the power lines and power sources present in the environment when the signals are recorded. The interference manifests itself as a continuous sinusoidal signal at 50 Hz (in Europe and Asia) and 60 Hz (in America). A digital filter can either be a *Finite Impulse Response* filter or an *Infinite Response Filter*. Depending on the type of filter, Eq.2.1 will be different.

2.1.1.1 Finite Impulse Response filters

Finite Impulse Response filters (FIR) are moving average filters without feedback of the past outputs. The term *finite* comes from the fact that a single sample will impact a finite number of output samples. FIR filter uses only current and past input digital samples to obtain a current output sample value. Since there is no feedback in a FIR filter, the output is guaranteed to be stable at any time: the output signal cannot grow without bound if the input signal is limited in its amplitude. Moreover, FIR filters can be designed with a linear phase response, meaning that

the time delay introduced by the filter is the same for all frequencies. One of the disadvantages of FIR filters is that the filter length increases with the desired sharpness in the magnitude response transitions. Very long filters are needed to produce very sharp transitions, which can be a problem in the design of a BCI as the time delay introduced by the filter will increase.

2.1.1.2 Infinite Impulse Response filters

IIR filters use current input sample value, past input and output samples to obtain current output sample value. The filter is said to have an *infinite* impulse response because a single input will affect an infinite number of output samples because of the feedback used. The great advantage of IIR filters is that they can produce sharp transitions in the magnitude response without increasing their length. In comparison with FIR filters, IIR filters can be unstable because of the feedback. A critical point in the design is thus the stability of the filter. Another critical point is that IIR filters often have a nonlinear phase response: different frequencies will experience different time delays.

2.1.1.3 Adaptive filters

The filters defined earlier have constant coefficients, independent from the input values. Another category of filters consists in adaptive filters. Adaptive filters modify their behavior with respect to the input signal in order to attenuate the frequency ranges which are subject to lots of artifacts.

2.1.2 Spatial filtering

The quantity measured between two electrodes is the electrical potential (in volts), and the voltage signal obtained is called a channel. The electrical signal acquired depends strongly on the placement and orientation of the channels electrodes. By selecting the pairs of electrodes that comprise each channel, the channel can become more sensitive to some sources and less sensitive to other sources. By weighting and combining channels, it is possible to reconstruct some desired set of channels. This procedure is known as *spatial filtering*. In general, spatial filters are used to improve source localization, to enhance the sensitivity and to suppress artifacts. All spatial filers can be described as a linear combination of selected channels and represented in a matrix notation

$$Y = WX \tag{2.2}$$

where each row of X consists of P consecutive digital signal sample from one of N channels, each row of W is a set of N channel weights that constitute a particular spatial filter, and each row of Y is a resulting spatially filtered channel (M spatially filtered channels x P samples). Five different spatial filters are described and there are basically two different types of filters. The two first filters, the Common Average Reference and the Laplacian filters are called *data independent filters*, while the Principal Component Analysis (PCA), Independent Component Analysis (ICA) and the Common Spatial Pattern (CSP) filters are called *data dependent filters*. These filters, more complex in terms of derivation, can provide more precise results.

2.1.2.1 CAR

A *Common Average Reference* spatial filter can be realized by computing at each time step the mean of all channels and then substract the mean from each individual channel. This type of filter helps reducing the effect of artifacts present over all channels, for example the 50 Hz power line interference.

2.1.2.2 Laplacian

The *Surface Laplacian Filter* is computed using the second spatial derivative. This computation is equivalent to talking a central channel of interest and substracting the mean of all channels at a fixed radial distance from this central channel. Adjacent channels are highly correlated since they have similar positions relative to many brain sources. By eliminating this correlated activity, a Laplacian filter emphasizes highly localized activity. This localized activity is the one we are looking to isolate.

2.1.2.3 Principal Component Analysis

PCA determines the set of weights W that transform the set of channels X into a set of new channels Y having the same dimensionality, such that the filtered signal in Y associated with the first principal component is the linear combination of the signals in X that produces the maximum amplitude variance. When using PCA, the resulting components are orthogonal to each other and the resulting channels are thus uncorrelated with each other. To be uncorrelated, the expected value of two channels x and y must satisfy

$$E\{XY\} = E\{X\}E\{Y\} \tag{2.3}$$

Each successive principal component accounts for a smaller portion of the amplitude variance of the original channels. As the first principal components capture most of the variance of the channels in X, the dimensionality of the problem can be reduced. Ideally, this process is used to discard little relevant information.

2.1.2.4 Independent Component Analysis

Whereas PCA aims at determining the set of weights W that produce components that are uncorrelated with each other, ICA seeks to produce the matrix W that offers components independent to each other as outputs. To be statistically independent, the joint probability distribution function $F_{XY}(x, y)$ of channels x and y must satisfy

$$F_{XY}(x, y) = F_x(x)F_y(y) \tag{2.4}$$

Thus, independence is a stricter constraint that uncorrelatedness. Because the resulting spatial weights tend to correspond to localized activity, this process can also be understood as a form of source separation or source localization. ICA presents a relatively challenging implementation in BCI applications. The algorithms are relatively complex, and the number of independent sources must be accurately approximated to isolate the wanted source information.

2.1.2.5 Common Spatial Pattern

CSP is a similar approach to PCA except that W in Eq.2.2 is determined such that the corresponding components produce minimum variance for one condition and maximum variance for the other, and vice versa. For CSP, the channels in X are first bandpass-filtered, making the variance equivalent to bandpower. As with PCA and ICA, only the relevant output components are retained. This spatial filter will be explained in detail in Chapter 5 as it is implemented in our realtime application on a FPGA.

2.2 Feature Extraction

In this section, the feature extraction step of the processing chain is explained. The most common algorithms will be briefly described, and the one implemented on the experimental board will be further explored in section 5. The feature extraction step constitutes the transformation of

fundamental signal characteristics into a feature vector. There exists three different types of feature extraction: temporal (time) methods, spectral (frequency) methods and mixed time-frequency methods. Note that when the feature extraction method is applied, the signal is often decomposed in blocks where the method is applied for each block. This allows to construct a feature vector with different features corresponding to the different time blocks. For example, in our simulations, if the signal duration was 10s, we often decomposed the signal into blocks of 1s and then applied the feature extraction method to the 10 different blocks in order to construct the feature vector. This procedure, often used in the field of BCI for efficient analysis, is also known as *windowing and framing*. This sections aims at describing the feature properties and the different methods to extract these features. Some of the most common temporal, spectral and time-frequency methods are explained.

2.2.1 Feature properties

The feature vector is the output of the feature extraction step. It is therefore important to clearly understand the critical properties of these features:

1. Noise and outliers: features can be noisy and contain outliers because of the low signal to noise ratio of EEG signals.
2. High dimensionality: feature vectors can be of high dimensionality. Indeed, features are often captured from several channels and time segments.
3. Time information: BCI should contain time information as brain activity patterns.
4. Non stationarity: BCI features are non-stationary since EEG signals may vary rapidly over time.

2.2.2 Temporal methods

2.2.2.1 Peak detection

The peak detection method is a really straightforward and basic feature extraction method. This method simply determines the maximum or minimum value in a specific time block and uses that value as a feature.

2.2.3 Spectral methods

2.2.3.1 Band power

The band power method is one of the most intuitive and simplest methods for tracking amplitude modulations in a signal. The different steps, shown in Fig.2.2, are the following ones:

1. The original signal is captured.
2. The first step is to isolate the frequecy of interest by applying a bandpass filter to the signal.
3. Next, the signal is squared or its absolute value is taken in order to have a purely positive signal.
4. Finally, the adjacent peaks are filtered using a low-pass filter.

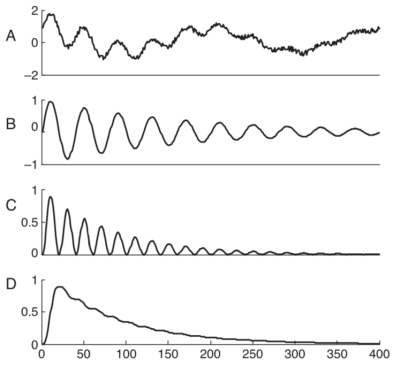


Figure 2.2: Method to extract the band power from a signal. Figure taken from [1].

The features vector will then be constituted of the integrals of the timeblocks of the smoothed signal.

2.2.3.2 Fast Fourier Transform

The Fast Fourier Transform (FFT) is an effective implementation of the discrete Fourier Transform. The FFT represents the frequency spectrum of a digital signal. The FFT takes a digital signal with N samples and produces N frequency samples uniformly spaced over the frequency range $[\pm F_s/2]$. A simple estimate of the power spectrum can be obtained by simply squaring the FFT magnitude. The power is then evaluated as the feature. Thanks to its ease of impementation, the FFT is often used as a model for comparing other spectral methods.

2.2.3.3 Autoregressive

Autoregressive (AR) modeling is an alternative to Fourier-base methods. It assumes that an EEG signal is equivalent to a signal generated by filtering white noise with an IIR filter. *White noise* is random noise that has the unique property of being uncorrelated to any delayed version of the signal itself. The method is based on the computation of the IIR filter weights in order to shape the spectrum to match the actual spectrum of the signal being analyzed. As the relevant information is contained in the weights, it is common to use the weights themselves as features. One of the issues with AR modeling is that the accuracy of the model depends on the model order. A too low model order can have negative effects on the spectrum, whereas a too large order can create artifical peaks in the spectrum. It is important to note that the order of the model depends a lot on the spectrum of the signal and its sampling rate. If the sample frequency increases, the model order should increase as well to avoid distortions of the spectrum. On the contrary, one of the advantages is that because the IIR filter defines the signal spectrum, AR modeling can achieve higher spectral resolution than the FFT. Moreover, the IIR filter structure can accurately produce model spectra with sharp and distinct peaks, which is often the case in the case of EEG signals.

2.2.4 Time frequency methods

2.2.4.1 Wavelets Transfrom

The wavelets analysis is a time-frequency method that solves one major drawback of other methods. Indeed, in other methods, the temporal and spectral resolution are very dependent on the model order, length of the segment and other parameters. Wavelets analysis solves this problem by doing a time-frequency representation of the signal. The joint resolution makes it a very good candidate for BMI applications and the extraction of details in EEG signals, which cannot be obtained by the FFT method for example, as explained in [43]. This paragraph

first compares wavelet analysis with short-time Fourier transform, an other time/frequency method. Then, the working principle is presented and several analysis windows are described and eventually, the discrete wavelet transform is introduced. When applying time/frequency methods one has to bear in mind that a time/frequency resolution trade-off exist for such methods. This described by Heisenberg :

$$\Delta t \Delta \omega \geq \frac{1}{2} \tag{2.5}$$

However the major advantage that wavelet analysis has over other time/frequency methods such as short-time Fourier transform (STFT) is the particular choice of time/frequency resolution. Indeed, a good time resolution is chosen for high frequencies while for slowly varying signals, a good frequency resolution is chosen. Fig.2.3 depicts the particular time/frequency resolution used in wavelet analysis and compare it with the other well-known time/frequency method. One may notice that the STFT uses a uniform time/frequency resolution, which is very inconvenient for high-frequency signals that receive the same temporal resolution as low-frequency signals.

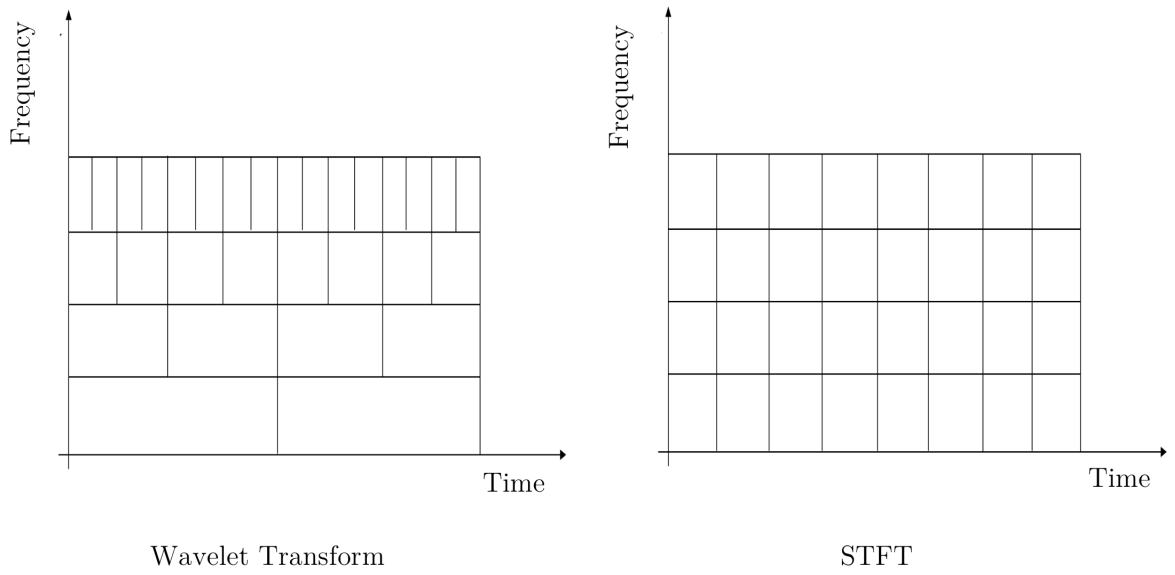


Figure 2.3: Time/frequency resolution of wavelet transform. Time/frequency resolution of short-time Fourier transform.

Wavelet analysis is based on the decomposition of a signal into a linear combination of a particular set of wavelets. To create such a wavelet set, one must select a **mother wavelet** and perform translation and dilatation of the mother wavelet. Wavelets have a finite-duration, also known as the *support*. The mother wavelet function $\Psi_{a,b}(t)$ is given as :

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi(\frac{t-b}{a}) \tag{2.6}$$

where a and b are the scaling factor and the shifting factor respectively. There are various mother wavelets, each one having specific characteristics and mathematical properties. One may cite Haar, Deaubechies, Meyer, Morlet, Symlet and Coiflet. Fig.2.4 depicts these 6 different wavelet types and table 2.1 compares advantages and drawbacks of these six wavelet type. Different properties make a good wavelet

1. Orthogonal wavelets are usually preferred because they preserve signal energy and therefore information within the signal.

2. Symmetrical wavelets are appreciated because they are reversible.

3. Compact support wavelets (less coefficient/taps) lead to processing power requirement and higher computational efficiency.

4. Having lots of vanishing moments. This allows to obtain well-defined detailed signals form the wavelet decomposition. More details can be found in [78] [79].

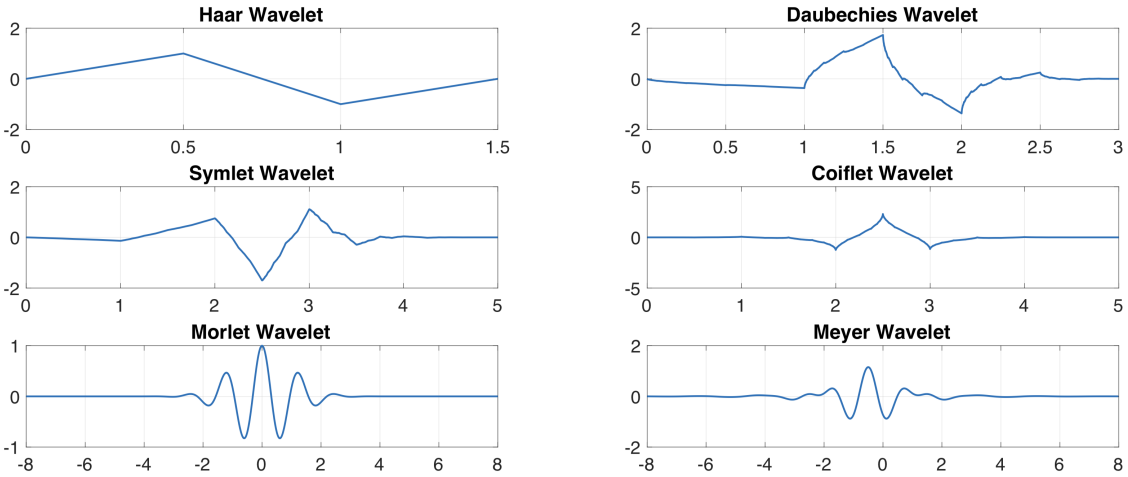


Figure 2.4: Different wavelet types. Certain wavelets perform better than other under particular circumstances

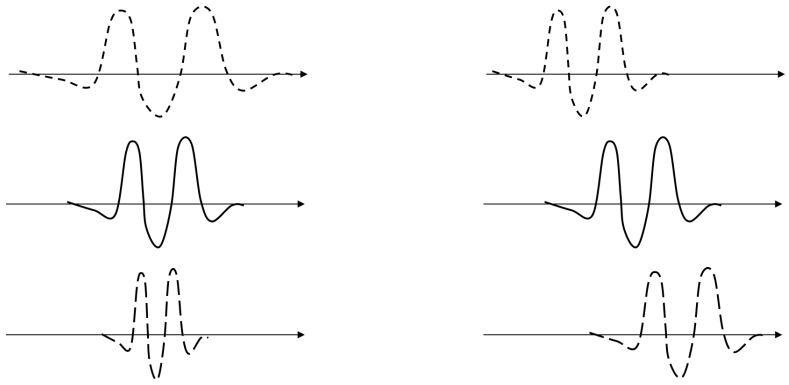
Among different wavelet types, some of them are more appropriate for a BCI design. Indeed, given the nature of EEG signals, which have a very weak amplitude, one need to preserve the energy contained within the signal. Therefore one will particularly emphasize the choice of orthogonal wavelets for a BCI design. Once one has selected a mother wavelet, one may define the continuous wavelet transform :

$$X_\omega(a,b) = \int_{-\infty}^{+\infty} x(t)\Psi^*_{a,b}(t)dt \tag{2.7}$$

where $x(t)$ is the input signal and $\Psi^*_{a,b}(t)$ is the complex conjugate of the mother wavelet. The parameters a and b are the scaling and shifting factors, respectively. For each a and b set, a different wavelet transform exists, leading to $\mathbb{R}^2$ different possibilities of wavelet transforms for the same $x(t)$ signal. Fig.2.5 depicts the concept of scaling and shifting the mother wavelet for obtaining different transforms. One may understand, based on Eq.2.7, the reason why such a transform performs better than the Fourier transform. Indeed, a Fourier transform represents a signal as a sum of sines. A sinus function is not localized in time compared to a wavelet with compact support. Therefore, one cannot represent a transient signal accurately with infinite support basis functions such as the sinus function. Likewise the discrete Fourier transform that can be implemented in software, a discrete wavelet transform (DWT) exists and is introduced in the next paragraph.

	Advantages (+)	**Drawbacks (-)**	**Comments**
Haar	1. Simplet type of wavelet (only two taps). 2. Low processing time. 3. The Haar wavelet is an orthogonal wavelet	1. Not smooth (continuous but not differentiable)	Special case of Daubechies wavlelet.
Daubechies	1. Most used wavelet type. 2. Smooth wavelet support (especially for higher order). 3. Orthogonal and compact support abilities.	1. 8-taps filter that requires more computationnal power. 2. Not necessarily symmetric.	Several Daubechies windows exist (from `db1` to `db45`)
Symlet	1. Least asymmetric wavelets 2. Maximum number of vanishing moments for a given compact support	1. Still not perfectly symmetrical.	Symlets are designed based on Daubechies.
Coiflet	1. Wavelets have nearly symmetrical distribution. 2. They have a maximum number of vanishing moments (likewise symlets).	1. Still not perfectly symmetrical. 2. No equation has been established for designing a generic Coiflet wavlet.	Several Coiflet windows exist. Each one with a different support and a different number of vanishing moments.
Meyer	1.The Meyer wavelet is an orthogonal and symmetrical wavelet. 2. Indefinitely differentiable with infinite support.	1. Large support (more than 8 taps)	/
Mortelet	1.The Mortelet wavelet is an orthogonal and symmetrical wavelet. 2. Lots of vanishing moments 3. Indefinitely differentiable.	1. One of the largest support. This wavelet is definitely not compact	Properties close to Meyer wavelet but with a larger support.

Table 2.1: Comparison of 6 different wavelet types.



Scaling operation Shifting operation

Figure 2.5: Shifting and scaling operations of the mother wavelet.

The output of the Fast Fourier Transform can be realized from a set of parallel bandpass filters (a so-called filter bank), with each filter centered at uniform frequency intervals. In contrast, wavelet analysis designs a filter bank to achieve an improved time-frequency resolution. The mother wavelet is thus used to construct a template for each temporal FIR bandpass filter in the filter bank.

The DWT transform is applied several times in a row. Each step corresponds to one level of decomposition and is exactly applied in the same way as the one-level transform. One will thus first introduce the one-level transform. The DWT of a signal $x(t)$ is obtained by passing the input signal through a set of two filters and a bank of 2-fold downsamplers. Among the filters, one is a high-pass filter $h[n]$ and the other one is a low-pass filter $g[n]$. These filters are quadrature mirror images one to another [3] ($g[L-1-n] = (-1)^n h[n]$). Outputs of filters are described by :

$$y_{\text{low}}[n] = x[n] * g[n] \sum_{k=-\infty}^{k=+\infty} x[k]g[n-k] \tag{2.8}$$

$$y_{\text{high}}[n] = x[n] * h[n] \sum_{k=-\infty}^{k=+\infty} x[k]h[n-k] \tag{2.9}$$

Afterwards, both outputs are decimated by a 2-fold downsamplers and filtered again by an anti-aliasing filters. The low-pass filter output is called the **approximate coefficient** while the high-pass filter output is known as the **detailed coefficient**. Theses names are related to the fact that details are captured for high frequency with a good localization in time. The high-pass filter is in fact used to extract a layer of detail from the signal, and the low-pass filter yields a smoothed version of the signal without that detail content. It is thus called the approximation coefficient.

Next, the multilevel DWT is a generalization of the aforementioned operation. Indeed, the basic scheme (2 filters and 2 dwonsamplers) is repeated using the approximation signal as input of the filterbank. Fig.2.6 depicts the multilevel DWT decomposition.

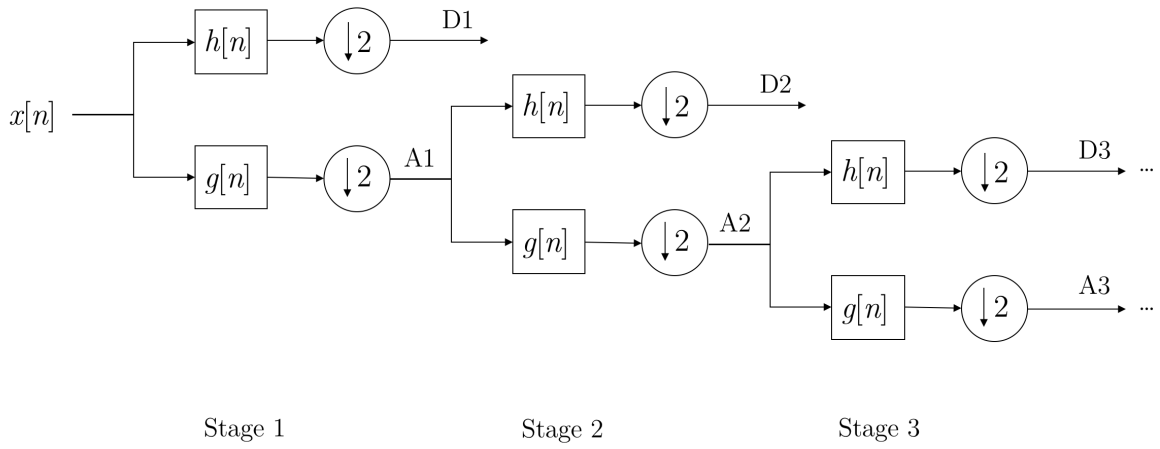


Figure 2.6: Sub-band decomposition of DWT filterbank; $h[n]$ is the high-pass filter, $g[n]$ the low-pass filter.

The wavelet based feature extraction has proven to be one of the best methods compared to any other transform based method, which is why it will is implemented on our experimental board. The Wavelets method will thus be further described in chapter5.

2.3 Classification

During the previous steps, the processing chain purpose was to establish a feature vector that contains more informative and discriminative data, based on electrical signals recorded from the brain. Generally, these features vectors cannot directly communicate the BCI user intent. Therefore the purpose of the classification is to assign a class to each of these features vectors. Each class is related to one command that can be recognized by the BCI. In this work, there are three different classes, each one corresponding to

1. A left arm movement.
2. A right arm movement.
3. No movement or 'rest'.

This part of the processing chain is slightly different from previous ones because it requires a training phase, using a training data set, before recognizing and classifying new features vectors. There exists different categories of machine learning processes. Indeed, the learning can be either online/offline, unsupervised/supervised learning.

2.3.1 Classification characteristics

2.3.1.1 Online/Offline

In offline learning, a training data set is recorded before the training process. One will use this set of training data for training the classifier. Generally, the offline training is performed on a powerful computer and only coefficients obtained during the training process are used for classifying new data (on a micro-controller or FPGA e.g.). In online training, the model is continuously updated when new batches of data are recorded. It does not involve a complete retraining operation but an update of the model weights based on new observations. These two methods have pros and cons.

	Advantages (+)	**Drawbacks (-)**
Offline Training	1. The classifier can process information directly because the model is already known. 2. Classification can be executed with low computational power and low memory processors (FPGA or micro-controller).	1. The model is sometimes out-of-date and needs to be retrained. 2. It cannot adapt to new operation conditions. 3. Data are registered in a fixed environment and might be different from the data captured in a everyday life environment. 4. It does not the progressive mental evolution of the user (in terms of fatigue /pain/...).
Online Training	1. Each data is used to train the update the model. Adaptive behavior. 2. Generally more accurate than Offline training. 3. possibility to create a more consistent model based on more batches of data.	1. Requires high computational power and large memory.

To summarize, offline training is unavoidable when implementing a real-time classification on a low-power CPU (such as an FPGA). However, such a training does not reflect real-time conditions and the model may rapidly become out-of-date. Therefore, a frequent training is required. **For this work, an offline classification is chosen.**

2.3.1.2 Unsupervised/supervised

In unsupervised learning one does not have any target or classes to predict. It is used for clustering population in different groups. The purpose is to model the underlying structure behind the data. The term *unsupervised* means in that particular case that there is no correct or wrong answers and that the algorithm is free to discover interesting structures within the data. On The other hand, *supervised* learning is used when one has a training set where **predictors** have inputs vectors X and expected answers/predictions Y. Each unit of training data (also called observation set) is in fact composed of a feature vector and its expected output class (also called training label). The purpose is to establish the mapping that exists between the input vectors and the expected outputs predictions $Y = f(X)$. **For a BCI design, one must select a supervised learning algorithm.**

Notice that there also exist semi-supervised machine learning techniques that can be used on some micro-controller or FPGA. These techniques are based on an offline training followed by memory efficient adaptive steps. This allows to obtain a model less influenced by non-stationary of brain signals. However such techniques still can not be used with the material employed in this work.

Once the model is trained, one must evaluate it. This process is performed in two steps :

1. Measure of the accuracy of the model using a cost function, e.g. the mean-squared error. The smaller the cots function is, the more accurate the model is.

2. A test data set of features vectors and expected class goes trough the classifier. The classification rate is established. Notice that this test set must be different from the training set.

This section is organized according to the following structure : description of a binary classifier, main problems that a classifier may face, description of some classifiers and comparison of these

classifiers.

2.3.2 The classifier

Let us take the simplest case where only two classes are assigned to objects, as described in [45]. Such a classifier is called a *binary classifier*. The goal is to estimate a function $f : \mathbb{R}^N \longrightarrow \{-1, +1\}$ from a given function class F using training data generated according to an unknown probability distribution $P(x, y)$ with

$$(x_1, y_1), ..., (x_k, y_k) \in \mathbb{R}^N \times \{-1, +1\} \tag{2.10}$$

and such that f will correctly classify a new pair (x, y). A very simple example would be to assign the object to the class $+1$ if $f(x) \geq 0$ and to the class -1 otherwise.

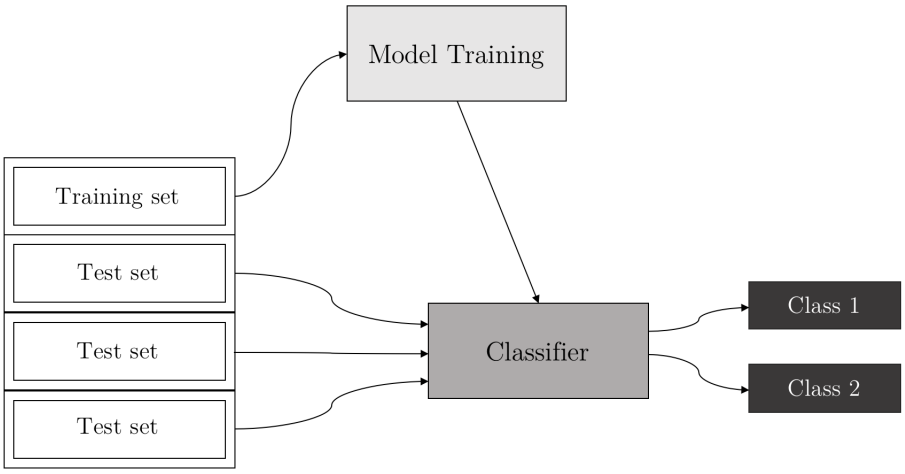


Figure 2.7: Classification scheme with the training and the test set. The classifier is a binary classifier with two possible output classes.

Different types of classifiers are used in the field of BCI. Some types of classifiers can be found in [45] and [46]. This section explores and explains some of the most popular algorithms and their basic properties.

2.3.3 Classifiers major problems

There exist two main problems : The curse of dimensionnality and the bias-variance trade-off. These two problems are really well explained in the previous Master Thesis ([4],[3]) and will therefore not be recalled here.

2.3.4 Linear Classifiers

Linear classifiers are discriminant algorithms using linear functions to separate classes. Their implementation is quite simple and they may therefore be the most popular algorithms for classification in BMI applications. The two most known algorithms for linear classifiers are Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM).

2.3.4.1 Linear Discriminant Analysis (LDA)

The LDA uses hyperplanes to separate the data in different classes, as shown in Fig.2.8.

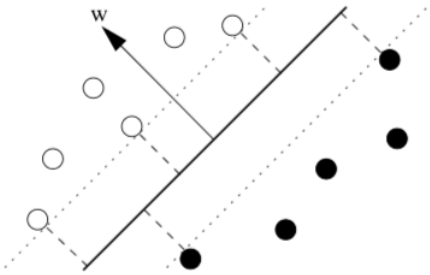


Figure 2.8: LDA classifier

The hyperplane is obtained by looking for the projection that maximizes the distance between the two classes and minimizes the interclass variance. This technique has a very low computational cost, which makes it very suitable for online BCI systems. It is also very simple to use and generally provides good results. However the use of hyperplanes can lead to poor results for complex feature vectors.

2.3.4.2 Support Vector Machines (SVM)

SVM is similar to LDA in the sense that it uses a discrimant hyperplane to separate classes. However, it selects a hyperplane that maximizes the margin, i.e. the distance from the nearest training points.

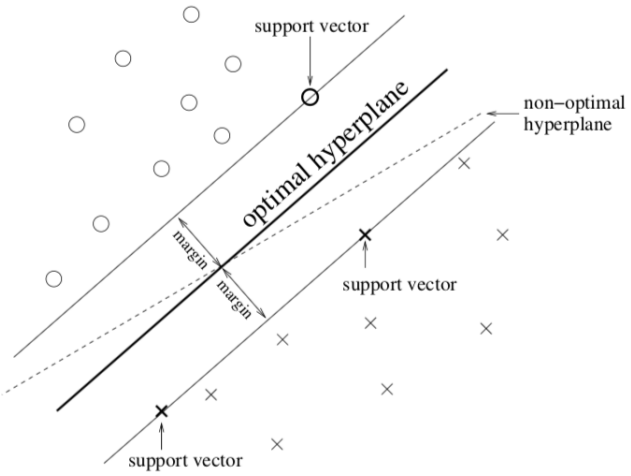


Figure 2.9: SVM classifier

This classifier is widely used and has been applied to a large number of BCI problems. However, one disadvantage of this classifier is that it can quickly become memory consuming when the number of support vectors increases.

2.3.5 Nonlinear Bayesian classifiers

These classifiers produce nonlinear decision boundaries. They are also generative, which enables them to perform more efficient rejection of uncertain samples than discriminative classifiers. This type of classifiers is however less used than linear classifiers or Neural Networks in the literature.

2.3.5.1 Bayes quadratic

Bayesian classifiers aim at assigning the class to the feature vector following the hihest probability. Bayesian classifiers follow a conditional probability model using the Bayes rule to compute the so called *posterior probability* that a feature vector corresponds to a given class. The Bayes rule computes the probability that a feature vector x belongs to a class C_k and can be expressed as

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)} \tag{2.11}$$

Bayes quadratic assumes a normal distribution of data. The term *quadratic* comes from the fact that quadratic boundaries are done by the classifier.

2.3.6 Hidden Markov Model (HMM)

Hidden Markov Model is a very famous classifier in the field of speech recognition essentially and has not been used widely in the field of BCI. It is now emerging in the field of handwritten word recognition [48]. A HMM based classifier usually incorporates one individual HMM for each class where each HMM is built up from a set of states. The input of such a classifier is a sequence of feature vectors $x_1, ..., x_n$. In the decoding phase, the input sequence is then mapped to a sequence of states $s_{i1}, ..., s_{in}$ and for each mapping a likelihood value is defined by the HMM. The optimal mapping is the one maximizing the likelihood. The likelihood of the optimal path is the score of the sequence of feature vectors for the considered HMM. The class corresponding to the HMM with the highest score will be assigned as the output class.

2.3.7 Nearest Neighbors classifiers

These classifiers consist in assigning a feature vector to a class according to its nearest neighbor. The neighbor can be a feature vector from the training set as in the case of k Nearest Neighbors (kNN). These classifiers are quite common in the field of BCI [72] [73].

2.3.7.1 k Nearest Neighbors (kNN)

This technique is based on the assignment of the dominant class among the k nearest neighbors in the training set to a unseen point. In the field of BCI, a metric distance is often used . If k tends to be high and enough training samples are present, this classifier can approximate any function and produce nonlinear decision boundaries.

2.3.7.2 Mahalanobis distance

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

$$d_c(x) = \sqrt{(x - \mu_c)M_c^{-1}(x - \mu_c)^T} \tag{2.12}$$

This is a simple but robust classifier, that proved to be suited for different BCI applications [70] [71]. However, this classifier is not widely used in the literature

2.3.8 Neural Network

Artificial neural networks (ANN) are computing/predicting/classifying systems inspired by biological neural networks. Basically it is a set of connected units, also known as **neurons** that can transmit signal to other neurons when a connection is established between these neurons. In commons implementation, an ANN is made of 4 elements :

1. Neurons are units that receive signals and that can modify them by adding offsets to these signals.

2. Connections make a link between neurons. They are characterized by a weight that can either emphasize the signal coming from a particular neuron or decrease its amplitude.

3. Layers consist of a set of neurons. Within a layer, neurons are not connected together but connected to neurons from another layer. A neural newtork is generally made of an input layer, an output layer and zero, one or several hidden layers in-between.

4. An activation function, after each layer, that takes neurons from one layer and send them to the next one trough the function. This function is generally non linear with respect to the input. There exists several activation functions although the **sigmoïd** function is the most popular one.

Fig.2.10 depicts a basic neural network with one input layer (green layer), one hidden layer (blue layer) and one output layer (red layer). The input is a feature vector $x \in \mathbb{R}^{4x1}$ while the network is made of 5 neurons and one output. Activation function are not depicted but one may imagine such a function downstream of the hidden layer.

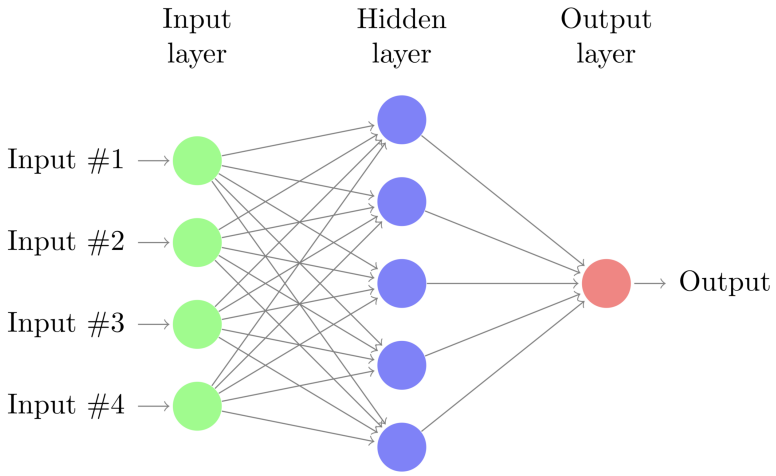


Figure 2.10: Neural Network illustration with 4 layers.

Together with linear classifiers, Neural Networks (NN) are the category of classifiers mostly used in BCI research. Several types of Neural Network classifiers exist, the most common one being the Feedforward neural network with one or several hidden unit. also known as MultiLayer Perceptron (MLP). However, other types exist such as Convolutionnal Neural Network (CNN) or Recurrent Neural Network (RNN). After this brief introduction about neural networks, one may present feedforward neural network.

2.3.8.1 Feedforward Neural Network

Feedforward neural networks are characterized by unidirectional connections between the neurons and layers. The connections do not form a cycle. The *feedforward* term is used because signals only travel forward in the network, from the input nodes to the output nodes trough hidden layers. Feedforward networks are essentially used for supervised learning and assume that the data to be learned is neither time-dependant nor sequential. There exist two types of feedforward neural network : (1) Single-layer perceptron and (2) Multilayer perceptron.

Single-layer perceptrons do not contain any hidden units (Fig.2.10 without blue layer). For

example, one may consider a unique perceptron with n-dimensional input feature vector $\vec{x}$, a n-dimensional weight vector $\vec{w}$ and a scalar bias/offset b and a sigmoïd activation function :

$$o = f(\vec{w} \cdot \vec{x} + b) \qquad \text{with } f(u) = \frac{1}{1 + \exp(-u)} \tag{2.13}$$

Fig.2.11 depicts both the single-layer one-output perceptron (left) and the sigmoïd activation function (right). A single-layer perceptron may be understood as a linear combination of the input feature vector $\vec{x}$ using a weight vector $\vec{w}$ and a bias b. The resulting combination goes trough an activation function. The result is the output perceptron that has a value between 0 and 1. Classification boundaries are defined based on perceptron values.

In order for a perceptron to learn a correct classification, one training set is required: $X_{\text{training}} = \{\vec{x}, y\}_N$ where $\vec{x}$ is a feature vector among N training feature vectors and y is the class that corresponds to the given feature vector. At this point, one must compute weights $\vec{w}$ and bias b to obtain a good classification rate, i.e. such that the output perceptron o is close to the corresponding output y.

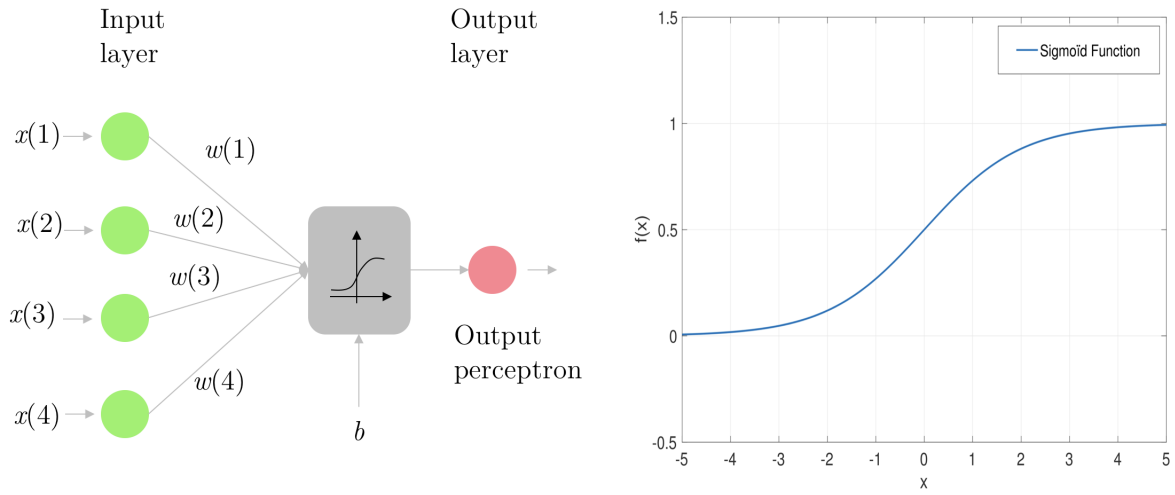


Figure 2.11: Single-layer perceptron with a unique output perceptron and a sigmoïd function as activation function. Sigmoïd activation function has co-domain between 0 and 1.

The learning process requires a mathematical formulation. One may use a cost function that computes the distance between the output perceptron o and the expected outputs y of the training set. A commonly used cost function is the **mean-squared error** (MSE)

$$MSE(X_{\text{training}}) = \frac{1}{2N}\sum_{k=1}^{N}(o_k - y_k)^2 = \frac{1}{2N}\sum_{k=1}^{N}(f(\vec{w} \cdot \vec{x_k} + b)o_k) - y_k)^2 \tag{2.14}$$

Minimizing this cost function corresponds to an unconstrained optimization problem that can be solved using a line-search method based on gradient method or Newton method. Notice that this problem is, given the non-linear activation function, not a quadratic program (QP) which means that convexity is not necessarily ensured. Therefore one may not be sure of converging towards a global minimum. Nevertheless, one may define the gradient method :

$$\vec{w_{i+1}} = \vec{w_i} + \alpha_{w,i} d_{w,i} \qquad b_{i+1} = b_i + \alpha_{b,i} d_{b,i} \tag{2.15}$$

Where i is the iteration index, $\alpha_{w,i}$ and $\alpha_{b,i}$ are the step lengths. $d_{b,i}$ and $d_{w,i}$ are the directions of descent. In the aforementioned equation 2.15, one may define using a **gradient method** :

$$d_{w,i} = \frac{\partial E(X_{\text{training}})}{\partial \vec{w}_i} \qquad d_{b,i} = \frac{\partial E(X_{\text{training}})}{\partial b_i} \tag{2.16}$$

This defines a training protocol for the single-layer neural network.

1. Compute, using the training set, all the perceptrons values o_k with $k = 0...N$
2. Compute the partial derivatives of the error function with respect to $\vec{w}$ and b.
3. Select a step size α that ensures a monotonic decrease of the cost function. This step size can be either fixed or data-dependant.
4. Update weight and bias values according to the line-search method.
5. Repeat the process until the cost function converges to a minimum value.

To summarize, a single-layer neural network is made of both one input and output layer. An activation function precedes the output layer and produces a perceptron used for classing the input feature vectors. Notice also that **single-layer perceptrons are linear classifiers**. That means that they can only learn linearly separable patterns. For that reason, multilayer perceptrons have been introduced.

A multilayer perceptron (MLP) is composed of at least three layers of neurons: an input layer, one or several hidden layers, and an output layer [46]. Unlike single-layer perceptrons, they can learn non-linearly separable pattern as well. Fig. 2.10 depicts a MLP with one hidden layer but one may imagine a MLP with several hidden layers. Due to its feedforward organization, the output of the hidden layer depends only on the input layer. When composed of enough neurons and layers, MLP can approximate any continuous function. Moreover, they can classify any number of classes, and this makes them very flexible classifiers adaptable to a large variety of problems.

Likewise single-layer perceptrons, MLP Neural Networks are generally trained by standard backpropagation algorithm based on a cost function which can be the mean-squared error [47]. More details about MLP can be found in chapter 6.

2.3.9 Decision trees

A decision tree is a tree in which each branch node represents a choice between a number of alternatives and each leaf node represents a decision. It is a type of supervised learning algorithm that is mostly used in classification problems. A decision tree is illustrated in Fig.2.12.

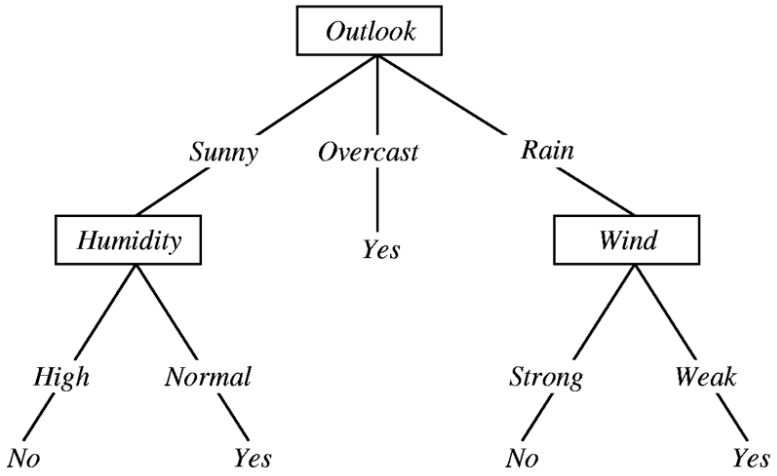


Figure 2.12: Decision tree for an example of weather classification.

2.4 Conclusion

A BCI is composed of a processing chain through which the EEG signals are processed in order to give a relevant information at the end of the chain. In this Chapter, the whole processing chain has been visited step by step. First, the preprocessing step was explained, with the frequency and spatial filtering. We saw that different types of filters and methods existed to delete artifacts and irrelevant information contained in the raw signal. We also saw that the feature extraction step could be implemented with different methods. Indeed, there exist temporal, frequency and time frequency methods.

Chapter 3

BCI Hardware and some improvements

After having established a first draft of the BCI specifications in Chapter 1, one must specify the hardware that will be used for designing the BCI. The hardware consists mainly in three parts: the BMI board made of two ADCs and one FPGA for respectively acquiring and processing EEG signals, the helmet used for positioning electrodes and eventually the electrodes themselves that measure EEG signals from the scalp.

As a reminder, here are the main BCI specifications, as described in Chapter 1.

1. The BCI is active, non-invasive and acquires EEG signals.
2. Recorded signals are based on electrical brain rhythms (μ essentially) linked to event-related potentials.
3. The BCI is able to recognize 3 different motor tasks : Right-hand movement, left-hand movement and rest state.
4. The BCI works in real-time and is synchronous. It means that it works sequentially in 2 steps : It first records EEG signals when a user executes a given motor task during a certain duration (defined in chapter 4) and then it processes these raw signals to predict the user's intent. Notice that the user must perform only one motor task during a given record.

This chapter is organized as follows: in section 3.1, the BCI board defined in [3] is briefly explained. Then the hardware improvements, in terms of helmet and electrodes, are presented in section 3.2. In section 3.3, one will briefly discuss about the choice for the number of electrodes used for signal acquisition and eventually, in section 3.4, hardware specifications are precised.

3.1 BCI board

The BCI board developed in 2016 by Maxime Schrame and Benjamin Delhaye [3] aims at realizing a real-time and portable device. Moreover the system is designed for a long-term development purpose. Therefore it is entirely programmable, using different types of algorithms for each stage of the processing chain. A general overview of the BCI board is depicted in Fig.3.1. One will briefly explain the different blocks of this diagram.

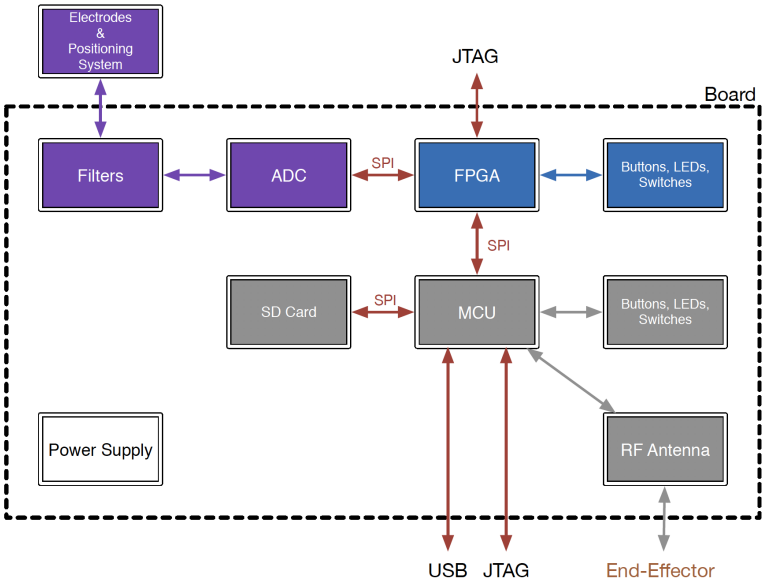


Figure 3.1: General overview of BCI board.

3.1.1 The board.

This is the structure that contains all the blocks (FPGA, ADCs). This board is designed as a printed-circuit board (see Fig.1) where all functional blocks are soldered onto. This PCB has several specifications :

1. This is a 4-layers PCB where units are disposed with either a SMD or trough holes configuration.
2. The power supply is 5V USB with a maximum supply current of 500 [mA].
3. The size of the board is 14 x 7 [cm], which makes the BCI portable.
4. The PCB has several interfaces : USB, JTAG UART, LEDs, buttons, switches, micro-SD slot.

The PCB connects the different functional blocks described below.

3.1.2 Filter

The block that follows the acquisition system (helmet and electrodes) is the input filter. This is in fact a bank of first-order lowpass and passive RC filters. The main advantage of using such filters is the very low noise added to EEG signals coming from the electrodes. The cut-off frequency is chosen to be 6.78 [kHz] which is lower than the $f_{\text{clock,ADC}}/4 = 512$ [kHz] (required by the ADCs for avoiding spectral folding and aliasing. More details in [3]).

3.1.3 ADCs

These blocks aim at digitizing analog EEG signals coming from filter stage. This part is critical because EEG signals have a very low amplitude. They should not be corrupted by additional noise. Therefore special ADCs dedicated to physiological signals acquisition must be chosen in order to obtain the best possible signal-to-noise ratio.

The ADS1299 from TI is selected for this purpose. This a 24-bits Sigma-Delta ADC devoted to EEG signals acquisition. This ADC has 8 input channels. It is also characterized by a low

input-referred noise and a high CMRR. Moreover, it has a Serial Peripheral Interface (SPI) which is convenient for communicating with downstream blocks.

These ADCs also have other interesting features: a common reference can be chosen for each input such that each EEG signal has the same voltage reference. Furthermore, a lead-off detection allows to determine whether electrodes are correctly in contact with the scalp. Eventually, a bias drive is available. It helps to reduce the DC voltage between measurement electrodes and the reference electrode. More details can be found in the datasheet [52].

3.1.4 FPGA

This represents the processing unit of the BCI board. The complete processing chain will be implemented within this block. Among all possible FPGA models, the `Altera MAX1050` is selected. It consists of 50kLEs (logic elements), 101 I/Os pins and a large flash memory (1.6 million of memory bits) which is desirable for implementing softcores. A 52 [MHz] clock is provided to the FPGA in this work. Notice that the FPGA has a built-in PLL (phase-locked loop) which allows to overlock the clock frequency if needed.

Moreover, it has several embedded multipliers that can be used for speeding up sofware multiplication. The FPGA will be programmed in `Verilog` using `Quartus` as development tool.

Furthermore, the FPGA is connected to an host computer via a JTAG connection. This connection allows the FPGA to be programmed but also defined a communication channel between the FPGA and the host computer when running code on a softcore. More details can be found in the datasheet [53].

3.1.5 MCU

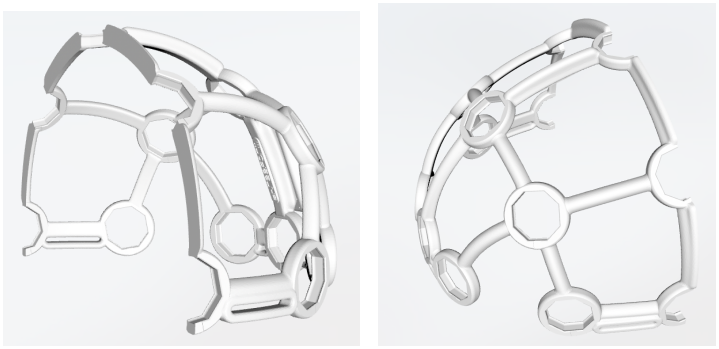
The selected microcontroller is CC430 from TI. It is connected to the FPGA by a SPI interface. It can be directly programmed in C/C++ and offers some additional computational power and embedded memory (4 kB of SRAM and 32 kB of flash memory). Moreover, it can be connected to a micro-SD card offering much more embedded memory (up to 64 MB). More details can be found in the datasheet [54].

3.1.6 Buttons, switches and LEDs

Two buttons are directly connected to the FPGA. The *On* button is generally used for reseting the FPGA and the ADCs. Four LEDs are connected to the FPGA too. This allows to communicate the current state within the processing chain of the FPGA to the user. Finally four switches are used for defining several BCI functioning modes.

3.2 Hardware improvements

In the previous section, the BCI board is briefly described. However, the acquisition material connected to the board is not presented. This is the purpose of this section. First, let us recall that the the *Ultra Cortex Mark III* version of the open source helmet has been used in the previous years [3] [4]. This helmet may contain up to 22 electrodes. In Fig.3.2, the frame of the helmet is depicted.

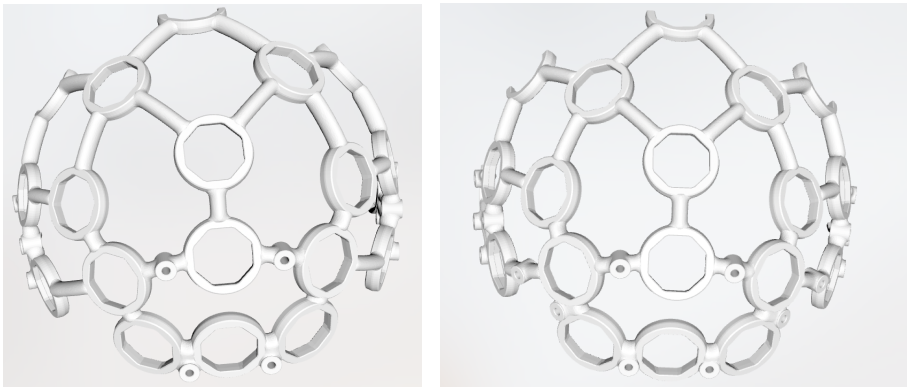


(a) Back of the *Ultra Cortex Mark III* helmet

(b) Front of the *Ultra Cortex Mark III* helmet

Figure 3.2: *Ultra Cortex Mark III*

Following the advices given in [3] and [4], one has chosen to use the next version of the helmet, the *Ultra Cortex Mark IV*, which is displayed in Fig.3.3. At first sight, one may see that the last version supports more electrodes (35 in total).

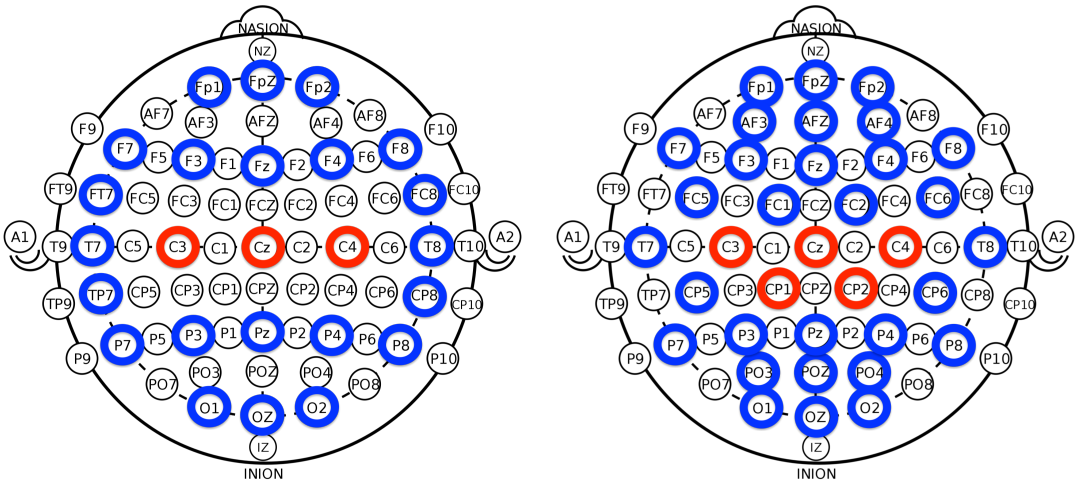


(a) Back of the *Ultra Cortex Mark IV* helmet.

(b) Front of the *Ultra Cortex Mark IV* helmet.

Figure 3.3: *Ultra Cortex Mark IV*

The electrode fastening system is the same for the two helmets, but the electrodes placement is quite different, as shown in Fig.3.4. The electrodes indicated in red are the ones that are present on the helmets. The blue circles correspond to free slots where electrodes can be added.



(a) *Ultra Cortex Mark III* helmet with 3 electrodes. (b) *Ultra Cortex Mark IV* helmet with 5 electrodes.

Figure 3.4: Comparison between the electrodes placement of the two different helmets.

As defined in the general specifications, the BCI system measures encephalic activity and brain rhythms linked to event-related potentials over the sensory motor cortex when a motor task is executed. One must therefore dispose electrodes over this area. The corresponding electrodes are C3, C4 and Cz. Until now, three electrodes have been used in the development BCI in [3] [4]. The decision to extend the number of electrodes from 3 to 5 has been made. Indeed, electrodes CP1 and CP2 were added to the new version of the helmet in order to collect more signals. These two additional signals can provide more information and allow to implement better spatial filters.

Besides the new helmet, one has decided to select new electrodes. Indeed, as explained in [55], reusable electrodes have a limited lifespan and must be changed after several usage hours. One has decided to select standard dry and reusable AgCl electrodes from *Florida Research Instruments* [56]. One has also select additional wet electrodes from *Florida Research Instruments* [56] too in order to have the ability to study two acquisition systems in parallel. Fig.3.5 depicts the two types of electrodes used in this work.

(a) Florida Research Instruments wet electrodes (b) Florida Research Instruments short-spike dry electrodes

Figure 3.5: Two types of electrodes used in this work.

3.3 Number of electrodes

In previous works [3] [4], authors have used three electrodes : C3, Cz and C4. In this work two additional electrodes (Cp1 and Cp2) are used as described earlier in this chapter. The choice of number of electrodes is based on the work of W. Speier et al. in [57]. Indeed, they have developed a particular algorithm based on P300 signals and have investigated the correlation between the classification rate and the number of electrodes in order to determine the optimal electrodes number for a given application. They have shown, for three subjects, that the classification rate increases with the number of electrodes. This is depicted in Fig.3.6.

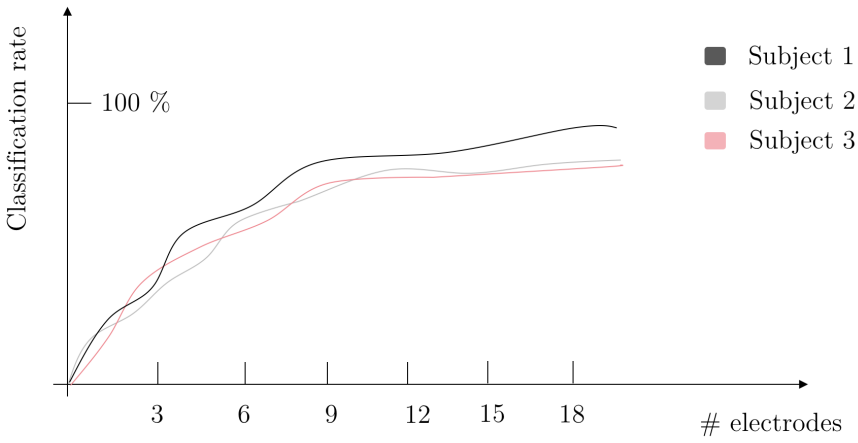


Figure 3.6: Evolution of the classification rate with respect to the number of electrodes.

Based on these results, one needs to increase the number of electrodes as far as possible. However, using the FPGA MAX1050 allows the author to implement their processing chain (extensively explained in chapter 5) with at most 5 electrodes because of the limited memory of the FPGA (more details in chapter 5). One has therefore choose to use the 5 electrodes located above the sensory-motor area.

3.4 Conclusion and BCI hardware specifications

To summarize, this chapter reintroduces briefly the BCI board designed in [3]. It explains next the new *Ultra Cortex Mark IV* helmet that can host more electrodes, especially over the sensory-motor area. In addition, new electrodes have been purchased. These electrodes are either dry electrodes that can be used directly or wet electrodes that requires the usage of a conductive paste. Eventually, the BCI hardware specifications are defined. Fig.3.7 depicts the new helmet with 5 electrodes connected.

Figure 3.7: *Ultra Cortex Mark IV* helmet with 5 electrodes connected : C3, Cz, C4, Cp1 and Cp2.

Next chapter will deal with software considerations, trough the `MATLAB` tool, and will aim at defining BCI software specifications and timing specifications (duration of a signal record and time required in order for the BCI to make a prediction of the user's intent). One may now establish, as shown in Fig.3.8 the hardware specifications of BCI the system. These specification will be added to specifications defined in chapter 2 and recalled at the beginning of this chapter.

1. The BCI system will use the BCI board designed in [3].
2. The BCI system will use a 3D-printed *Ultra Cortex Mark IV* as helmet.
3. The BCI system will acquire EEG signals over 5 electrodes : C3, Cz, C4, Cp1 and Cp2. This choice is justified as follows : recording more signals will lead to a better classification rate. However using more than 5 electrodes is not possible with the FPGA MAX1050, which is limited by the number of logic elements and the memory.
4. Given that this work aims at developing a BCI, one will not connect the BCI to an end-effector to apply a command. One will instead use the host computer connected to the FPGA via JTAG to print the results of the BCI prediction.

Figure 3.8: Hardware specifications for the BCI

Chapter 4

Algorithm exploration tool

So far, we have described the BCI in terms of general and hardware specifications. The system characteristics (synchronous, active and non-invasive BCI based on several brain rhythms linked to event-related potentials) have been investigated in Chapter 1. The hardware (both electrodes-helmet system and the acquisition board using FPGA MAX1050) has been investigated in Chapter 3. However, we have investigated neither the signals processing characteristics (software specification) of the BCI nor its usage condition (How the user has to behave for using the BCI ? How long should he/she wait for the BCI to make a prediction of his/her movement's intent ?).

In this chapter, we will first investigate the usage conditions of the BCI in terms of signal acquisition (also known as the acquisition protocol). This will lead to training databases creation, which will be used in Chapter 5 and 6. Then, we will focus on the BCI software, which represent the most critical part of the BCI design. To do that, we will proceed in two steps :

1. An algorithm exploration tool, developed in `MATLAB`, is described. This application, realized before the implementation on the experimental system, aims at performing offline analysis of EEG signals for determining the processing chain that will offer the best results in terms of classification accuracy. The EEG signals can be loaded in the application with files from real acquisitions using the experimental system. A basic EEG simulation tool, which will not be described because it is left as a further improvement, can simulate EEG signals where a movement occurs. This algorithm exploration tool allows to choose several types of combinations for the processing chain and explore the different possibilities when it comes to BCI implementations.

2. Using the `MATLAB` tool, one will establish a broad comparison of different algorithms for determining which combination performs the best in term of classification rate. The best combination will define the software specifications of the BCI real-time system developed in this work. The software will then be implemented in real-time on the experimental device in chapter 5.

This chapter is organized as follows. First, timing specifications are defined in section 4.1 and the acquisition protocol for establishing the training data sets is described in section 4.2. In section 4.3, a description of a growing and interesting toolbox is presented. This opensource toolbox called `FieldTrip` provides a huge set of `MATLAB` functions to analyze and process EEG data. Then, from section 4.4 to 4.8, the `MATLAB` application is explained by describing each step forming the processing chain of the BCI, ie. the signal generation step, the, preprocessing, the feature extraction and finally the feature extraction. In section A broad comparison of different algorithms is then performed. Finally, in section 4.10,the specifications for the software aspects of the BCI are defined.

4.1 Timing Specifications of the BCI

In previous chapters, one has defined, as reminder, several characteristics of the BCI:

1. The BCI is active, non-invasive and acquire EEG signals.
2. Recorded signals are based on different electrical brain rhythms (mostly μ rhuthm) linked to event-related potentials.
3. The BCI is able to recognize 3 different motor tasks : Right-hand movement, left-hand movement and rest state.
4. The BCI works in real-time and is synchronous. It means that it works in 2 steps : it first records EEG signals of a given movement during a certain duration (4 [s] in this work) and then it processes them. Notice that the user must perform only one motor task during a given records.
5. Recording system is based on 5 electrodes (C3, Cz, C4, Cp1 and Cp2) positionned thanks to the helmet *Ultra Cortex Mark IV*.

These specifications do not precise, however, several points. The first point that must be precised concerns the acquisition period. One need to record EEG signals during a certain time. This period must be defined according to a trade-off : on the one hand, the longer the acquisition lasts, the more information one has for predicting the user's intent. On the other hand, the shorter the acquisition period, the closer the BCI gets to a an asynchronous system. The choice is made based on characteristic transient periods of brain ERPs signals, which are used in this work. As described in Chapter 2, the characteristic time of ERPs is between 0.5 [s] and several seconds depending on the duration of the motor task itself [58] [59]. It is anyway well-localized in time and therefore one may choose **an acquisition time of 4 seconds**. This corresponds to a movement duration of about 3 [s] which seems coherent.

Moreover, one must impose constraints about the processing time such that BCI system provides a prediction of the user's intent within a timing limit. One will choose arbitrarily 10 % of the recording time. Therefore, the whole processing chain must be executed in real-time in less than 0.4 [s]. One may now define the timing specifications of the BCI system. Fig.4.1 depicts such specifications. With these specifications, one may ensure that the BCI will provide a prediction of the user's intent in less that 4.4 [s] after the system initialization.

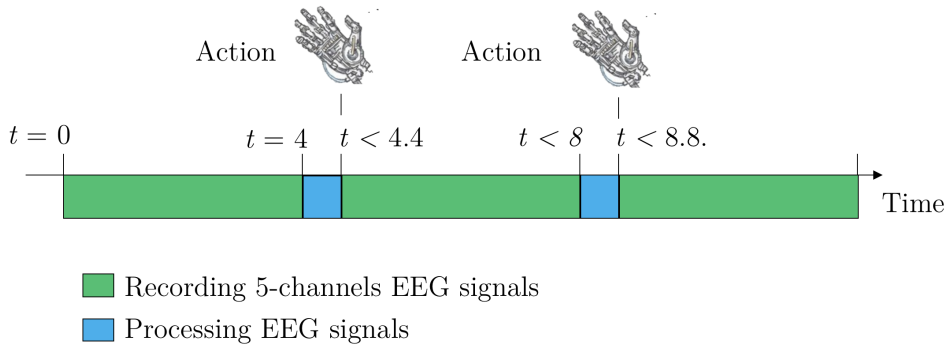


Figure 4.1: Recording and processing timing specifications of the BCI real-time system.

4.2 Recording protocol for training database generation

In order for the BCI to classify feature vectors in real-time, one must train it and evaluate its performances. Therefore, one must record a huge quantity of signals for creating a training data

Type	**Advantages (+)**	**Drawbacks (-)**	**Comments**
60 records in a row	Fast acquisition	Does not take accurately the helmet positions variations	20 left-hand - 20 right-hand - 20 rest (order is not important here)
30 records in a row	Intermediate acquisition time and good model of helmet positions variations	/	10 left-hand - 10 right-hand - 10 rest (order is not important here)
15 records in a row	Excellent model of helmet positions variations	Slow acquisition	5 left-hand - 5 right-hand - 5 rest (order is not important here)

Table 4.1: Advantages and drawbacks of the different acquisition methods.

set and a test data set. Creating such a database is challenging because recording conditions must be equivalent for both the database creation and the BCI real-time usage. Especially, there are five critical points :

1. One needs to record 4 [s] EEG signals and wait for 0.4 [s] before the next acquisition.
2. One needs to records the database in several trials. The helmet must be removed between the trials in order to take have a training database with different helmet positions. Variations due to the helmet position on the head is thus taken into account.
3. One needs to perform successive recording in a realistic sequence. Realistic means a sequence of movements that the user is likely to execute. For example, the user will barely never have performed a right-hand movement 30 times in a row. Therefore one must not record the training database by executing a right-hand movement 30 times in a row neither.
4. The way that the patient is told to start a given movement and which movement to start.
5. The way the movements must be executed during the training data set creation and the real-time usage have to be similar (movement amplitude, movement rhythm).

The first concern is solved by inserting a software delay of 0.4 [s] between successive recording operations. Regarding the second issue, a trade-off exists. If the helmet is removed and reset on the head afterwards, there will be variations in the helmet position. This drastically increases the time needed for establishing the database. One has thus discussed three possible solutions, as explained in Table 4.1.

Choosing an acquisition protocol based on a series of 30 successive 4-seconds acquisitions over 5 channels without removing the helmet seems to be the best compromise. It indeed allows us, for a database of 1500 trials (500 for each of the 3 tasks) to model 50 different helmet positions.

Regarding the third concern, a trade-off exists too. One has just defined an acquisition protocol based on a series of 30 successive 4-seconds acquisitions. However, the order of the sequence (i.e. the order of successive signals records) plays an important role in the classification rate (desrcibed in details in Chapter 6). Indeed, the brain undergoes habit effects [60] [61] which can drastically influence the performance of a BCI. For example, if a user raises ten times in a row his/her right arm and then raises the right arm once more, event-related potentials will be surrounded by slow cortical potentials and other electrical signals that will introduce a bias in the recorded signal. This bias can lead to a better classification rate than the real one. This habit effect disappears when the user alternates between each of the possible motor tasks. Figure 4.2 depicts this habit effect :

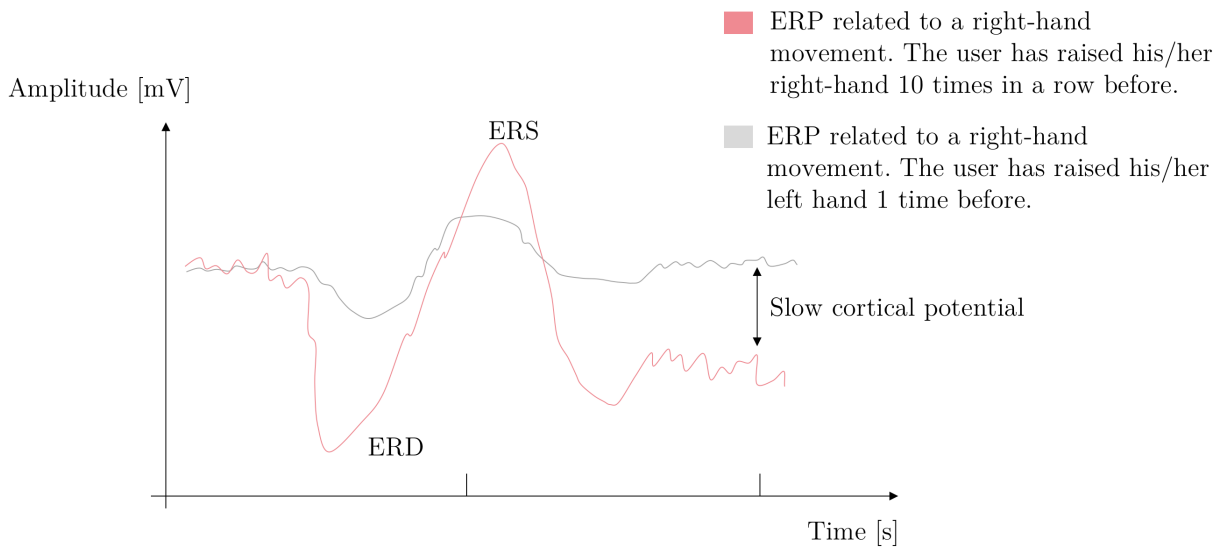


Figure 4.2: Event-related potentials for the two same tasks but with different initial conditions. **In red**, the given movement (right-hand here) has already been executed 10 times before the one related to this ERP (right-hand here). Habit effect is present. **In grey**, a different motor task (left-hand here) has been executed before this motor task (left-hand here). No habit effect is present.

By inspecting Fig.4.2, one may understand the issue with habit effects. If the database is recorded in 3 stages ie. first all right-hand movements, then all left-hand movements and eventually all rest states, this may lead to an excellent classification rate. Indeed, as depicted in Fig.4.2, event-related potentials linked to motor tasks that are repeated several times in a row (in red) are much easier to distinguish than event-related potentials linked to a sudden motor task that is not executed before (in grey).
One must therefore choose the most appropriate acquisition protocol, i.e. the most realistic one regarding the BCI usage and avoiding a protocol where habit effects can occur. One has defined 3 possible protocols.

The second protocol that records 30 successive 4-seconds signals in 15 series of 2 movements seems to be the most appropriate acquisition protocol.

Regarding the fourth issue, one may cite several protocols found in the literature [62] [63]. Indeed, one may use a sound clue [62] or just a visual clue to indicate which movement to start. Regarding the hardware available for this BCI design, one will choose the visual clue. This consists in printing in the computer console connected to the FPGA the movement that must be executed and the moment one need to start the movement. Fig.4.3 depicts the complete protocol used for creating the database. Acting like this allows to establish the following data base.

$$\{E_n\}_{n=1}^{N} \quad \text{with } E \in \mathbb{R}^{\text{ch x time}} \tag{4.1}$$

where N is the number of trials in the database, ch is the the number of channels used in the BCI design. In this work, the number of channel is 5 and time $= 4 \cdot F_s$ corresponds to the number of samples used for one record of one channel. This data set corresponds in fact to the training data set that will be used later in this chapter. It is made of 500 trials for each motor task (right-hand, left-and an rest state). The total data set contains thus 1500 trials ($N = 1500$).

Type	Advantages (+)	Drawbacks (-)	Comments
1 (well-defined order)	May lead to a better classification rate.	Not realistic at all. The user will never execute 10 times in a row the same action.	Sequence is defined as 3 series of 10 movements : 10 left-hands followed by 10 rest states and then 10 right-hands.
2 (realistic order)	This order is realistic.	/	Sequence is defined as 15 series of 2 movements, e.g. : 2 left-hands followed by 2 rest states, then 2 right-hands followed by 2 rest sates again,...
3 (random order)	Elimination of habit effects	Not necessarily realistic.	Sequence is defined as 30 series of 1 movement in a random order, e.g. : 1 left-hand followed by 2 rest state, then 1 right-hand followed by 2 left-hand again,...

Table 4.2: Three possible protocol for recording signals in order to generate the training database.

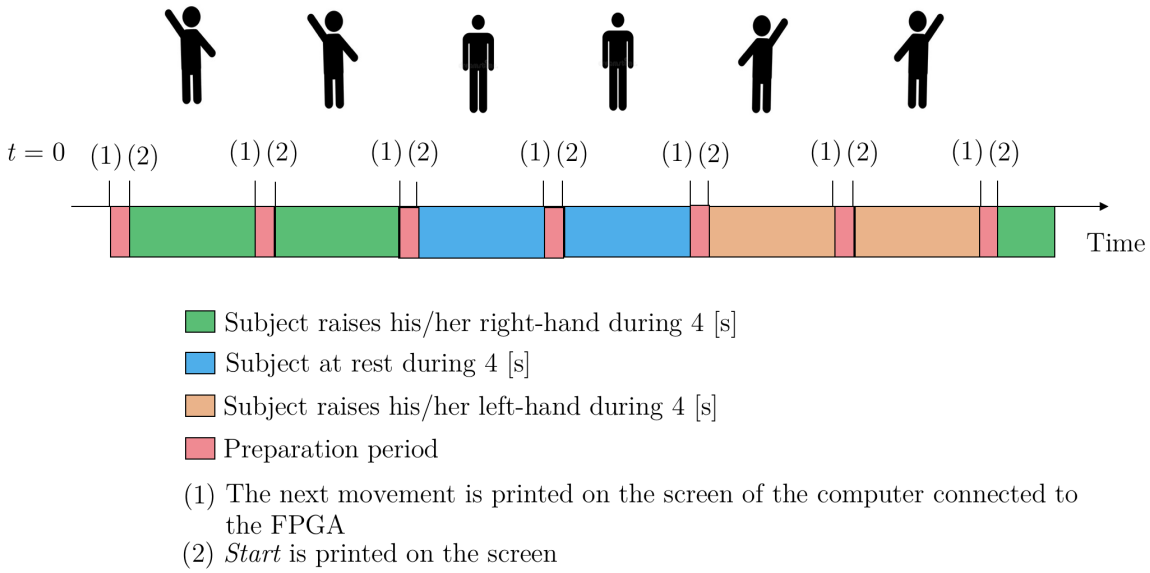


Figure 4.3: The complete protocol is explained in this figure : for a set of 30 acquisition, 15 series of two movements are asked from the patient who trained the system. The user is fed with movement that he/she has to execute 0.5 [s] before the movement recording operation starts.

Regarding the last concern, one has chosen the following protocol. Figure 4.4 depicts the protocol used for a right-hand movement.

1. Right-hand movement : the user, seated, rocks his/her right arm form a position with the arm along the body to a position where the arm is extended, perpendicular to the body, in the same plan as the body itself. The right-hand is extended too. This forms a cycle executed three times during the 4 [s] acquisition period.

2. Left-hand movement : the user, seated, rocks his/her left arm form a position with the arm along the body to a position where the arm is extended, perpendicular to the body, in the same plan as the body itself. The left-hand is extended too. This forms a cycle executed

three times during the 4 [s] acquisition period.

3. Rest state : the user, seated, keeps his/her arms and hands along the body and does not move during the 4 [s] acquisition period.

(1) Right arm along the body

(2) Right arm extended

(3) Right arm back to initial position

Figure 4.4: Protocol used for recording a right-hand movement. The left-hand movement is similar, but for the other arm while the rest state is equivalent to state (1) of this figure for the whole recording period.

To summarize the database is established using the type-2 protocol described earlier. With this protocol, the user places the helmet on his head and records 30 trials in a row. Each trial is characterized by a given motor task (right, left or rest), lasts 4 seconds and is recorded over 5 channels. The user must execute 2 times the same motor task and then execute an other 2-movements sequence. This defines 15 series of 2 movements. Afterwards, the user removes the helmet. After a few seconds, the user may place it again and restart a record of 30 trials. Notice that two additional training database acquired with protocol 1 and 3 are nevertheless recorded. These data sets will be used in chapter 6 in order to compare results variations related to training protocols.

4.3 Introduction to the FieldTrip toolbox

One will now describe the algorithm exploration tool used for defining the BCI real-time software. This tool is partially based on the `FieldTrip` toolbox. This is a great instrument for EEG analysis and this section aims at introducing this toolbox and opening doors for further use of this resourceful toolbox.

4.3.1 The toolbox

`FieldTrip` is the `MATLAB` software toolbox for EEG analysis that is being developed by a team of researchers at the Donders Institute for Brain, Cognition and Behaviour in Nijmegen, the Netherlands in close collaboration with collaborating institutes. The development of `FieldTrip` currently receives support from the ChildBrain project and from the Netherlands Organization for Scientific Research (NWO) and has previously been supported by the Human Connectome project and BrainGain [49].

The toolbox offers advanced analysis methods of EEG and invasive electrophysiological data, such as time-frequency analysis, source reconstruction using dipoles, distributed sources and beamformers and non-parametric statistical testing. `FieldTrip` contains high-level functions that you can use to construct your own analysis protocols in `MATLAB`. Furthermore, it easily allows developers to incorporate low-level algorithms for new EEG analysis methods.

All the information, tutorial videos, code examples and explanations can be found in [49] and [50]. The toolbox can be downloaded at [51].

4.3.2 The FieldTrip structure

When using the `FieldTrip` toolbox, it is important to be comfortable with the `FieldTrip` structure. The classical structure is illustrated in Fig.4.5.

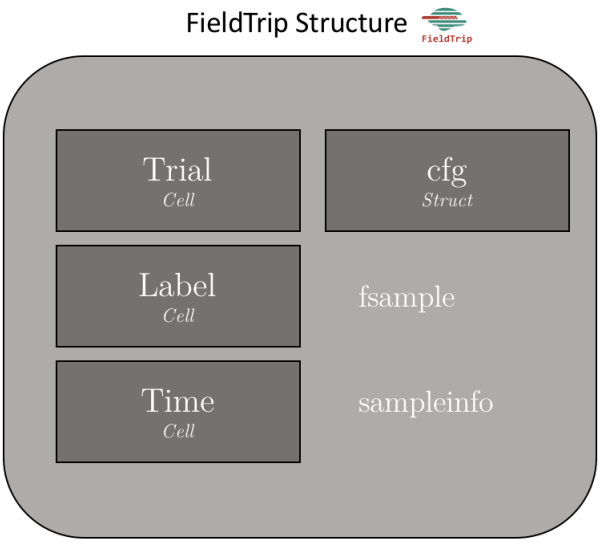


Figure 4.5: Organization of a FieldTrip structure containing EEG data.

The structure presents the following elements:

1. The EEG data contained in a single matrix *channels* × *samples*. The *Trial* cell can contain different trials, where each trial is similar to a matrix with a number of rows equal to the number of channels and a number of columns equal to the number of samples.
2. The cell *Label* that contains the different names of the channels, for example 'C3' or 'C4'.
3. The cell *Time* that contains the Time vector. This vector is extremely important since it defines the different time points at which the data is sampled.
4. The *cfg* structure that contains all sorts of important parameters for the `FieldTrip` functions that will be applied to the EEG data.
5. *fsample*, the sample frequency.
6. *sampleinfo* that contains the number of EEG data samples.

4.3.3 FieldTrip functions

The `FieldTrip` toolbox provides lots of functions useful for EEG analysis, going from the preprocessing to the artifact rejection, ICA decomposition and more. The website [49] can be visited in order to discover all the different possibilities and documentation. Once the toolbox is downloaded, it is possible to see that the different functions are extremely well documented. Only some of the most important functions are detailed in this section.

4.3.3.1 Preprocessing

The function *ft_preprocessing* is one of the first and most important functions present in the `FieldTrip` toolbox. This function is extensively used in our MATLAB application and is therefore explained in detail. It automatically takes care of the first steps in processing the data. This means that *ft_preprocessing* can be used for:

1. Loading the data in the `FieldTrip` structure.
2. Cutting up the data into trials if needed.
3. Baseline correction.
4. Detrending the data.
5. Filtering the data (Highpass filter, lowpass filter, bandpass filter, Notch filter).

Before using the function, all the parameters have to be inserted in the *cfg* structure of the `FieldTrip` data. The different parameters, among others, can be the type of filter, its cutting frequency, the filter order, the type of window used,...

The output of this function is the `FieldTrip` structure containing the preprocessed data and lots of functions can then be applied on it.

4.3.3.2 Frequency analysis

The function *ft_freqanalysis* computes the frequency or time-frequency representation of temporal data. The input data should be organized as is a classical `FieldTrip` structure and obtained from the *ft_preprocessing* function. This function can be specified with parameters such as :

1. The method and type of Fourier transform for calculating the spectra.
2. The type of time frequency transformation such as the Wavelets transform, for example.
3. The desired outputs of the functions.

4.3.3.3 Artifact rejection

In EEG recordings, some artifacts such as EMG or EOG signals can be present and falsify the signals. It is therefore important to first detect these artifacts and then suppress them. `FieldTrip` also helps for this purpose by providing the functions *ft_artifact_xxx* and *ft_rejectartifact*, where *xxx* is the type of artifact. The first function detects the specified artifact and constructs a window around the artifact position. The second function removes the data segments containing artifacts. It returns a structure with a modified *trial* which can later be used for preprocessing only the clean data. Some options can also be specified for these functions, such as :

1. The type of artifact to detect.
2. The minimum acceptable length for a time window around an artifact.
3. A parameter to specify if the artifact has to be completely rejected or partially.

4.3.3.4 Other functions

A lot of other interesting functions are also present for different interesting analysis such as :

1. EEG source reconstruction.
2. Statistical analysis and multiple comparison for EEG/MEG data.
3. Data visualization.
4. ...

4.3.3.5 A typical pipeline for data analysis

An example coming from the `FieldTrip` documentation is presented in Fig.4.6. Data is taken from several subjects and then passed through the preprocessing function mainly for filtering. Then, artifact rejection is operated to remove the undesired artifacts. A frequency analysis is then performed. Finally, a source analysis of the data is done and some statistics can be operated on the results.

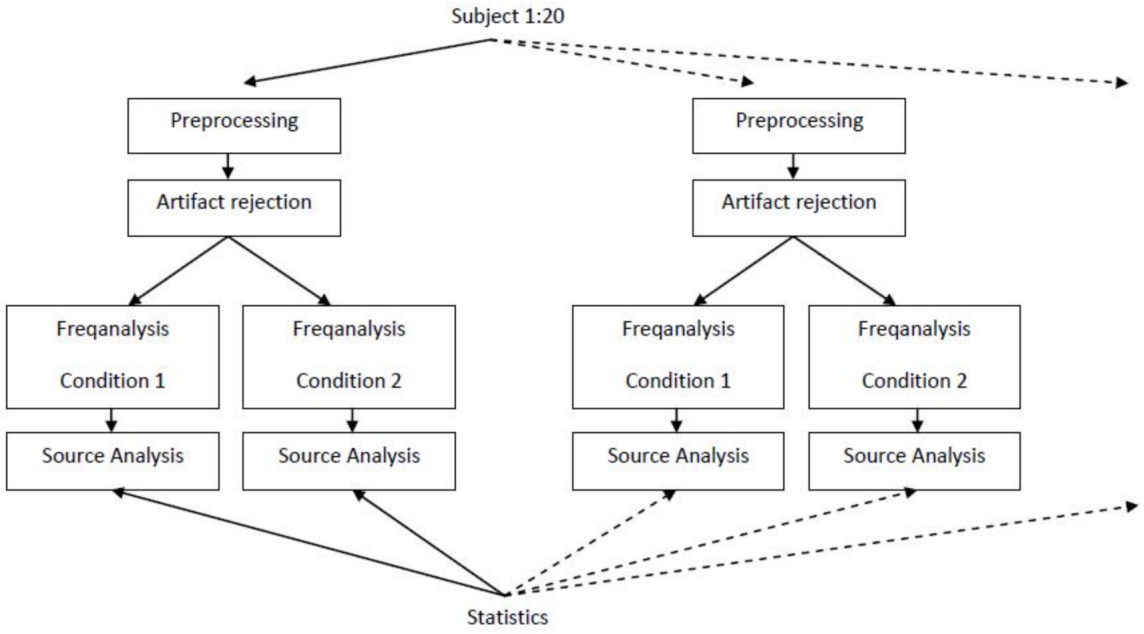


Figure 4.6: Typical pipeline using the FieldTrip functions.

4.3.4 Conclusion

The `FieldTrip` offers a lots of possibilities for EEG data processing. Some of these possibilities were explained in this section but the best way to fully see its capacities is to visit the well documented website and explore all the functions and script examples. This toolbox, continually improved by a growing community of BCI adepts, is clearly something that should be deepened and used.

4.4 Application : General Overview

The `MATLAB` application is organized in panels where each panel represents a step in the BCI processing chain, as described in Fig.4.7. Every rounded rectangle corresponds to a panel in the application, where several parameters can be changed so that we can choose the different combinations to perform.

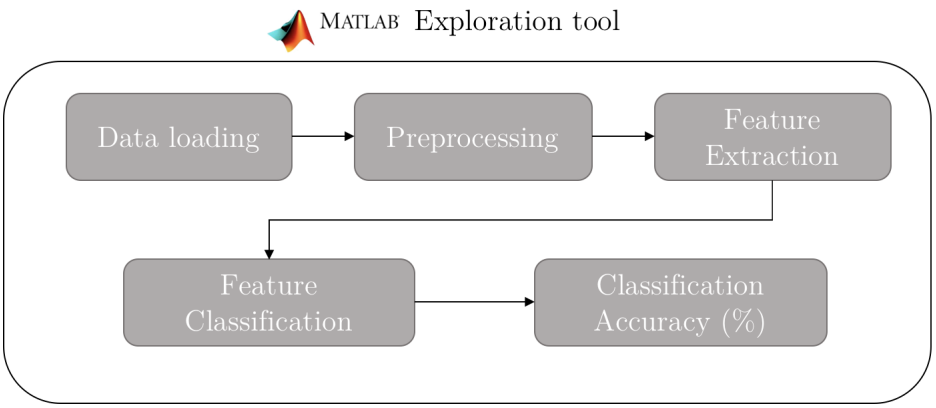


Figure 4.7: General overview of the exploration tool.

The next sections will successively describe each panel application, and thus each step of the processing chain.

4.5 Application : EEG signals acquisition

We will first focus on EEG signals acquisition. For this purpose, EEG signals are loaded after acquisition on an experimental device. Notice that the authors have thought about an extra possibility for obtaining EEG signals based on signals simulation. This possibility would offer several advantages :

1. This would offer the possibility to start an offline implementation of a BCI without having an experimental device yet.

2. This would offer the possibility to obtain quickly a very large training database. Such a database would help to investigate how different classification algorithms behave when the training database becomes larger and larger.

Therefore one has investigated several simulated EEG signals algorithms [64] [65] [66]. One may cite several techniques : Markov process amplitude (First-Order MPA, nonlinear MPA) or Artificila neural networks or autoregressive models. Nevertheless all these techniques remain complex and require a very important assessment process in order to prove the validity of simulated EEG signals. Moreover such a realization falls outside the scope of this work.

The authors have thus decided to insert the possibility in the `MATLAB` tool, which could be implemented in the future as explained in the conclusion and further improvements Chapter 6. In Fig.4.8, one may observe the first panel of the tool. Bear in mind that the EEG simulation tool is not implemented here and therefore clicking on the 'Simulate EEG data' will not work.

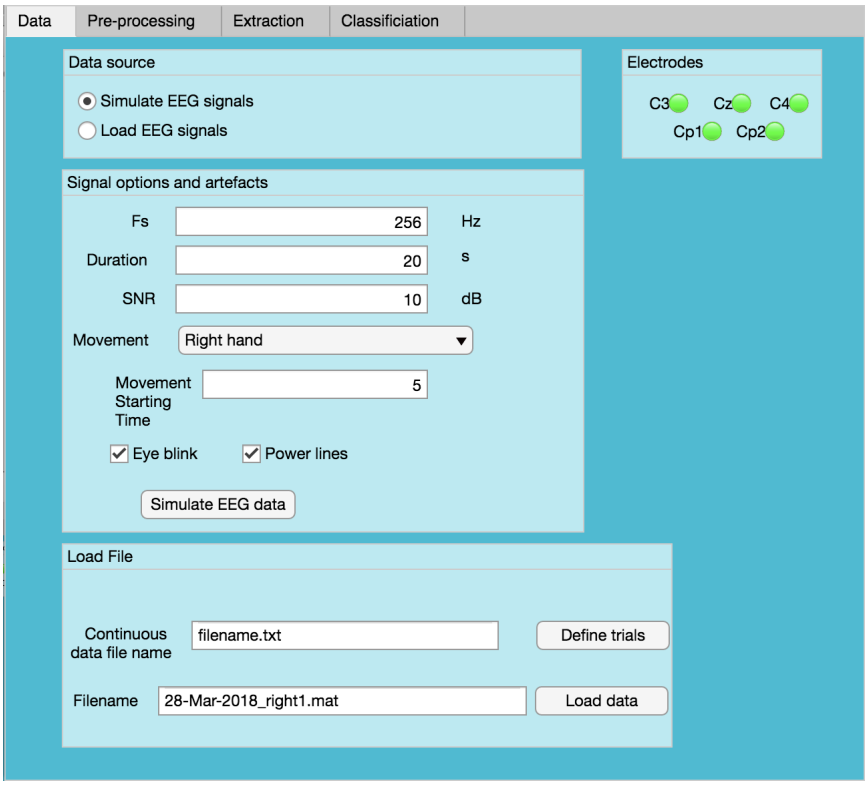


Figure 4.8: Application tab to simulate or load the EEG data

On the other hand, the second method consists in loading EEG signals from existing files. When the acquisition is performed with the experimental board, the signals are sampled and saved in hexadecimal for each electrode. A `MATLAB` script converts the file written in hexadecimal into trials store in `.MAT` files. Within these files, one may notice 6 vectors : one vector for each electrode (signals units are [mV]) and a time vector, as illustrated in Fig.4.9.

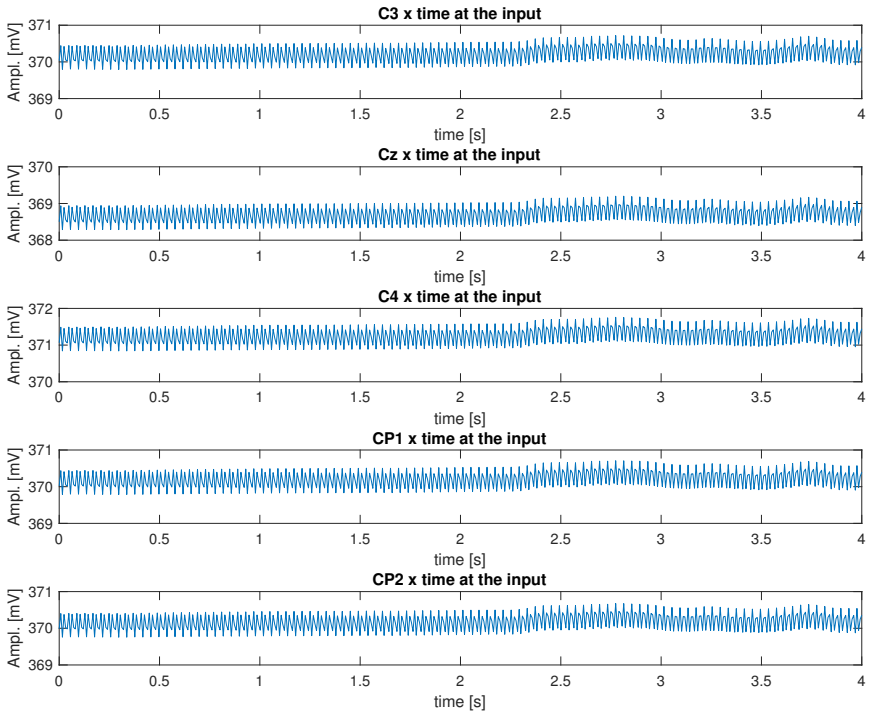


Figure 4.9: Loaded EEG signals where a movement occurs after 2 seconds.

4.6 Application : Frequency and spatial filtering

Once the data is loaded in the `MATLAB` tool, the prepocessing step can take place. This step includes frequency and spatial filtering. The preprocessing panel of the application is presented in Fig.4.10.

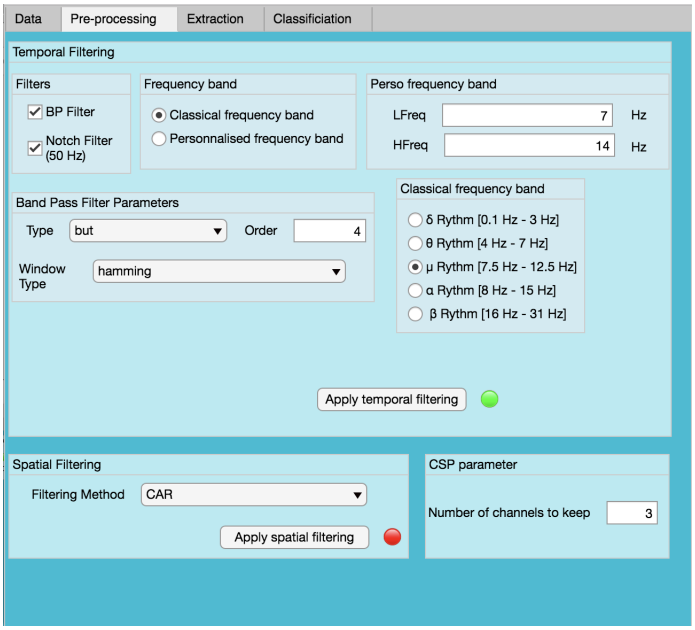


Figure 4.10: Preprocessing panel of the application : preprocessing consists in frequency and spatial filtering options.

Several options allows the user to design and test its own frequency and spatial filters :

1. The filter type to apply: bandpass filter and/or Notch filter.
2. The frequency band to filter. It can be either a common frequency band corresponding to a specific rhythm (for example, μ rhythm) or a frequency band specified by the low pass and high pass cutt-off frequencies.
3. Bandpass filter parameters.
4. The type of spatial filtering that can be chosen among common average reference (CAR), common spectral pattern (CSP) or Laplacian (LAP).
5. The number of channels to keep if a channel reduction is preformed with spatial filters.

The data when it is filtered in the μ frequency band is shown at Fig.4.11. It is possible to see that the high frequency content has been eliminated from the signals.

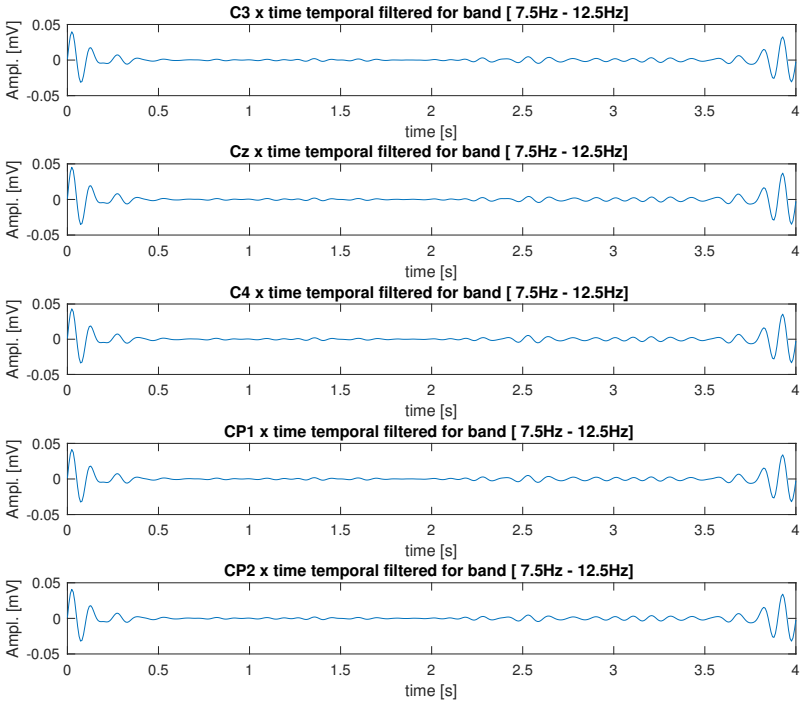


Figure 4.11: Loaded real data with a bandpass filter isolating the μ band.

After frequency filtering, the EEG signals are passed to a spatial filter. Here, the spatial filter chosen is the CAR spatial filter and the EEG signals are shown in Fig.4.12. After spatial filtering, it is easier to detect that something is happening after 2 seconds for the loaded data: we see that the signal becomes stronger.

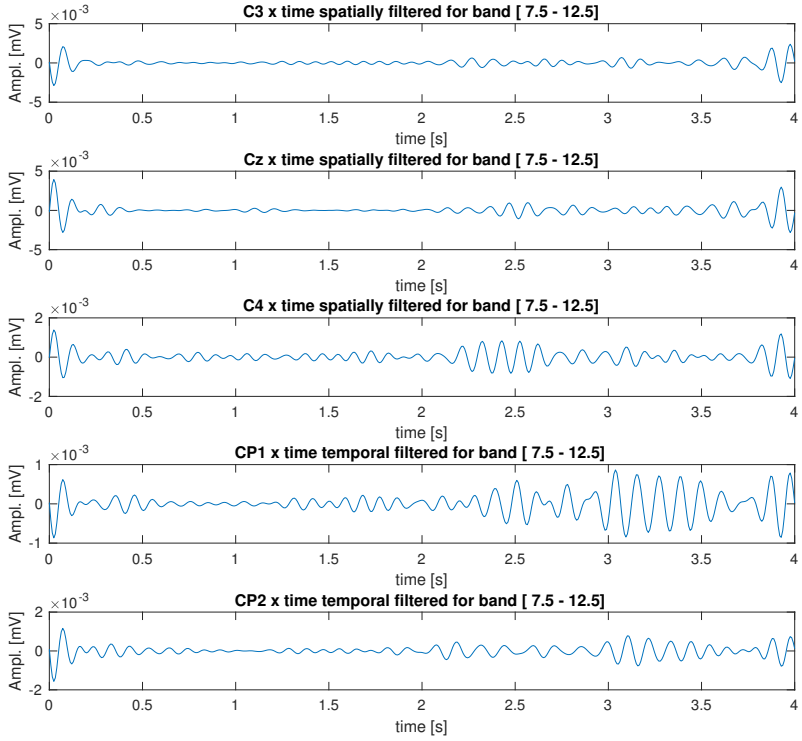


Figure 4.12: Loaded real data with a CAR spatial filtering.

4.7 Application : Feature extraction

The feature extraction step comes just after the preprocessing step. Just as a recall, it consists in computing a feature vector based on a particular algorithm. The output of this step is thus a feature vector that is then passed to the classification step. The feature extraction tab is shown at Fig.4.13.

For the feature extraction step, a few things can be chosen

- The windowing and framing parameters. These two parameters are illustrated in Fig.4.14. The signal is contained in several windows. In these windows, frames are defined where the feature extraction method is applied for each frame. When choosing appropriate parameters, this method is particularly useful to isolate segments of the signal where a movement occurs, for example.
- The feature extraction method among the ones explained in the previous chapter.
- Some parameters for the Wavelets method and the AR method.

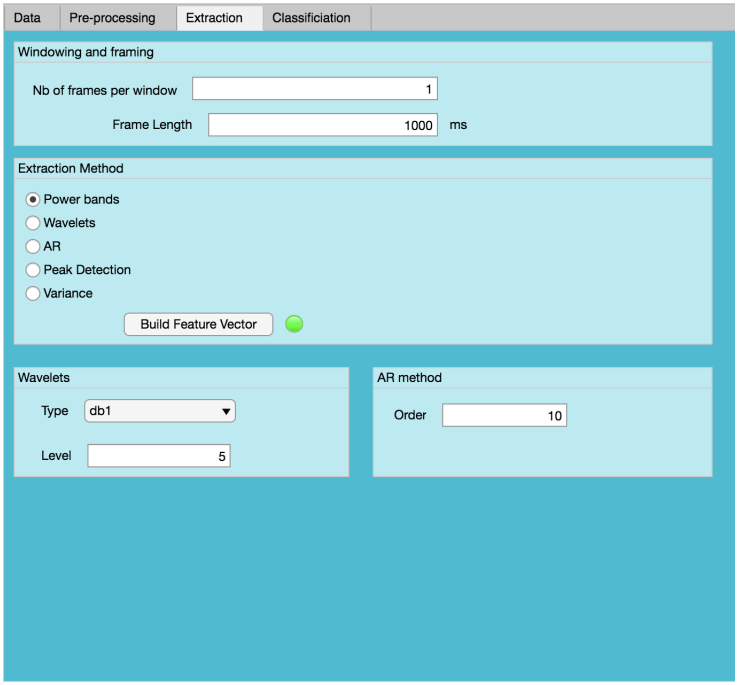


Figure 4.13: Application tab with feature extraction options.

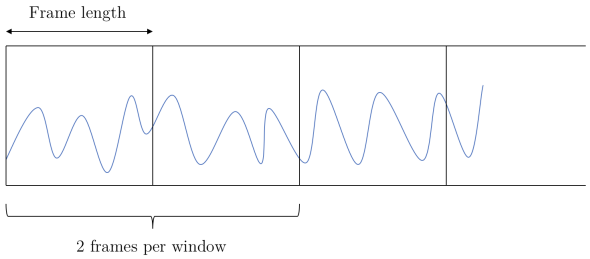


Figure 4.14: Windowing and framing concept.

4.8 Application : Feature classification

The feature classification tab is the last step: it assigns a class to the data and in our case, tells if a left hand, right hand or no movement at all is detected.

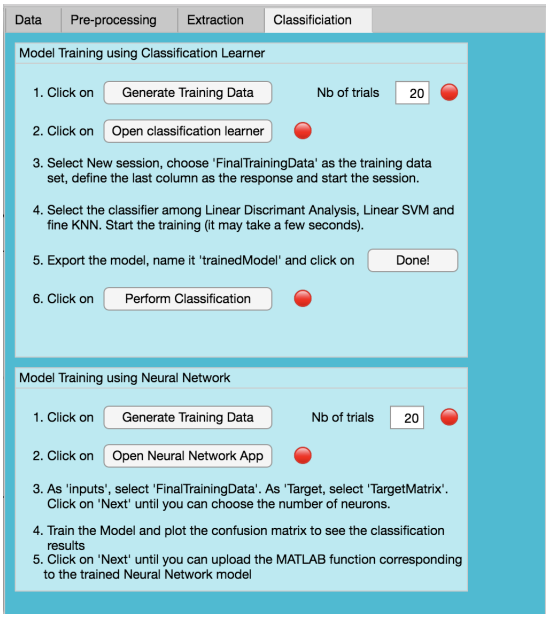


Figure 4.15: Application tab with classification options.

The classification can be separated in three steps:

1. First, the model needs training data to be trained. It is thus necessary to pass all signals through the entire processing chain made of the choices made earlier in the tab. All the parameters chosen before need to be saved in order to be able to generate the training data.

2. Once the training data is generated, we can now choose the type of classifier and train it.

3. After the model training, we can now export the model and use it to classify new data.

The classification tab is shown in Fig.4.15. It allows to choose between a large choice of classifiers using the already existing `MATLAB` application *Classification Learner*. It is possible to choose classifiers such as LDA, kNN, SVM, and many others. It also offers the possibility to train a Neural Network classifier with the `MATLAB` application *Neural pattern recognition app*. A small step by step tutorial is also explained so that the application is user friendly.

4.9 Results and methods comparison

After the description of the algorithm exploration tool, we can use it in order to determine the best combination for the processing chain and to find best offline results. Based on these results, one will define the software specifications of the BCI real-time system that will be implemented in Chapter 5. This section is organized as follows : one will first describe the conditions used for establishing the comparison, then one will describe the different comparisons that have been performed and eventually, the results are presented and commented.

4.9.1 Conditions and requirements

To explore the different algorithms, the authors use the training database constructed beforehand in this chapter. As reminder this database is made of 1500 trials with 500 for each motor task that the BCI real-time system must recognize. Several conditions and requirements are defined in order to execute the comparison :

1. The training database is $\{E_n\}_{n=1}^{N}$ with $E \in \mathbb{R}^{\text{ch x time}}$ established earlier in this chapter using a type-2 protocol.

2. The BCI uses event-related potentials linked to the μ-SMR brain rhythm. One will therefore impose a frequency filter for the classical μ band (8-12 [Hz]).

3. Each 4 [s] is used completely as defined in the training database. No windowing and framing operation is performed.

4. The accuracy of the classifier is used as assessment criterion. This corresponds to the number of correct predictions made by the BCI offline system over all predictions made.

5. Each combination is executed three times in order to take variations of the classification stage into account (classifiers are determined using an optimization problem that uses certain initial conditions. These initial conditions vary, leading sometimes to slightly different results for a same classifier).

6. A 2-order Notch filter removing 50 [Hz] perturbations with a quality factor $Q = 20$ is implemented in all cases and is thus not a part of the comparison.

4.9.2 Comparison tables

In Table 4.3, Table 4.4 and Table 4.5, the classification accuracy for the training set is displayed in function of

- The feature classification algorithm and the parameters associated to the algorithms
- The type of the frequency filter (FIR or IIR)
- The type of spatial filter, one filter per table (CAR for Table 4.3, LAP for Table 4.4 and CSP for Table 4.5).
- The feature extraction algorithm and the parameters associated to the algorithms(order of AR, wavelet type)

Note that some parameters have been fixed but will be varied later, such as

- The Wavelet level that is fixed to 5
- The order of the FIR frequency filter, set to 96
- The order of the IIR frequency filter, set to 3

Classification accuracy [%]										
	LDA		**SVM**		**kNN**		**MLP**		**Tree**	
	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR
AR										
6	54.2	52.5	55	53.6	58.3	55.9	57.6	54.8	55	53.3
7	55.1	53.7	57	58.6	56.9	55.2	58.3	59.9	55.2	54.7
8	62.3	59.7	63.9	56.5	66.5	62.9	68.6	66.1	63.6	62.3
9	61.2	57.4	60.8	59.1	64.7	61.4	67.2	64.9	64.2	62.1
10	60.1	56.3	61.3	57.8	62.9	61.2	65.8	64.3	62.4	59.6
11	59.7	57.5	56.1	56.4	56.3	60.9	63.2	61.2	61.1	60.9
12	58.3	55.1	56.4	54.8	61.4	63.9	64.1	60.7	62.6	58.8
Peak detection	64.1	63.4	62.3	60.7	70.7	71	72.5	68.9	70	68.9
Power Bands	74.2	73.1	70	69.9	70.4	71	74.7	71.8	73.5	72.7
Wavelets										
db1	68.8	67.2	70.1	68.5	71.3	76.1	81	76.5	72.1	71.8
db2	71.8	69.7	69.5	72.3	74	75.9	73.1	72.3	70	73.7
db3	64.7	62.1	73.9	70.7	70.1	71.2	70.1	72.2	69	68.5
db4	63	63.3	68.7	72.1	69.4	73.4	63.5	74.1	66.8	69.9
coif1	59.9	58.8	61.6	61.2	64	63.4	66.6	65.5	62.9	61.3
coif2	52.1	53.4	62	60.6	63.4	59.4	65.9	61.8	63.1	62
coif3	56.7	55.8	62.4	63.5	65.4	64	67.5	66.4	62.9	63.1
coif4	58.8	56.2	61.2	59.8	68.4	65.5	70.8	63.9	64	58.3
sym1	60	59.9	62.3	61.3	64.1	62.4	68.3	63	61.1	58.7
sym2	53.6	58.1	58.1	57.7	62.3	59.1	64.2	60.2	60.1	59.9
sym3	58	57.5	60.3	60.8	59.9	61	63.5	62.7	59.1	56.3
sym4	58.8	56.8	56.2	55.4	63.4	60.6	64.1	63.9	62.3	60.8

Table 4.3: CAR

Classification accuracy [%]										
	LDA		**SVM**		**kNN**		**MLP**		**Tree**	
	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR
AR										
6	57.3	55.2	58.9	56.2	59.2	58.7	61.8	60.1	55.1	53
7	58.8	57.6	60.1	59.9	62.5	61.5	65	62.4	62	58.9
8	67	64.8	64.5	63	62.5	60.9	69.5	65.8	62.4	60.6
9	65	62.8	66.8	65	61	59.9	67.5	64.7	62	61.5
10	61.5	60	62.9	61.5	59.5	55	53	55.4	60	58.7
11	55.2	52	58	56.2	53.4	52	62.5	60	59.9	56
12	56.5	53	55.1	52.3	56.9	57.2	61.3	59	56.5	56
Peak detection	65.2	63.5	66.9	65.8	69.9	67.5	75.8	75.2	68.4	66.2
Power Bands	67.1	64.8	69.2	66.4	70.1	68.9	77.2	79.5	70	68.8
Wavelets										
db1	79.3	76.4	85	81.2	73.7	72.5	85.7	83.9	78.7	77.1
db2	65.2	60	72.3	66.4	68.9	67.3	77.7	74.1	66	64.8
db3	62.1	58.9	68.7	65.6	64.5	62.9	76.8	74	63	58.7
db4	60.8	57.3	67	63.9	68.8	66.3	73.4	70.5	62	60.6
coif1	53	51.3	62	59.4	61.2	58.1	79.7	77.4	60.7	56.3
coif2	55	53.2	58.7	55.6	62	59.8	76	74.2	63	61.3
coif3	54.5	53.8	55.4	52.9	56.3	55.2	68.3	65.6	62.6	59.5
coif4	55.3	52.7	56.8	53.4	58.6	55.8	65.6	62.1	64.3	60.9
sym1	78	76.5	83.3	80.9	77	75.3	83.4	82.8	72.3	71.8
sym2	54	54.3	59.2	58.9	56.1	54.3	74	70.8	67.3	66.9
sym3	55.7	53.9	61.2	60.9	57.3	55.8	68.7	68.3	61.2	58.9
sym4	53.7	52.9	58.3	58.8	56.4	55.1	63.9	61.8	60.9	59.9

Table 4.4: LAP

Classification accuracy [%]										
	LDA		**SVM**		**kNN**		**MLP**		**Tree**	
	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR	FIR	IIR
AR										
6	58.4	57.3	62.7	60.2	55.6	58.9	63.3	62	59.8	56
7	63.3	61.8	65.5	63.1	66.7	64.9	68.8	67.1	59.3	58
8	69.1	66.8	71.2	72.9	73.4	71.8	75.6	73	69	67.4
9	68.3	67.5	70.1	71.2	69.5	68.4	74.9	73.5	69.8	67.5
10	65.7	64.6	67.5	66.5	62.9	61	70	68.5		
11	62.3	61.6	63.4	62	64.5	61.2	68.7	66.9	59	56.4
12	57.3	56	61.5	60	63.4	62.8	65	63.8	59.7	58
Peak detection	58.9	58	60.1	59.5	61.8	60.9	65.1	63	60.1	59.6
Power Bands	60.5	60.1	61.9	62.4	64.6	63.8	66.2	65.5	62.1	59.2
Wavelets										
db1	89	85.7	88.3	86	82.7	81	96.7	96.2	87	85.4
db2	78	75.8	86	81.5	77.7	78.1	90.4	86.6	82.9	80.1
db3	75	72.7	81.9	80	75.6	75	84.9	83.2	81.5	78
db4	72	70.6	73.5	71.4	75	72.1	80.2	78.1	72	71
coif1	65	61.2	66.5	64.3	67.6		82.2	75.1	61.2	58.8
coif2	63	61.5	60.7	59.7	61	62.1	77.7	75.4	54.7	5 2.1
coif3	62.1	60. 7	58.9	62.1	60.8	57.6	72.5	69.9	69	65.8
coif4	60	58.7	61.4	60.2	63.5	62.1	70.5	69.7	66	64.9
sym1	87	85.8	94	92.1	71	68.1	95	94.6	88	85
sym2	78	75.6	85.1	83.2	63.6	60.1	91.7	88.8	80.7	
sym3	75.5	72.9	78	77.4	79.5	78	85	82.5	80	78.2
sym4	76	72.4	73.5	72	76	74.5	86.3	84.2	83.4	80

Table 4.5: CSP

In Table 4.6, the FIR filter order is investigated and the optimal number of Neurons in the MLP Neural Network as well.

Classification accuracy [%]					
	FIR Filter order				
	10	15	20	25	30
Neurons					
10	86	88	90.1	84.5	83
20	93.5	94.6	92	87	86.5
30	92.9	98	95.3	90	85.1
40	93.5	98.4	96	92.4	86.5

Table 4.6: MLP Neural network and filter order

In Table 4.7,

- Mix1 is a combination of the mean, variance,maximum and minimum
- Mix2 is a combination of the Willison amplitude,variance,maximum and minimum
- Mix3 is a combination of the variance,mean , MFL and Average amplitude change

- Mix2 is a combination of the Average amplitude change,variance,maximum,minimum

Classification accuracy [%]				
	Information			
	Mix1	Mix2	Mix3	Mix4
Wavelet Level				
1	74.5	77	72.6	73
2	75	78.5	74	72.9
3	77.5	81	82	80.5
4	88.5	89.9	82	86.5
5	92.5	98.3	93.9	91.3

Table 4.7: Wavelet Level and parameter to extract

In Tables 4.3, 4.4, 4.5, 4.6 and 4.7, the classification accuracy according to the type of feature extraction algorithm and classification algorithm are shown. The best accuracy per classification algorithm and parameter is highlighted. Based on these tables, we can draw several important observations.

- FIR filters provide, in general, better results than IIR filters with the type and order of filter chosen. Indeed, almost all of the highlighted results, which are the best results for the classification algorithm, are the ones obtaines with a FIR filter. The order of the filter that is the most suitable will be explored later in Table4.6.

- The classification results tend to vary with the type of spatial filter used. Indeed, as we can see in Table 4.3, the spatial filter CAR does not help to have a strong classification accuracy. However, in Table 4.4 with LAP and Table 4.5 with CSP, the accuracy tends to be more accurate by between 5 and 15%. Table 4.5 offers the best results. This can be explained by the fact that CSP is a more advanced spatial filter method. This is a method that needs training and when trained, the filter offers better performances than linear filters such as CAR or LAP.

- With AR feature extraction method, the results vary a lot with the order of the model. We have gathered results from order 6 to 12, and the best results seem to always be obtained with model order 8 or 9. These results are coherent with [4].

- The Wavelet feature extraction method provides very good accuracy in general, eventhough it strongly depends on the wavelet type. The type of wavelet that stands out compared to others is the *db1* type, just followed by the *sym1* type. These two types almost produce the best results for every classification algorithm and spatial filter. *coif* wavelets seem to offer poor results compared to other type of wavelets.

- The Peak detection and Power Bands methods offer good classificationr results. These two methods can be seriously considered for the real time application since they are both easy to implement.

- By observing Table 4.6, we can see that both the FIR filter order and the number of Neurons for the MLP Neural Network have an influence on the classification accuracy. Indeed, we can see that the optimal filter order stands between 10 and 20. This can be explained by the fact that under this range, the order is not sufficient to have sharp transitions and remove the unwanted frequencies with a good precision. Above this range, the filter order seems to be too high since the classification accuracy decreases. The number of Neurons plays a

great role in the classifier accuracy. The more there are Neurons, the better the classifier will be. However, the number of Neurons have to be carefully chosen since complexity of the algorithm and the computation time directly come from this parameter.

- As seen in Table 4.7, the level of the wavelet decomposition has a big impact on the classification results. It has been found that the level 5 was a good compromise and offers very good results. The type of information to extract has an influence on the results as well. Indeed, we can see in Table 4.7 that the accuracy varies in function of the parameter to extract. The different mix of characteristics offer different classification results, with the best results obtained with the second mix that includes the Willison amplitude,variance,maximum and minimum.

4.10 Conclusion and BCI software specifications

This chapter introduced some important features of our work during this year: the algorithm exploration tool. This tool, entirely coded in `MATLAB`, helped us to determine the best combinations of algorithms for preprocessing, feature extraction and feature classification. Even if all algorithms are not explored with this tool, a big number of combinations can be chosen and investigated.

This chapter first explained the timing specifications and the recording protocol for the generation of the database, which will be extensively used for the training of the algorithms models. Then, a short introduction to the FieldTrip toolbox was presented. This toolbox was used during the realization of our tool and opens door to further improvements with its huge set of functions for EEG signals processing. The tool was then presented by going through the EEG signals loading or simulation options, the preprocessing options and the feature extraction and classification options. Finally, the classification accuracy of various algorithms combinations were computed to finally highlight the most promising ones. One may now establish, thanks to these results, the software specifications of BCI the system as shown in Fig.4.16. These specifications are chosen with the results obtained from the `MATLAB` application and exploration of the different algorithms and methods.

1. The BCI system will use a bandpass FIR filter over the μ-band of order between 10 and 20. After complementary assessments, one has chosen a 16th-order filter. On top of that an order 2 notch filter with quality factor $Q = 20$ is chosen.
2. The BCI system will use the CSP algorithm as a spatial filter, which provides good results.
3. The BCI system will use the MLP Neural Network algorithm as a classifier. This type of classifier is clearly better than other classifiers when observing the results from Tables 4.3,4.4 and 4.5. The number of Neurons is chosen following Table 4.6. Choosing 30 neurons seems to be a good choice since it provides very good results. Even if in theory, increasing the number of neurons will enhance the accuracy of the classifier, it also increases the computation time and a compromise must therefore be found.
4. The BCI system will use the Wavelet decomposition algorithm with 'db1' window as a feature extraction method. The level of the Wavelet decomposition is set to 5 and the information to extract is chosen according to Table 4.7, which is the second mix of characteristics that include the Willison amplitude, variance, maximum and minimum for each detailed level of decomposition.

Figure 4.16: Software specifications for the BCI real-time system.

Chapter 5

Real-time application

In order to design a BCI real-time system, one must first establish its specifications. This was progressively done in the previous chapters. Once the specifications are established and the hardware system is available, one may proceed to the software implementation.

This chapter focuses on the real-time application and implementation on the FPGA MAX1050. This implementation is realized in two parts : a `Verilog` part and then a `C/C++` part programmed in a softcore (programmed itself in `Verilog`). By combining these two parts, the complete processing chain described in Chapter 4 is implemented and the BCI real-time system is functional. Performances in terms of classification accuracy are assessed in Chapter 6.

This chapter is organized as follows : First, the complete specifications of the BCI real-time system are presented in section 5.1. Then, a general overview of the software is presented in section 5.2. Next, the details about `Verilog` implementation are presented in section 5.3 and the details about `C/C++` implementation are presented in section 5.4. Eventually, the authors will also present some performances in terms of memory and timing management that are important to stress in section 5.5.

5.1 Specifications

Let us recall here all the specifications established at the end of Chapters 1, 3 and 4. These specifications are depicted in Fig.5.1 and Fig.5.2.

1. The BCI system will use the BCI board designed in [3].
2. The BCI system will use the 3D-printed *Ultra Cortex Mark IV* as helmet.
3. The BCI system will acquire EEG signals over 5 electrodes : C3, Cz, C4, Cp1 and Cp2. This choice is justified as follows : recording more signals could lead to a better classification rate. However using more than 5 electrodes is not possible with the FPGA MAX1050 considering the particular processing chain that must be programmed. Indeed the number of logic elements and the memory are limited.
4. Given that this work aims at developing a BCI only, one will not connect the BCI to an end-effector to apply a command. One will instead use the host computer connected to the FPGA via JTAG to print the results of the BCI prediction.

Figure 5.1: Complete specifications (general, hardware and software).

5. The BCI system will use a bandpass FIR filter over the μ-band of order between 10 and 20. After complementary assessments, one has chosen a 16th-order filter. On top of that, a 2nd-order Notch filter with quality factor $Q = 20$, is used.

6. The BCI system will use the CSP algorithm as a spatial filter, which provides good results.

7. The BCI system will use the MLP Neural Network algorithm with 30 neurons as a classifier. This seems to be a good choice since it provides very good results.

8. The BCI system will use the Wavelet decomposition algorithm with 'db1' window as a features extraction method. The level of the Wavelet decomposition is set to 5 and the information to extract is a mix of the Willison amplitude, variance, maximum and minimum for each detailed level of decomposition.

Figure 5.2: Complete specifications (general, hardware and software).

5.2 General overview

The processing chain implemented in real-time is based on the six classical successive stages :

1. The acquisition system using analog-to-digital converters for measuring raw brain signals from five electrodes and using a sampling rate of 250 [Hz].

2. The filter stage made of a notch filter that rejects the 50 [Hz] artifact and then a bandpass filter that isolates the μ-rhythm from the raw signal.

3. This stage is made of two steps. First a 2-fold downsampling operation for obtaining $F_s = 125$ [Hz] is made. Then, a spatial filter performs a signals combination in order to emphasize independent sources and remove signals artifacts.

4. The features extraction stage, based on discrete wavelet transform (DWT), that keeps only highly informative data of each channel over 5 levels (variance/minimum/maximum/Willison amplitude) and that maps them onto a vector of smaller size. This dimension reduction allows a faster processing for downstream operations. Notice that the features vector is then log-normalized.

5. The classification stage that maps the features vector onto a 3-dimension decision vector that contains in each entry the probability of each motor task. This is done using feed-forward multilayer perceptron neural networks.

6. The prediction stage when the predicted user's intent is printed on the screen of the host computer connected to the FPGA via JTAG as mentioned in the BCI hardware specifications in Chapter 3.

Figure 5.3 depicts a general overview of the complete processing chain. Within this processing chain, some parts are implemented in `Verilog` programming. These parts correspond to stages 1 and 2 (acquisition, bandpass and Notch filters) while other parts are implemented in `C/C++` programming using a softcore programmed in `Verilog` on the FPGA. These parts correspond to stages 3 to 6.

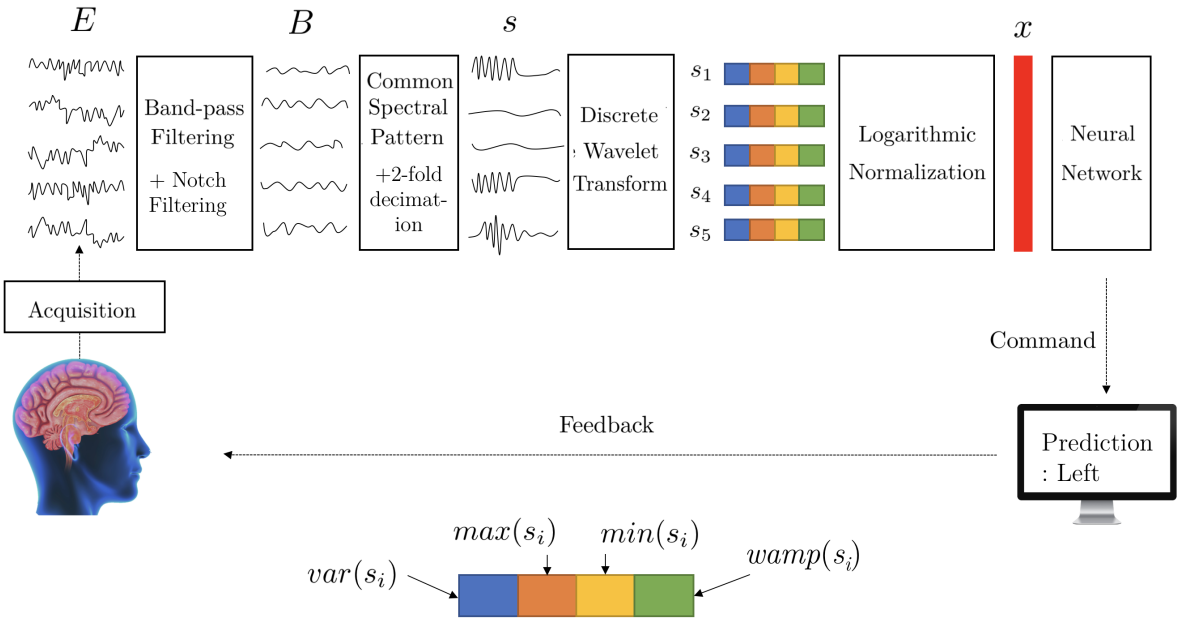


Figure 5.3: General overview of the real-time processing chain.

One will introduce some conventions. First, EEG signals used for programming and training the system are described as

$$\{E_n\}_{n=1}^{N} \in \mathbb{R}^{\text{ch x time}} \tag{5.1}$$

where N is the number of trials for training the system, ch is the the number of channels used in the BCI design. In this work, the number of channels is 5 and $time = 500$ corresponds to the number of samples used for one detection and for one channel during a 4 [s] acquisition. This data set corresponds in fact to the training data set obtained using protocol 2 with $F_s = 125$ [Hz] (see Chapter 4) with 500 trials for each motor task (right-hand, left-and an rest state). The total data set contains thus 1500 trials ($N = 1500$). B represents the output of the frequency bandpass filter.

$$\{B_n\}_{n=1}^{N} \in \mathbb{R}^{\text{ch x time}} \tag{5.2}$$

The stage that follows the frequency filter is made of two steps. Firstly, the 2-fold downsampling operation is performed. $B_{\text{downsampled}}$, also known as B_d, is the output of the downsampler.

$$\{B_{n,d}\}_{n=1}^{N} \in \mathbb{R}^{\text{ch x time}} \tag{5.3}$$

Next the spatial filter itself is implemented using a CSP filter. s represents the output of the spatial filter.

$$\{s\}_{n=1}^{N} = W \cdot \{B_{n,d}\}_{n=1}^{N} \qquad \text{with} \quad W \in \mathbb{R}^{\text{ch x ch}} \tag{5.4}$$

One question arises now : Why performing a downsampling operation after the frequency filter and not prior to it ? This question will be answered later in a dedicated section but roughly speaking it is related to a lack of logic elements on the FPGA MAX1050. So far, data dimensions remain constant. The features extraction stage, implemented with a wavelet decomposition, will decrease data dimensions. The wavelet decomposition is characterized by three properties :

1. l, the number of levels used for the decomposition. One has chosen $l = 5$. One keeps only decomposition levels and not approximation levels. (See Chapter 2.)

2. The window used for performing the decomposition which is a 'db1' window.

3. Quantities that must be extracted from each level of decomposition. One has chosen to keep four quantities from each level of decomposition and for each channel : the variance, the maximum value, the minimum value and the Willison amplitude.

In terms of dimensionality, $s \in \mathbb{R}^{\text{ch x time}}$ is a structure that contains 2500 samples. This structure is reduced to a features vector with a 100-dimension . This dimension corresponds in fact to the number of decomposition levels l used for each of the five channels (25 signals in total). Within these 25 signals, four quantities are extracted leading to a 100-dimension features vector. This vector is then log-normalized by expressing each entry of the vector between 0 and 1 and such that the sum of each entry of the vector is 1. This vector goes then trough the 'log' function. This is advantageous because features vectors may display a difference in terms of dynamic range in-between the successive trials. For example, features vectors of larger magnitude will dominate and thus bias the classification if it is not log-normalized. Moreover, log-normal transformation has shown [67] [68] an efficient transformation of unimodal distribution into a more Gaussian distribution which is useful for features classification.

The last stage of the BCI processing chain is the classification of features vectors. This is done using a feed-forward MLP neural network that has been trained. Fig. 5.4 depicts the training process performed before the real-time assessment of the software. Two additional conventions are defined :

$$\{x_{\text{train}}\}_{n=1}^{N_1} \in \mathbb{R}^{100 \text{ x } 1} \qquad \{y_{\text{train}}\}_{n=1}^{N_1} \in \mathbb{R}^{3 \text{ x } 1} \tag{5.5}$$

This represents respectively the data set of features vectors used for training the neural network ($N_1 = 1350$) and the expected motor task response. The y vector is organized as follows : (1,0,0) for a right-hand movement, (0,1,0) for a left-hand movement and (0,0,1) for a rest state.

$$\{x_{test}\}_{n=1}^{N_2} \in \mathbb{R}^{100 \text{ x } 1} \qquad \{y_{test}\}_{n=1}^{N_2} \in \mathbb{R}^{3 \text{ x } 1} \tag{5.6}$$

This represents the data set of features vectors used for testing the neural network and $N_2 = 150$. In fact, $N_1 + N_2 = N = 1500$. The training data set obtained in Chapter 4 is randomly divided into a training set and a small test set according to a 90%-10% partitioning. This means that the training database of 1500 trials is divided into 2 subsets. One of the subset is used for training the MLP while the other is used for testing this training. This allows to validate directly the classifier. We must pay attention to the fact that the test set does not represent a real-time assessment. It only represents an offline validation of the classifier. It is required for validating the coefficients of the classifier before being loaded in the BCI real-time system and eventually being assessed in real-time. Notice that this training is **supervised** and **offline**.

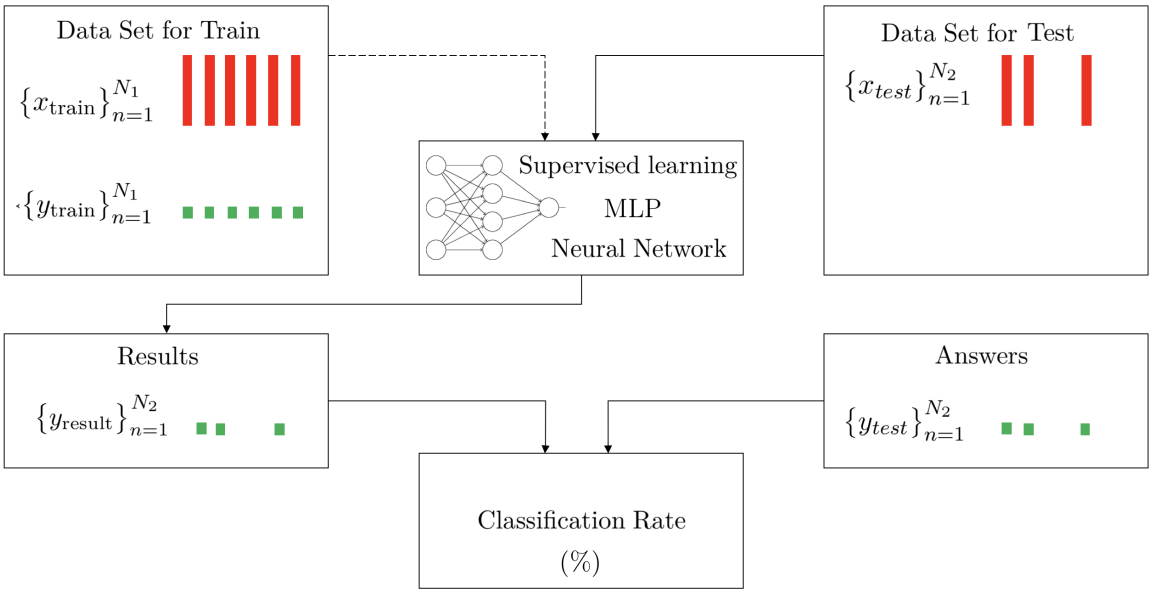


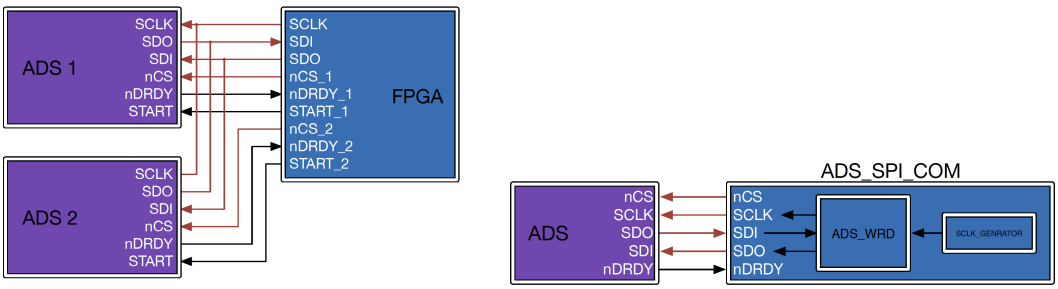
Figure 5.4: Supervised learning process of the feed-forward MLP neural network.

5.3 Verilog implementation

This section describes the first part of the real-time application which is programmed in `Verilog` language. It consists of three sub-parts : the analog-to-digital conversion through ADS1299 chips, the bandpass and notch filtering operations for each channel and eventually the softcore implementation, required for the remaining part of the processing chain is programmed in `C/C++`.

5.3.1 Analog-to-Digital conversion

Electrodes signals are acquired using two ADS1299 8-channels ADC chips implemented on the BCI board. Measured data is transmitted to the FPGA MAX1050 using a Serial Peripheral Interface (SPI). This communication protocol has been well and extensively explained in the work of Maxime Schrame and Benjamin Delhaye in [3]. The major points are described in Fig.5.5.



(a) Connections map between ADC's and FPGA [3].

(b) SPI communication module and connections with one ADC [3].

Figure 5.5: SPI protocol is used for communiaction between ADCs and the FPGA.

The SPI bus is a synchronous serial communication protocol that works in full duplex based on a master-slave structure. The bus is made of four connections : `SCLK` (master clock that allows the communication synchronization), `SDO` and `SDI` (these are the data wires, respectively leaving the master and entering the master) and eventually `nCs` (used by the master to indicate the

beginning of a communication from itself.) Fig.5.5 shows a connections scheme between the FPGA and the ADC's.

Notice that within this work, EEG signals are recorded using 5 electrodes. Therefore, using one ADC is sufficient for transmitting data to the FPGA. `ADS_SELECT` is thus constantly driven to a high-voltage, leading to the selection of the first ADC. The module responsible for the SPI interface is called `ADS_SPI_COM` and is made of two submodules.

1. `SCLK_Generator`. This clock is fixed to 6.5 [MHz] given that the ADC's are not designed to support the 52 [MHz] master clock.

2. `ADS_WRD`. It is responsible for the interface between FPGA and ADC's based on a Finite Sate Machine described in [3].

Thanks to this module, one may record data on the FPGA in order to operate the signal processing. Figure 5.6 depicts measured raw electrical signals using 5 electrodes.

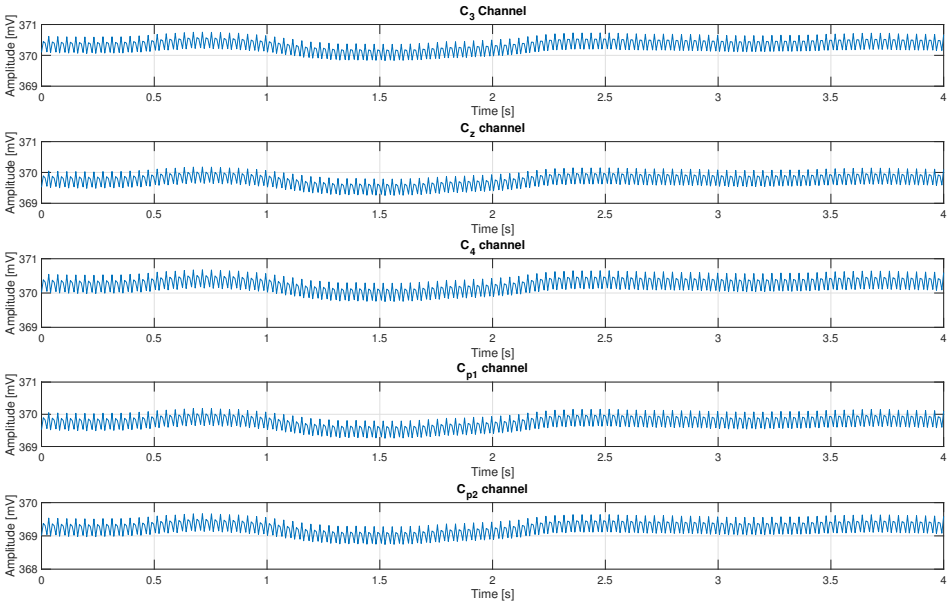


Figure 5.6: Raw electrical signals measured for C3, Cz, C4, Cp1 and Cp2.

5.3.2 Filtering

Filtering is the second operation implemented in `Verilog` programming. Each ADC channel uses a sampling rate of 250 [Hz]. A downsampling operation will be applied later in `C/C++` software. The frequency band chosen is the one corresponding to motor rythms, μ band (8-12 [Hz]), and a filter of order 16 is applied. On top of that filter, a Notch filter is used to reject the 50 [Hz] artifact due to environmental noise (AC power lines). `MATLAB` provides a filter design tool for `Verilog/VHDL` that allows to convert the filter directly into a `Verilog/VHDL` module. The two following paragraphs describe successively the design and implementation of bandpass filter and Notch filter.

5.3.2.1 Bandpass Filter

The bandpass filter is implemented as a FIR filter because such filters have three main advantages. Firstly, a FIR filter is unconditionally stable given its poles are inside the unit circle (located at [0,0] in the Gauss plan). Secondly, it has a simple mathematical description because the

current output is a linear combination of the current input sample and of several previous samples. Finally, a FIR filter may be designed to have a linear phase. IIR filters have a sharper frequency response than FIR filters but are more complicated to design and may exhibit an unstable behavior. For these reasons, the FIR filter seems to be a pragmatic choice. The equation describing a FIR filter can be derived from Eq.2.1 by eliminating the feedback term:

$$y[k] = \sum_{l=0}^{M} b_l u[k-l]$$

where M is the order of the filter and b_l are the filter coefficients. There exists an important trade-off to consider while designing a FIR filter: the delay introduced by the filter itself. For a FIR filter with symmetrical tap values ($b_{M-1-n} = b_n$), the group delay introduced by the filter is given by $\frac{M-1}{2}$. This group delay is constant for all frequencies given that there is no dependence with ω. It is also approximately equal to half of the filter order which means that increasing the filter order will lead to a larger delay. Fig. 5.7 depicts the direct-form implementation of the FIR filter. Such implementation is the most straightforward one but is not necessarily the best one in case of quantization using a few bits (8 bits quantization e.g.). In that case, one may think about Lattice or lossless Lattice realizations which show much better quantization properties.

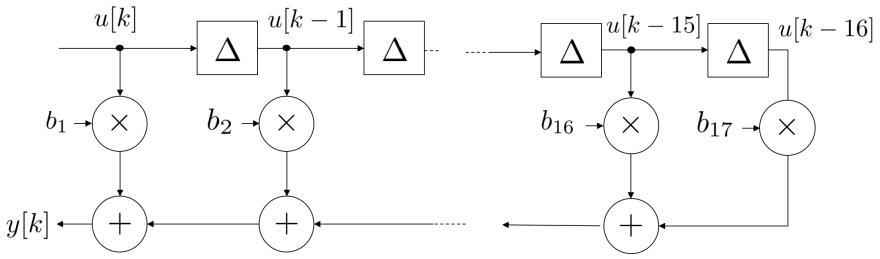
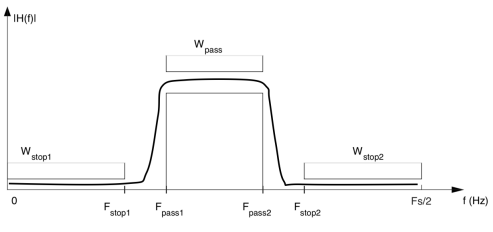


Figure 5.7: Direct-Form implementation of the bandpass filter.

The filter specifications are given in Fig.5.8. One may notice that, compared to previous work [3][4], the filter order is a design criterion although the stopband attenuation is not. The magnitude and phase responses of the filter are depicted in Fig.5.9. The quantization process in single-precision does not affect much the transfer function and therefore a direct-form implementation is sufficient.

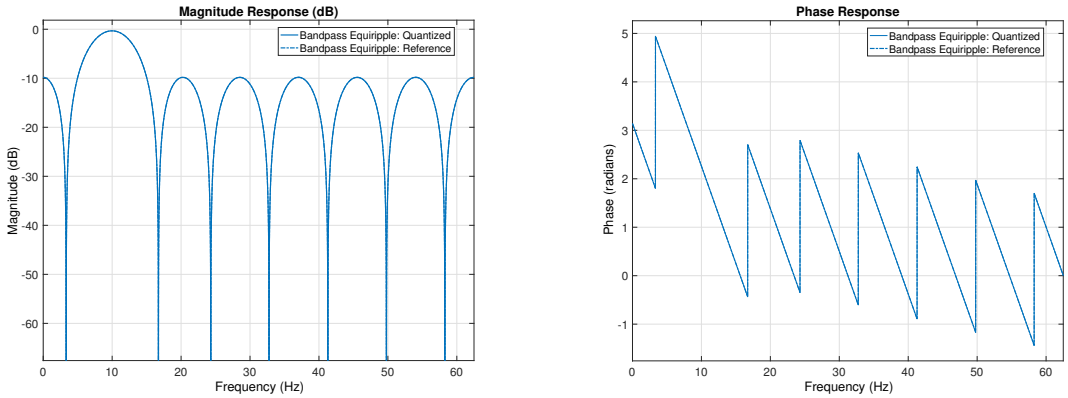
Filter Information	
Structure	Direct-Form FIR
Order	16
Quantization	Single-precision (16 bits)
Response Type	
Type	Bandpass Equiripple
Frequency Specifications	
Sampling Frequency	250 [Hz]
Low Stopband Frequency	5 [Hz]
Low Passband Frequency	8 [Hz]
High Passband Frequency	12 [Hz]
high Stopband Frequency	15 [Hz]
Magnitude Specifications	
Low Stopband Weight	0.5
Passband Weight	1
High Stopband Weight	0.5

(a) FIR bandpass filter specifications



(b) FIR bandpass filter frequency response

Figure 5.8: FIR bandpass filter specifications and expected frequency response.



(a) FIR bandpass filter magnitude response

(b) FIR bandpass filter phase response

Figure 5.9: FIR bandpass filter magnitude and phase responses.

5.3.2.2 Notch Filter

In order to eliminate the 50 [Hz] component due to environmental noise, a Notch filter is used. It filters electrical signals coming from the bandpass filters. An IIR filter is chosen for designing the Notch filter given its ability to produce a sharp response with low order filters. The equation describing an IIR filter is given by Eq.2.1. Fig. 5.10 depicts the direct-form implementation.

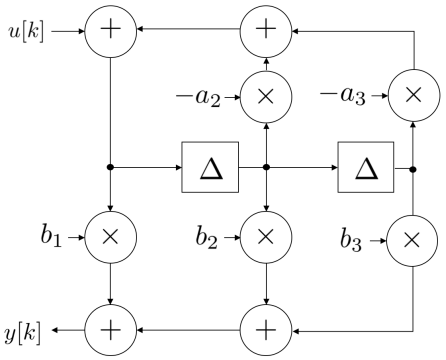
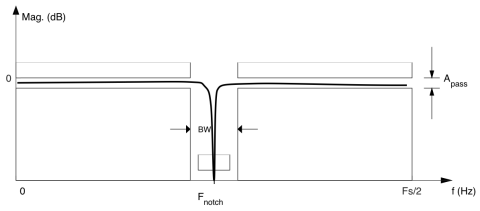


Figure 5.10: Direct-Form implementation of the Notch filter.

The filter specifications are given in Fig.5.11. The magnitude and phase responses of the filter are depicted in Fig.5.12. The quantization process does not affect much the transfer function and therefore a direct-form implementation is sufficient.

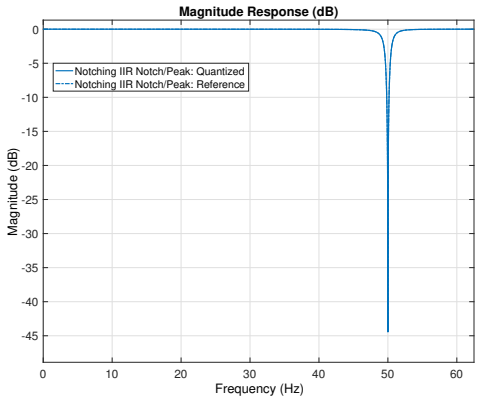
Filter Information	
Structure	Direct-Form IIR
Order	2
Quantization	Single-precision (16 bits)
Response Type	
Type	Single Notch
Frequency Specifications	
Sampling Frequency	250 [Hz]
Notch Frequency	50 [Hz]
Magnitude Specifications	
Passband Gain	1

(a) IIR Notch filter specifications

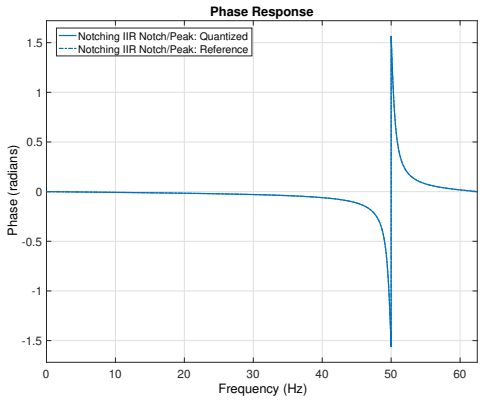


(b) IIR Notch filter frequency response

Figure 5.11: IIR Notch filter specifications and expected frequency response.



(a) IIR Notch filter magnitude response



(b) IIR Notch filter phase response

Figure 5.12: IIR Notch filter magnitude and phase responses.

5.3.3 Softcore NIOS II/f

In order to program a part of the processing chain in `C/C++`, one must insert a softcore on the FPGA. This softcore is a CPU that can be programmed in `C/C++` with an IDE such as `Eclipse`. This is the third element implemented in `Verilog`. Such a softcore will acquire data coming from bandpass and Notch filters and will provide it to the `C/C++` software. This data can then be processed in software.

Type	**Summary**	**Features**
NIOS II/f	Performance-optimized 32-bits RISC softcore.	JTAB Debug - Hardware multiply/divide - Instruction/data caches - External Interrupt Controller - MMU - MPU. - ECC RAM protection
NIOS II/e	Resource-optimized 32-bits RISC softcore.	JTAG Debug.

Table 5.1: Two different processors.

One might choose among 2 possible processors. Their features are shown in Table 5.1. One will use a NIOS II/f as softcore because the hardware multiplications and divisions are important for speeding up the execution of the processing chain and respecting the timing constraints imposed in Chapter 4. Indeed, numerous operations use a huge amount of multiplications and divisions, e.g. the DWT or the MLP neural network.

The specifications of the softcore are defined here :

1. An on-chip memory of 112 [kB], a data cache of 4 [kB] and an instruction cache of 4 [kB] for a total memory size of 120 [kB].
2. No burst-count signals related to the data/instruction cache. The burst mode is not used in this work.
3. 32-bits multiplication using 3 16-bits multipliers. This leads to a multiplication in 1 cycle.
4. 32-bits division using logic elements/SRT Radix-2. This leads to a division in 35 cycles.
5. Shift and rotate operations in 1 cycle thanks to pipelined logic elements organization.

Given this softcore, one may draw a map of the whole `Verilog/Qsys` design. This diagram is depicted in Fig.5.13. One may analyze this block diagram :

(1) The NIOS II/f. This represents the central element of the `Verilog` design and it is the only unit which is connected to all other units. Indeed, it controls all the external interfaces such as the 4-bits LEDs, the 4-bits switches, the 2 buttons or the JTAG UART. It also interacts with internal modules. These modules correspond to `myBPF` (5 modules in total, one for each channel), `myNotch` (5 modules in total, one for each channel), `ADS_SPI_COM` or the on-chip memory. Notice that this softcore has been implemented in this work.

(2) myBPF. This represent in fact five different modules that aim at bandpass filtering each EEG channel as defined earlier. It takes 24-bits data from ADCs trough `ADS_SPI_COM` as input, filter it and provide it to `myNotch` module as output. Notice that this module has been implemented first in [3] and filters coefficients have been modified in this work.

(3) myNotch. This represent in fact five different modules that aim at Notch filtering each EEG channel as defined earlier. It takes 24-bits filtered data from `myBPF` as input, process it and provide it to the NIOS II/f. Notice that this module has been implemented first in [4] and is re-used in this work.

(4) ADS_SPI_COM. This module has been extensively explained earlier. It allows the SPI communication between the ADCs and the FPGA. Notice that this module has been implemented first in [3] and is re-used in this work.

(5) On-chip memory. This is a built-in module that up to 120 [kB] of RAM memory to the NIOS II/f in order for the softcore to be able to execute the `C/C++` code.

(6) PIOs. The different PIOs consist simply in an interface for external hardware and/or internal modules. Here the PIOs are used for controlling LEDs/switches/buttons from the FPGA. Remark that filtered data coming from the filters modules are transferred to the CPU using PIOs too.

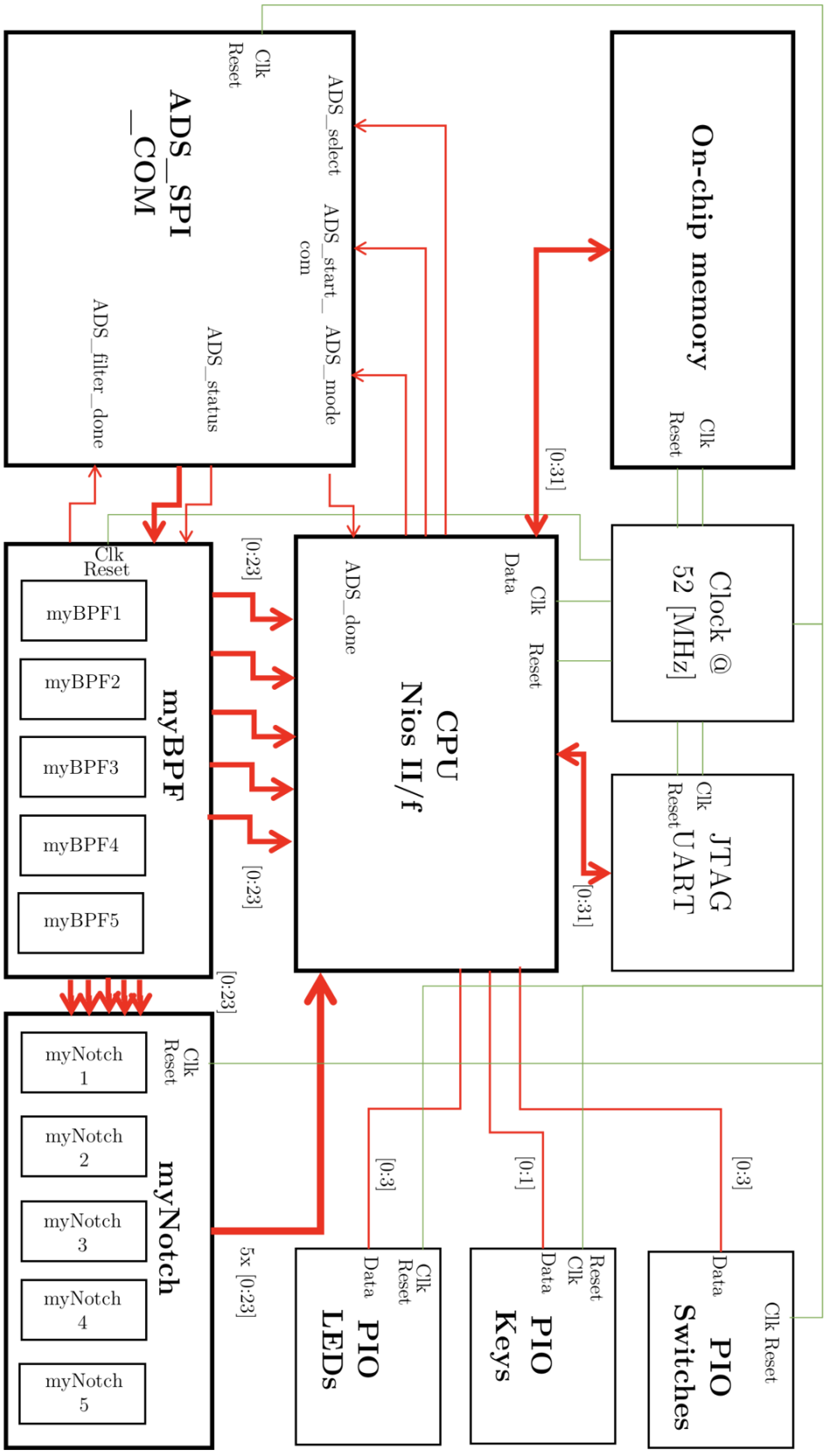


Figure 5.13: Block scheme of the Qsys and Verilog design.

5.4 C/C++ Implementation

This section describes the second part of the real-time application which is programmed in `C/C++` languages. It contains most of the processing chain including downsampling operations, the spatial filter, the features vector extraction based on discrete wavelet transform and the feature translation/classification based on a feed-forward MLP neural network. First and foremost, the choice of developing the most important part in `C/C++` software is justified by two reasons :

1. Developing in `C/C++` reduces the programming complexity and hence accelerates the development process. It allows indeed to use standard libraries such as `<math.h>` which is considerably time saving.

2. Considering a long-term development over years `C/C++` code is much more readable and easily adaptable than `Verilog` code.

This section is organized in five parts which represents the five blocks implemented in `C/C++` software : filtered signals acquisition over 5 channels in section 5.4.1, the downsampling operation in section 5.4.2, the spatial filter in section 5.4.3, the features vector extraction in section 5.4.4 and the features vector classification in section 5.4.5. For each part, the algorithm is presented and the input-output behavior using real data is shown.

5.4.1 Filtered signals acquisition

The first step of the real-time software is the acquisition of EEG data over 5 channels during a certain acquisition time $T_{\text{acq}} = 4$ [s]. As mentioned earlier in this chapter, the BCI implemented in this work is a synchronous one. It records first filtered EEG data (frequency filtering is programmed in `Verilog`) during a pre-defined acquisition time and process recorded signals afterwards. Recorded samples are stored in a $B_n \in \mathbb{R}^{5 \text{ x } 1000}$ matrix according to

$$B_n = [C3 \quad Cz \quad C4 \quad Cp1 \quad Cp2]^{\text{T}} \tag{5.7}$$

where each channel is a column vector (in $\mathbb{R}^{1000 \text{ x } 1}$). Notice that the representation of B_n is used for convenience. In practice one does not store the entire B_n matrix but only the downsampled version of B_n.

5.4.2 Downsampling

At this point, raw data coming from 5 electrodes are filtered in 2 steps : a Notch filter for 50 [Hz] rejection and a bandpass filter that keeps μ rhythm (8-12 Hz) for each channel. The resulting signals are transmitted to the NIOS II/f softcore in the form of a matrix $B_n \in \mathbb{R}^{5 \text{ x } 1000}$. Given the relatively small RAM memory available, we will perform a 2-fold downsampling. This will lead to the possibilities explained in Table 5.2

250 [Hz]-sampling memory size	**125 [Hz]-sampling memory size**	**Memory saving**
5 channels x 250 Hz x 4 bytes x 4 [s] = 20 [kB]	5 channels x 125 Hz x 4 bytes x 4 [s] =10 [kB]	10 [kB]

Table 5.2: Comparison of memory usage for acquiring EEG signals with two different sampling rates.

With a 2-fold downsampling, one will save about 10 [kB] which represent 8.33% of the total RAM memory. Moreover, given that the BCI must record sensory-motor rhythms around the μ rhythm (8-12 [Hz]) and that the ceiling frequency of SMR rhythms is the γ rhythm (max 45

[Hz]), choosing a sampling frequency of 125 [Hz] is valid. Indeed, such a sampling frequency gives a bandwidth of 62.5 [Hz] which is above the largest frequency content due to γ rhythm. There is no aliasing effect risks.

A downsampling operation is performed in two steps. First, an anti-aliasing filter is used for safety in case of an input signal that occupies a frequency band larger than $\omega = Fs/2 = 62.5$ [Hz] (π in normalized frequency). Using such a filter ensures to avoid spectral folding phenomena. From a block diagram point of view, implementing a 2-fold downsampling operation consists in keeping one sample out of two. Fig.5.14 depicts the block scheme implementation and algorithm 1 presents the implementation.

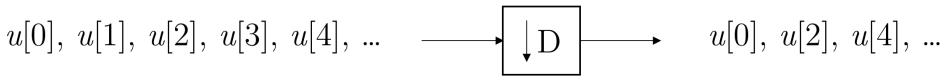


Figure 5.14: Block scheme implementation of a 2-fold downsampler.

Algorithm 1 2-fold downsampling.

Require: $C3[k]$ $Cz[k]$ $C4[k]$ $Cp1[k]$ $Cp2[k]$
  $k \leftarrow 0$
  $l \leftarrow 0$
  $F_s \leftarrow 250$
  $T_{\text{acq}} \leftarrow 4$
  for $k < F_s \cdot T_{\text{acq}}$ **do**
    if l is even **then**
      $C3_d[l] = C3[k]$
      $Cz_d[l] = Cz[k]$
      $C4_d[l] = C4[k]$
      $Cp1_d[l] = Cp1[k]$
      $Cp2_d[l] = Cp2[k]$
    end if
    $k \leftarrow k + 1$
  end for

The downsampling operation produces, as output, vectors which have half the size of inputs vectors. This leads to a loss of accuracy which is nevertheless not significant. Indeed one may observe the effect of the downsampling operation on a filtered C3 vector in figure 5.15.

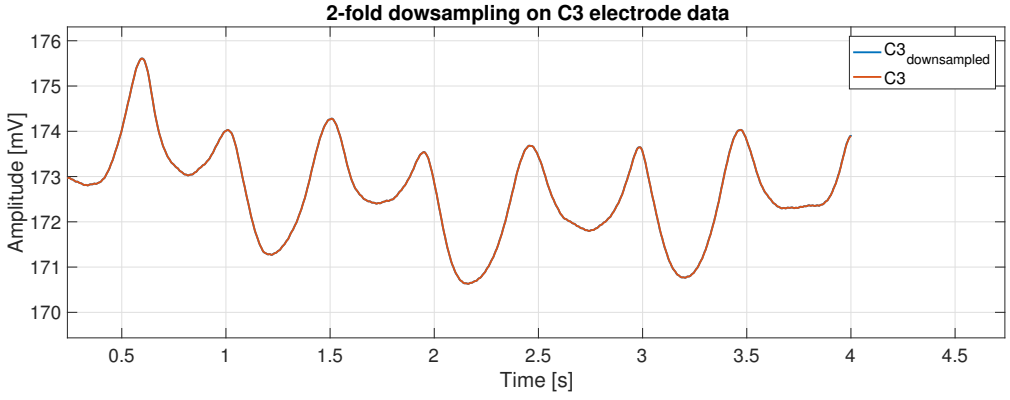


Figure 5.15: 2-fold downsampling of filtered C3 electrode data.

By observing this figure, one can not notice the effects of the 2-fold downsampling operation.

5.4.3 Spatial Filtering

At this level of the processing chain, the input is the downsampled and filtered data from the five electrodes. Data is organized in a matrix $B_{n,\text{downsampled}} \in \mathbb{R}^{5 \text{ x } 500}$ where each row is a vector containing the 4-seconds record for one channel. The purpose of spatial filtering is to enhance sensitivity to particular brain sources, to improve source localization, and/or to suppress certain artifacts. Spatial filters are generally performing a linear combination of input channels and are hence denoted by :

$$s = W \cdot B_{\text{downsampled}} \qquad \text{with} \quad W \in \mathbb{R}^{\text{ch x ch}} \tag{5.8}$$

where $B_{\text{downsampled}}$ is expressed as :

$$B_{\text{downsampled}} = [C3_d \quad Cz_d \quad C4_d \quad Cp1_d \quad Cp2_d]^{\text{T}} \tag{5.9}$$

Each electrode signal is considered as a column vector that contains 500 samples (4 seconds of acqusition at 125 [Hz]). Therefore, each row of $B_{\text{downsampled}}$ consists of filtered electrodes signal samples from C3 to Cp2 channel. Each row of W is a set of 5 channel weights that constitute a particular spatial filter. Eventually, each row of s is a resulting spatially filtered channel. Results obtained in the previous chapters indicate that CSP is the most appropriate choice of spatial filter. As a reminder, the purpose of CSP filtering is to discriminate two categories of signals, one with maximum variance and the other with minimum variance (here, these two categories are right-hand and left-hand movements). CSP is an approach that is based on four parts, as described in Fig.5.16.

1. Record a training data set for the three tasks implemented in this BCI design : right-hand movement, left-hand movement and rest state.

2. Produce two covariance matrix using sets of data : the right-hand movement mean covariance matrix R_{right} and the left-hand movement mean covariance matrix R_{left}. Each matrix is a 5-by-5 matrix that contains the mean spatial covariance of each channel. Why mean ? It is related to the fact that one uses the whole training database for establishing the covariance matrices, by taking the average of all the covariance matrices for each movement.

3. Solve an eigen-values problem offline that produces components with minimum variance for one condition (e.g. right-hand movement) and maximum variance for the other (e.g. left-hand movement). The eigen-values problem returns W.

4. Using W, one performs a linear combination of real-time recorded electrode data such that the output s is made of high-variance signals for one condition (e.g. right-hand movement) and low-variance signal for the other condition (e.g. left-hand movement). As one may see in Fig.5.16, the linear combination of input signals creates a low-variance signal and high-variance signal based on the training set.

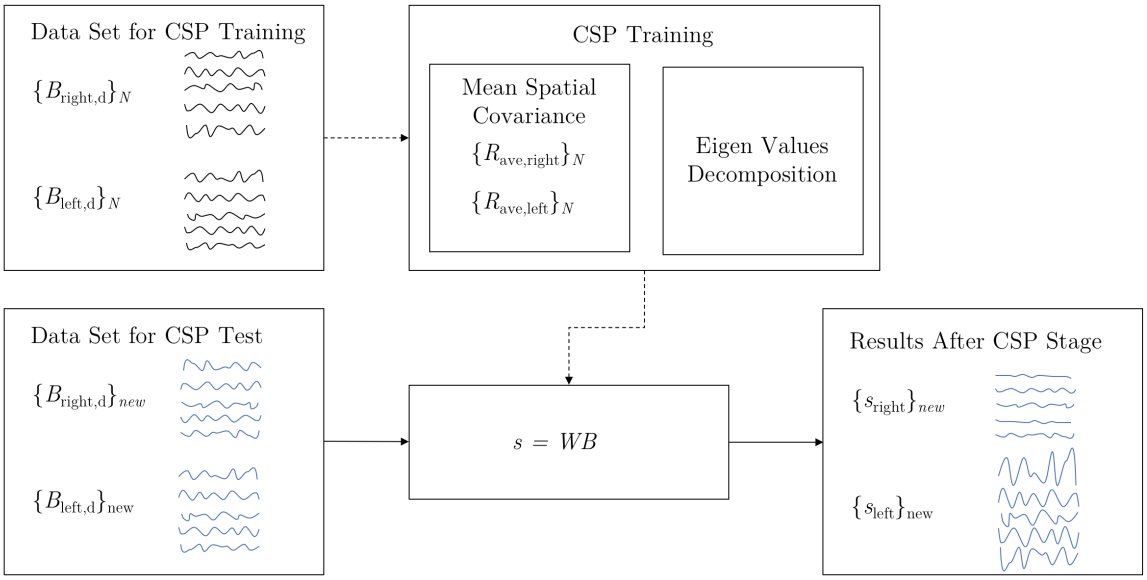


Figure 5.16: Common spectral pattern block scheme. It requires a training process for obtaining the weight matrix W. The output of CSP spatial filter is a low-variance signal for right-hand movement and a high-variance signal for right-hand movement.

Notice that CSP belongs to data-dependent spatial filters category because it is derived directly from each BCI data and is not based on geometrical considerations (Laplacian filters e.g.). Such filters tend to be more complex because they might need a training phase and/or more complex processing operations but generally produce better results. This section first describes the training algorithm executed offline on `MATLAB` followed by the application of the filter within the real-time software. Afterwards, some results are presented based on recorded signals are presented. In order to simplify notations, $B_{\text{downsampled}}$ is changed into B_d and still refers to the 5-by-500 matrix of downsampled electrodes signals.

5.4.3.1 Training and applying Common Spectral Pattern

The training part of CSP takes a training data set made of right- and left-hand movement trials (1000 trials in this work, i.e. 500 right-hand movements recorded during 4 seconds each. The same holds for left-hand movements) as input and produce a $W \in \mathbb{R}^{5\text{ x }5}$ as output. From a mathematical point of view, a convex quadratic optimization problem allows to obtain W :

$$\begin{aligned} \underset{W}{\text{maximize}} \quad & \text{tr}(W^{\text{T}}\Sigma_1 W) \\ \text{subject to} \quad & W^{\text{T}}(\Sigma_1 + \Sigma_2)W = I \end{aligned} \tag{5.10}$$

In the aforementioned equation, Σ_1 and Σ_2 are computed as :

$$\Sigma_1 = \underset{B_d \in \text{class 1}}{\exp} \frac{B_d B_d^{\text{T}}}{\text{tr}(B_d B_d^{\text{T}})} \qquad \Sigma_2 = \underset{B_d \in \text{class 2}}{\exp} \frac{B_d B_d^{\text{T}}}{\text{tr}(B_d B_d^{\text{T}})} \tag{5.11}$$

A solution to that problem can be found by solving two times in a row the standard eigen value problem. The algorithm used for solving this problem is described in 2. Once W matrix is known, one may generate the output of the spatial filter by performing a matrix multiplication $s = W \cdot B_d$ where s is the output of the spatial filter and W is just obtained thanks to algorithm 2. This is depicted in algorithm 3.

Algorithm 2 Common spectral pattern training.

Require: $\{B_{\text{right,d}}\}_N$, $\{B_{\text{left,d}}\}_N$ for $N = 500$

$k \leftarrow 0$

$\Sigma_1 \leftarrow \mathbf{0}^{5\text{x}5}$

$\Sigma_2 \leftarrow \mathbf{0}^{5\text{x}5}$

for $k < N$ **do**

$\Sigma_1 \leftarrow \Sigma_1 + \frac{\{B_{\text{right,d}}\}_k^{\text{T}} \{B_{\text{right,d}}\}_k}{\text{tr}\{B_{\text{right,d}}\}_k^{\text{T}} \{B_{\text{right,d}}\}_k}$ //Spatial covariance matrix for trial k of right-hand class

$\Sigma_2 \leftarrow \Sigma_2 + \frac{\{B_{\text{left,d}}\}_k^{\text{T}} \{B_{\text{left,d}}\}_k}{\text{tr}\{B_{\text{left,d}}\}_k^{\text{T}} \{B_{\text{left,d}}\}_k}$ //Spatial covariance matrix for trial k of left-hand class

$k \leftarrow k + 1$

end for

$\Sigma_1 \leftarrow \Sigma_1 / N$ //Mean covariance matrix for right-hand class

$\Sigma_2 \leftarrow \Sigma_2 / N$ //Mean covariance matrix for left-hand class

$\Sigma_1 + \Sigma_2 = UDU^{\text{T}}$ //U is a set of eigen vectors and D is diagonal

$P = \sqrt{D^{-1}} U^{\text{T}}$

$\widehat{\Sigma_1} = P \Sigma_1 P^{\text{T}}$

$\widehat{\Sigma_2} = P \Sigma_1 P^{\text{T}}$

$\widehat{\Sigma_1} = V \Gamma V^{\text{T}}$ //V is a set of eigen vectors and Γ is diagonal

$W = P^{\text{T}} V$

Algorithm 3 Common spectral pattern application.

Require: W, B_d

$k \leftarrow 0$

for $k < 500$ **do**

$s(1,k) = W(1,1) \cdot B_d(1,k) + W(1,2) \cdot B_d(2,k) + W(1,3) \cdot B_d(3,k) + W(1,4) \cdot B_d(4,k) + W(1,5) \cdot B_d(5,k)$

$s(2,k) = W(2,1) \cdot B_d(1,k) + W(2,2) \cdot B_d(2,k) + W(2,3) \cdot B_d(3,k) + W(2,4) \cdot B_d(4,k) + W(2,5) \cdot B_d(5,k)$

$s(3,k) = W(3,1) \cdot B_d(1,k) + W(3,2) \cdot B_d(2,k) + W(3,3) \cdot B_d(3,k) + W(3,4) \cdot B_d(4,k) + W(3,5) \cdot B_d(5,k)$

$s(4,k) = W(4,1) \cdot B_d(1,k) + W(4,2) \cdot B_d(2,k) + W(4,3) \cdot B_d(3,k) + W(4,4) \cdot B_d(4,k) + W(4,5) \cdot B_d(5,k)$

$s(5,k) = W(5,1) \cdot B_d(1,k) + W(5,2) \cdot B_d(2,k) + W(5,3) \cdot B_d(3,k) + W(5,4) \cdot B_d(4,k) + W(5,5) \cdot B_d(5,k)$

$k \leftarrow k + 1$

end for

5.4.3.2 Results

One will examine now the effects of the CSP spatial filter on electrodes signals. One may prove the effectiveness of CSP spatial filters by applying it for EEG electrodes signals. As reminder, the input-output relationship is a linear combination with weights contained in a matrix W. Fig.5.17 and Fig.5.18 depict the application of the CSP spatial filter for two different situations. In Fig.5.17, a right-hand movement is executed during 3 [s] and and is recorded by 5 electrodes (C3, Cz, C4, Cp1 and Cp2). That corresponds to the five left plots of Fig.5.17. On the right side of the same figure, resulting signals coming out of the spatial filter are denoted by s_1 to s_5. In Fig.5.18, a left-hand movement is executed during 3 [s] and and is recorded by 5 electrodes (C3, Cz, C4, Cp1 and Cp2). That corresponds to the five left plots of Fig.5.18. On the right side of the same figure, resulting signals coming out of the spatial filter are denoted by s_1 to s_5.

One may give an interpretation of these results. First, we need to notice that electrodes signals have similar wave shapes with slightly different amplitudes before undergoing the spatial filter. After the CSP filter, however, signals have different wave shapes. These wave shapes represents in fact signals coming from different and independent brain sources. Therefore, the CSP allows to enhance sensitivity to particular brain sources.

We will now try to show that each channel s_i is not necessarily as informative as others. For example, s_2 illustrates the working principle of CSP. Indeed, s_2 is related to the right-hand movement in Fig.5.17 and has a highly-varying behavior which is associated with a large variance value. In Fig.5.18, s_2 is related to the left-hand movement, is rather flat and hence has a less varying amplitude. This allows to discriminate well the two movements. This behavior can be noticed in s_1 too but is harder to identify in s_5. Therefore, some components may sometimes be useless for the features vector. However knowing in advance which components provide less information than others is a difficult task and therefore, no channel reduction is performed in this BCI design.

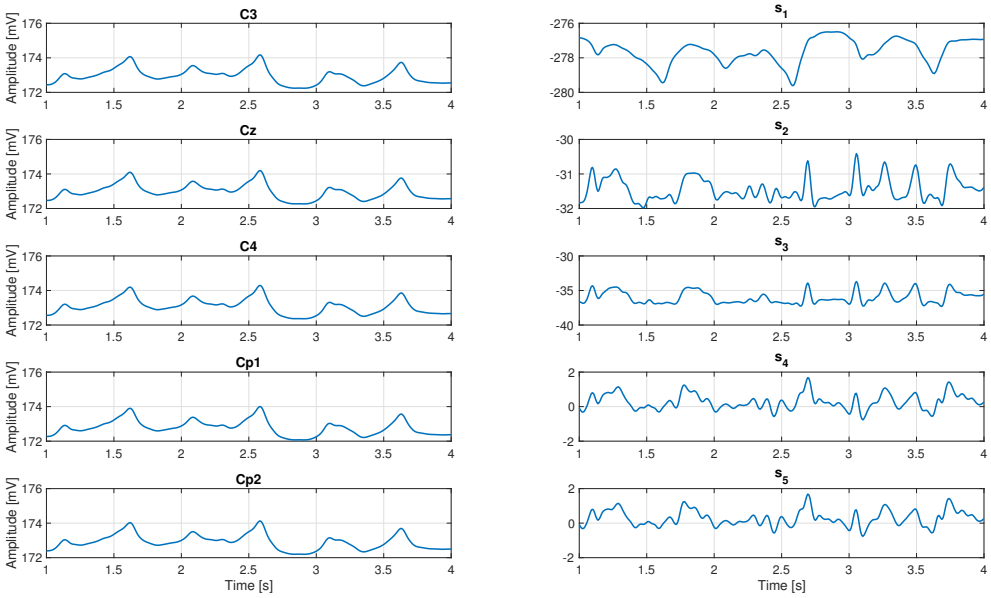


Figure 5.17: Common spectral pattern results for a right-hand movement of 3 [s].

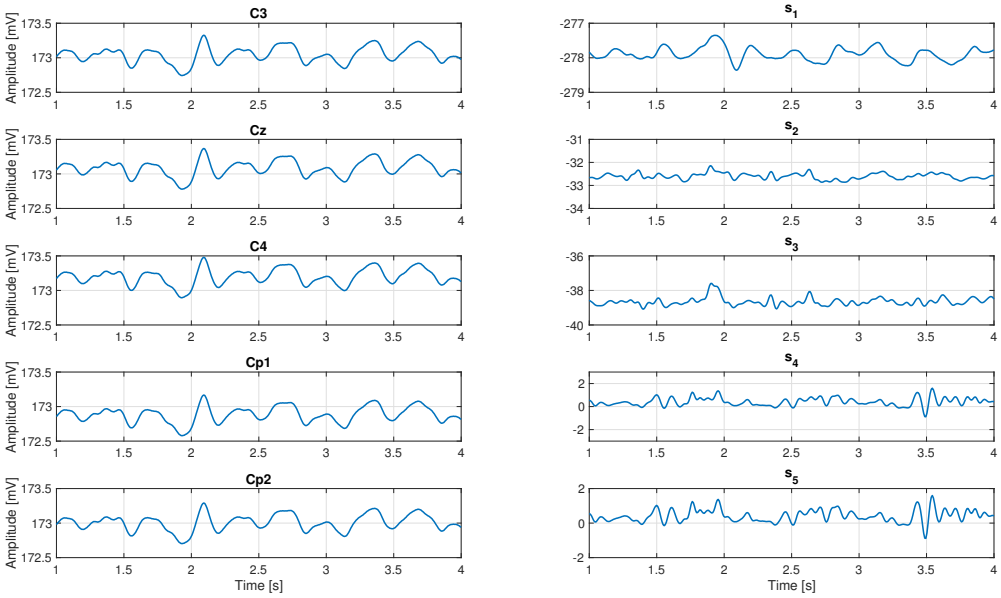


Figure 5.18: Common spectral pattern results for a left-hand movement of 3 [s].

5.4.4 Features extraction

After having reduced artifacts using a spatial filter based on the CSP approach, one must establish a features vector (i.e. a vector containing most important and discriminate information) based on a matrix $s \in \mathbb{R}^{5 \text{ x } 500}$ that keeps only relevant information and realizes a dimensionality reduction in order to be classified efficiently in real-time. By inspecting results obtained in the previous chapters, the best features that one may select among those presented is based on a discrete wavelet decomposition over 5 levels using a Daubechies window of first order ('db1'). For each decomposition level, four quantities are extracted : the variance, the maximum value, the minimum value and the Willison amplitude (WAMP). This section is organized into three parts. First, a small reminder on wavelet transform, DWT and `db1` window will be made. Then, the block diagram and implementations are detailed. Finally, some results are presented.

5.4.4.1 The Wavelet Transform

As explained in Chapter 2 related to the processing chain, the major advantage of wavelet analysis is its time-frequency resolution. Influence of temporal and spectral resolution is not dependent on the selected hyper-parameters (such as the model order in an auto-regressive features extraction e.g.). Compared to Fourier transform, the basis functions are not sinus functions but wavelets with compact support (i.e. localized in time). Due to that compactness, WT is well adapted for non-stationary signals. On top of the continuous wavelet transform (CWT), there exists a discrete wavelet transform (DWT) which is implemented for the BCI design in this work.

The DWT structure is based on a filterbank decomposition, as depicted in Fig.2.6. The input signal $x(t)$ is filtered by a low- and high-pass filter and then downsampled by a factor 2 to produce a detailed signal (also known as D_i, where i is the decomposition level) and an approximation signal (also known as A_i, where i is the decomposition level). This is based on a mother wavelet and a recurrence scheme for generating, step by step, a smoother discrete signal with less and less high frequency details. With this decomposition, high frequency content is captured in first

stages. Due to successive 2-fold downsampling operations, the sampling rate decreases. It starts at 125 [Hz] and ends up at 7.8125 [Hz]. This means that high frequencies content has an accurate time (due to high sampling rate) localization which is useful because high frequency signals vary rapidly. Regarding low frequency content, the decomposition captures it progressively at the end of the transform. This means that this content has a rough sampling rate leading to a bad localization in time (due to low sampling rate). However, the presence of a low frequency content means that the signals are slowly varying and therefore, one may use a rough time localization without impacting performances.

Regarding the window choice, one will use a 'db1' wavelet. This is in fact equivalent to the 'Haar' window. This window leads to the best classification using the `MATLAB` application as described in chapter 4. This wavelet has several advantages :

1. This is the simplest wavelet. It has only 2 taps/coefficients. Such a wavelet is thus memory efficient and has a low computational cost.

2. This wavelet belongs to an orthogonal wavelet basis, which means that it preserves the energy contained in the signal. Once again, this property is crucial for a BCI design. Indeed, recorded EEG signals from the scalp have a very low energy content which should be preserved as much as possible.

5.4.4.2 Block diagram and implementation

One must establish a features vector based on $s \in \mathbb{R}^{5 \text{ x } 500}$ which corresponds to the output of the spatial filter. Therefore, each signal contained in s will be subject to a 5-levels DWT using a `db1`. For each detailed signal, 4 quantities will be extracted : the variance, the maximum value, the minimum value and the Willison amplitude (WAMP). Figure 5.19 depicts a detailed block diagram of the features extraction stage. This process is executed in two stages.

1. Each s_i signal from the spatial filter stage is subject to a wavelet decomposition in 5 levels. Each level is characterized by a specific frequency band as depicted in table 5.3 and figure 5.20. At this point, one generates 25 signals : 5 detailed signals from D1 to D5 for each s_i signal from the spatial filter. One has thus :

$$U = [s1_{D1} \text{ ... } s1_{D5} \quad s2_{D1} \text{ ... } s2_{D5} \quad s3_{D1} \text{ ... } s3_{D5} \quad s4_{D1} \text{ ... } s4_{D5} \quad s5_{D1} \text{ ... } s5_{D5}]^{\mathrm{T}} \tag{5.12}$$

   The structure U has 25 rows while the number of columns varies with the decomposition level. This is described in table 5.3. The shape of U structure is described in figure 5.21.

2. The second stage is the extraction of some quantities from the U structure. For each row of U, four quantities are extracted and set into the features vector. The features vector is thus $x \in \mathbb{R}^{100 \text{ x } 1}$. The first quantity is the variance :

$$\text{var}(si_{Dj}) = \frac{1}{N-1} \sum_{k=1}^{N} (si_{Dj}[k] - \mu_{si_{Dj}})^2 \qquad \forall i, j \in [0; 5] \tag{5.13}$$

$$\max(si_{Dj}) \qquad \forall i, j \in [0; 5] \tag{5.14}$$

$$\min(si_{Dj}) \qquad \forall i, j \in [0; 5] \tag{5.15}$$

$$\text{WAMP}(si_{Dj}) = \sum_{k=1}^{N-1} f(|si_{Dj}[k] - si_{Dj}[k+1]|) \qquad \forall i, j \in [0; 5] \tag{5.16}$$

   In the Willison amplitude, f is defined as :

$$f(x) = \begin{cases} 1, & \text{if } x > t \\ 0, & \text{otherwise} \end{cases}$$

With t is a given threshold. For the BCI real-time system design, a threshold value of 0.2 has proved to be the most effective in terms of classification rate. These quantities have been chosen among others such as the mean or the average amplitude change because these ones produce most discriminate features vectors. This leads to a better classification. One may give an interpretation of these quantities. The variance represents the power contained within the signal. A motor task produces EEG signals that contain more power, and thus more variance than a rest state. The maximum and minimum give also an idea of the amplitude of a given signal. Eventually, the WAMP (Willison amplitude) compare two successive samples and gives therefore an idea of the rate of change. With the Willison amplitude, if this rate of change is larger than a given threshold, the feature extracted is 1. Otherwise it is 0. Therefore it produces different features for signals that vary rapidly and those which vary slowly.

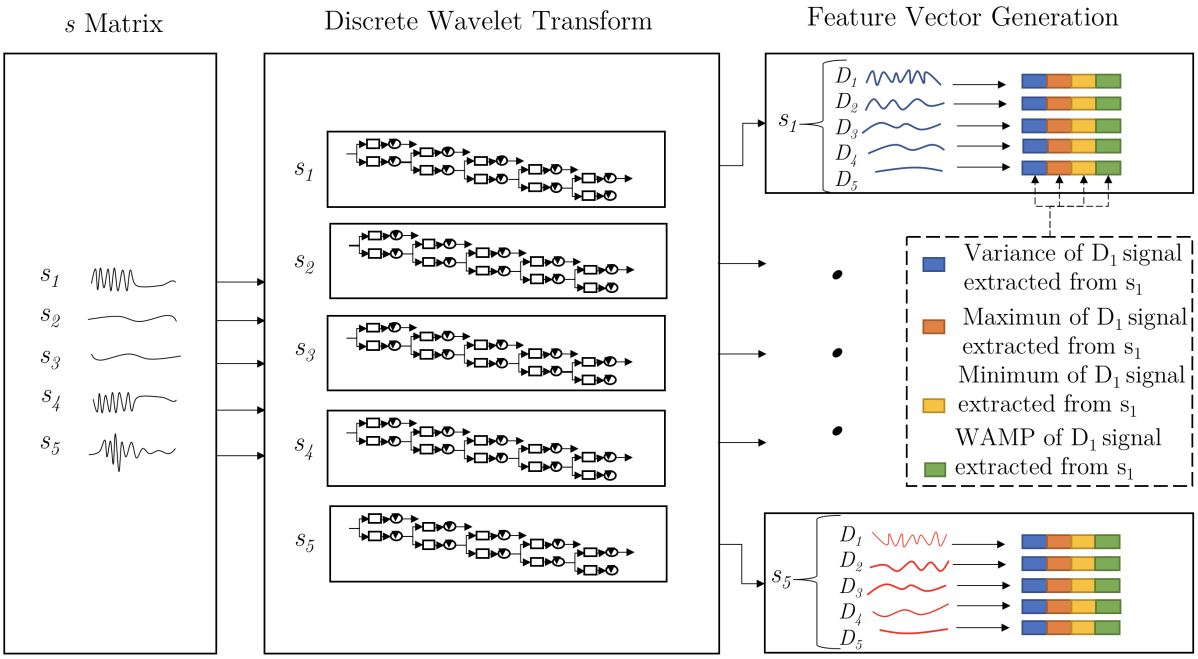


Figure 5.19: Detailed block diagram of the features extraction stage. First, each row of the s matrix undergoes a DWT of of 5 levels using a Daubechies window leading to 5 detailed signals (D1 to D5). After this stage, 25 signals are obtained (5 detailed signals for each of the 5 electrodes signals coming from the spatial filter). Four quantities are extracted from each of these 25 signals : The variance, The maximum, The minimum and the Willison amplitude.

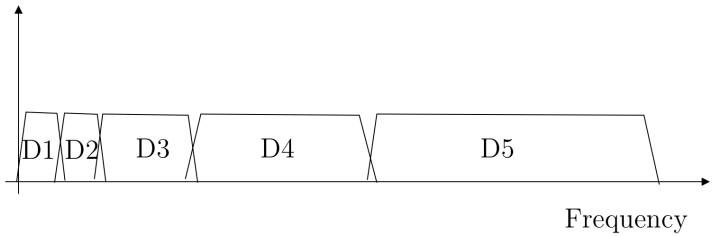


Figure 5.20: Bandwidth of each of 5 decomposition levels of the DWT.

Decomposition Level	Frequecy Range [Hz]	Brain Frequency Bands	Signal length [samples]
D5	0 - 4.16	θ	16
D4	4.16 - 8.31	δ	32
D3	8.31 - 16.62	μ and α	63
D2	16.62 - 31.25	β	125
D1	31.25 - 62.50	γ	250

Table 5.3: Successive decomposition level of each s_i signal generates 5 detailed signals that have different frequency ranges and bandwidths. Notice that each detailed signal also has a different length : $s1_{D1}$ has a length of 250 samples while $s1_{D5}$ has a length of 16 samples for example.

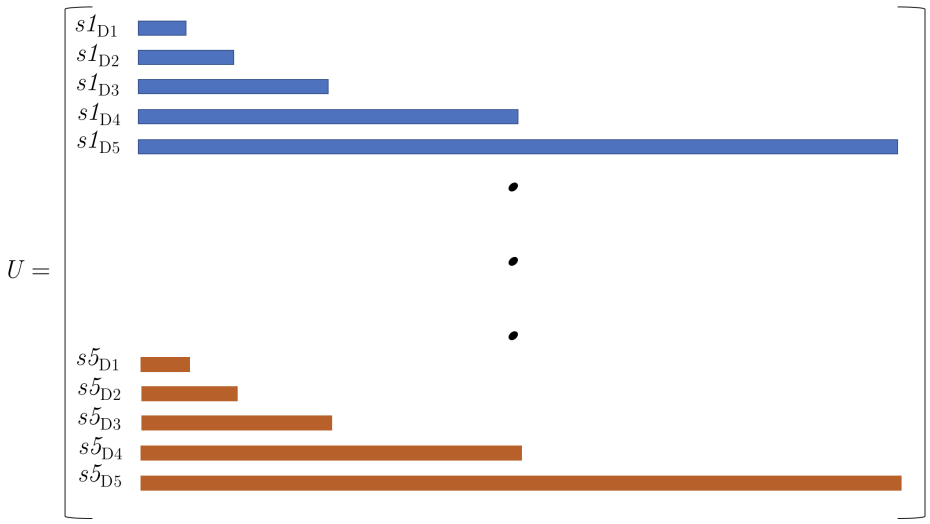


Figure 5.21: Shape of U structure that contains 5 decomposition signals for each of the 5 s_i signals.

The block scheme of the features extraction process is at this point clearly defined. Algorithm 4 and 5 are used for realizing this process. Notice that D_{high} represents the first-order Daubechies window ('db1') high-pass filter and D_{low} represents the first-order Daubechies window ('db1') low-pass filter (see Fig.5.22).

Once the decomposition is realized and quantities of interest are extracted, one must perform the log-normalization as defined in section 5.2. As mentioned in this section, one may find in the literature [67] [68] that in many cases unimodal non-Gaussian distributions can be shaped to have a more Gaussian distribution by applying a monotonically increasing nonlinear transformation. Such a transformation guarantees that the ordering of the features values remains the same with different spacing between the values. Algorithm 6 performs this transformation.

5.4.4.3 Results

The whole feature extraction stage is made of thee steps : DWT, quantities extraction for each decomposition level and log-normalization. One will now present some results produced by algorithms aforementioned in order to validate them. The first results depicted will show the DWT decomposition of a signal s_1 for a right-hand movement. One may notice in Fig.5.23

Algorithm 4 Discrete Wavelet Transform (part 1).

Require: $s \in \mathbb{R}^{5x500},\ D_{\text{high}} \in \mathbb{R}^{3x1},\ D_{\text{low}} \in \mathbb{R}^{3x1}$

```
k ← 0
l ← 0
for k < 5 do
  for l < 5 do
    s_high(l,:) = filter(D_high, 1, s(l,:))      //filter() is a defined in the sense of MATLAB.
    s_low(l,:) = filter(D_low, 1, s(l,:))        //filter() is a defined in the sense of MATLAB.
    s_high(l,:) ← s_high(l,:) ↓ 2                //2-fold downsampling.
    s_low(l,:) ← s_low(l,:) ↓ 2                  //2-fold downsampling.
    U((l+1)*5−k) = s_high(l,:)                   // Assigning the detailed signal inside the U strcuture.
    s(l,:) ← s_low(l,:)                          //reassigning s fro next decomposition level.
    l ← l + 1
  end for
  k ← k + 1
end for
```

Algorithm 5 Discrete Wavelet Transform (part 2).

Require: $U,\ x \in \mathbb{R}^{100x1}$

```
k ← 0
var_curr = 0
max_curr = 0
min_curr = 0
wamp_curr = 0
for k < 25 do
  var_curr = var(U(k,:))        //var() is a defined in the sense of MATLAB.
  max_curr = max(U(k,:))        //max() is a defined in the sense of MATLAB.
  min_curr = min(U(k,:))        //min() is a defined in the sense of MATLAB.
  wamp_curr = wamp(U(k,:))      //wamp() is a defined in the sense of MATLAB.
  x(k*4) = var_curr
  x(k*4+1) = max_curr
  x(k*4+2) = min_curr
  x(k*4+3) = wamp_curr
  k ← k + 1
end for
```

Algorithm 6 Discrete Wavelet Transform (part 3).

Require: $x \in \mathbb{R}^{100x1}$

```
k ← 0
sum ← = sum(x) //sum() is a defined in the sense of MATLAB.
for k < 100 do
  x(k) ← log(x(k)/sum)
  k ← k + 1
end for
```

that the D_1 level contains numerous samples and hence characterizes high-frequency content. Progressively high-frequency details vanish when the decomposition progresses. Eventually the D_5 signal contains only low-frequency content. One may notice that around the sample 100 (in Fig.5.23), D_1 signal presents a large variance due to the right-hand movement that will start. **This is fact the ERS in γ band prior to the movement onset for event-related potentials.** This can not be seen in the total signal s_1. Such a situation illustrates well the major interest of the time-frequency representation of wavelet transformations.

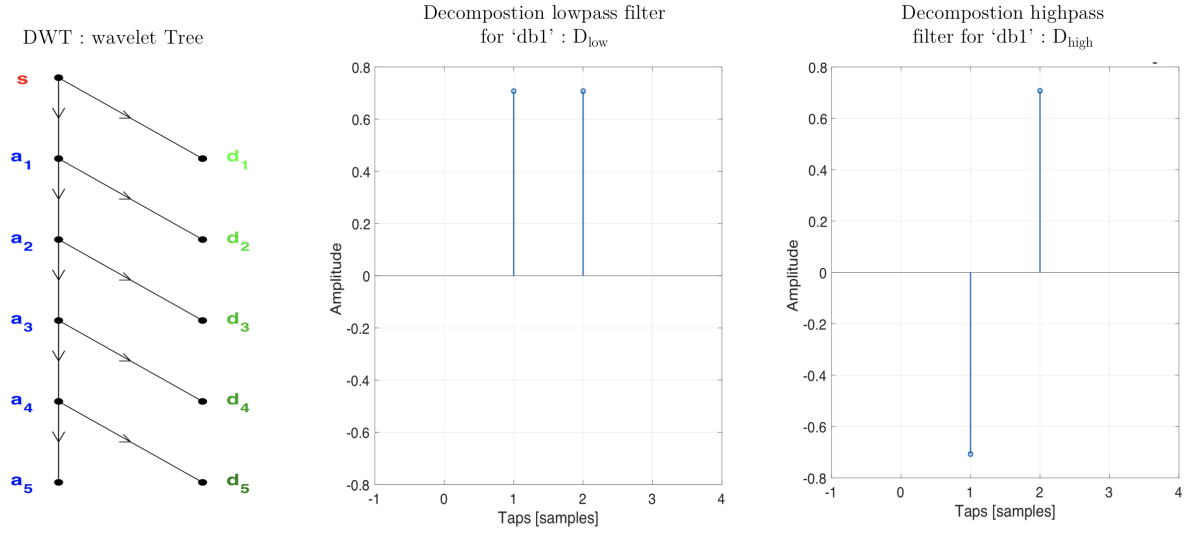


Figure 5.22: Structure of the discret wavelet transform : (1) wavelet tree, (2) D_{low} decomposition filter and (3) D_{high} decomposition filter. Both impulse responses contain 2 taps. This is a characteristic of 'db1' window.

Eventually one may present a 2D-projection of some entries (one can not plot a 100-dimensions vector) of the features vectors using the training database of 1500 trials established in Chapter 4. These features vectors are established using the whole processing chain described so far in this chapter. Fig.5.24 depicts two 2D projections (feature 91 vs feature 23/feature 81 vs feature 23). One may notice that the rest class is well separated from others. However the left-hand and right-hand classes overlap together. Notice also that entries of the features vectors are negatives given the log-transform. The features classification stage must now be trained in order to separate accurately these 3 classes.

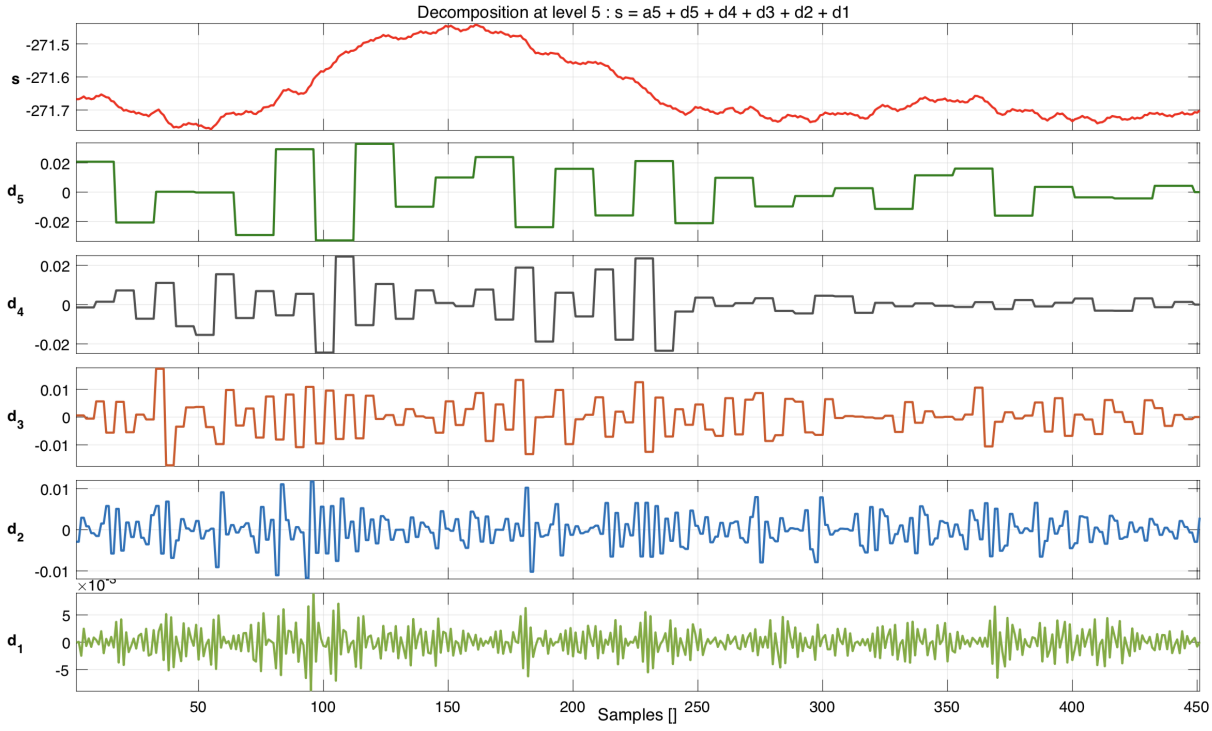


Figure 5.23: DWT of a given s_1 coming from of the spatial filter stage. The decomposition performed on 5 levels progressively 'peels off' high frequency content.

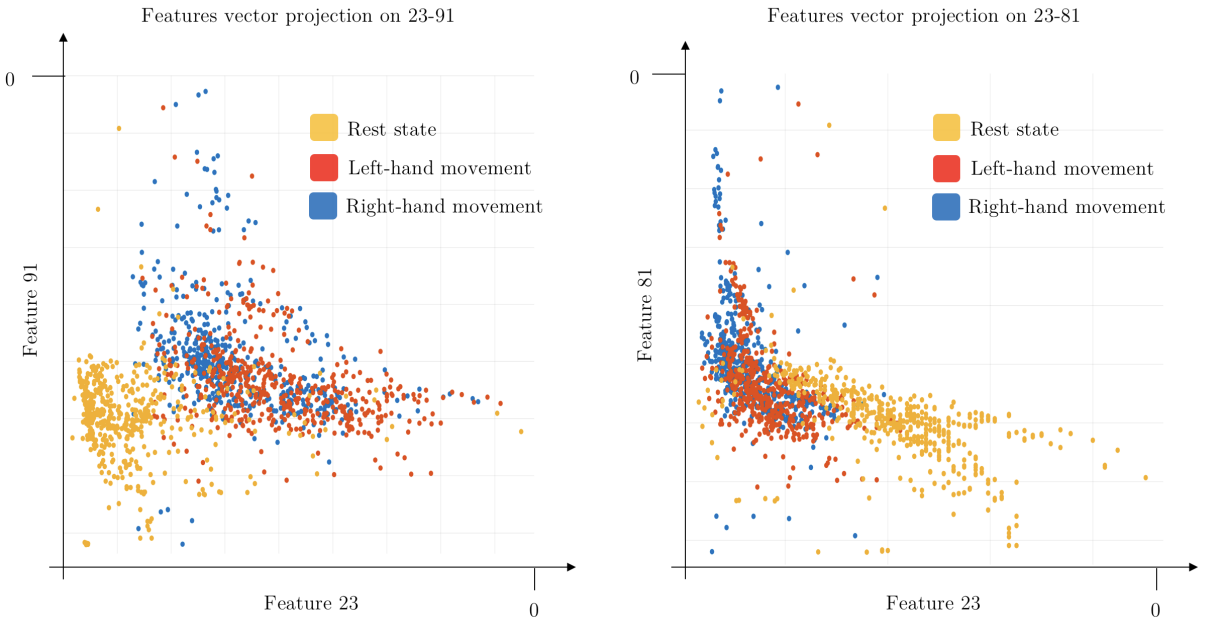


Figure 5.24: 2D projection of several entries (23, 81 and 91) of features vectors obtained with the whole processing chain described so far and using the training database of 1500 trials from Chapter 4.

5.4.5 Feature Translation and Classification

The last stage of the real-time processing chain is the features classification. One receives as input a features vector that contains discriminative/informative features and one must predict the class it belongs to. A training phase is realized before the real-time classification. By inspecting results obtained in previous chapter the more accurate classification algorithm that one may select among those presented is the multilayer perceptron neural network. This section describes the neural network implementation in `C/C++` on the FPGA. First, the MLP is explained more in details. This explanation follows the one from chapter 2. Next the training phase done via `MATLAB` is presented. Eventually, the software implementation is presented.

5.4.5.1 Multilayer perceptron

As reminder the multilayer percpetron is organized with 3 layers in this work.

1. The input layer made of as many nodes as the size of the input features vector. For this work, the input layer consists of 100 nodes. These input nodes are the only neurons that are not subject directly to an activation function.

2. Hidden layers are made of neurons that perform linear combinations of either previous neurons or input nodes. Next these neurons go trough a non-linear activation function. For this work, one hidden layer with 30 neurons is used and the activation function is the 'tansig' which is slightly different from the sigmoïd function and is depicted in Eq.5.17 and Fig.5.25.

3. The last layer of a neural network is called the softmax layer because its activation function is slightly different. It is basically a normalized exponential function that maps a n-dimensional vector of arbitrary input values into a n-dimensional vector where all entries are in the range of (0,1) and the sum of all entries is 1. For this BCI design, the output vector has a size of 3. Organization of this vector is depicted in figure 5.26.

The tansig activation function is given by :

$$a = tansig(n) = \frac{2}{1 + \exp(-2n)} - 1 \tag{5.17}$$

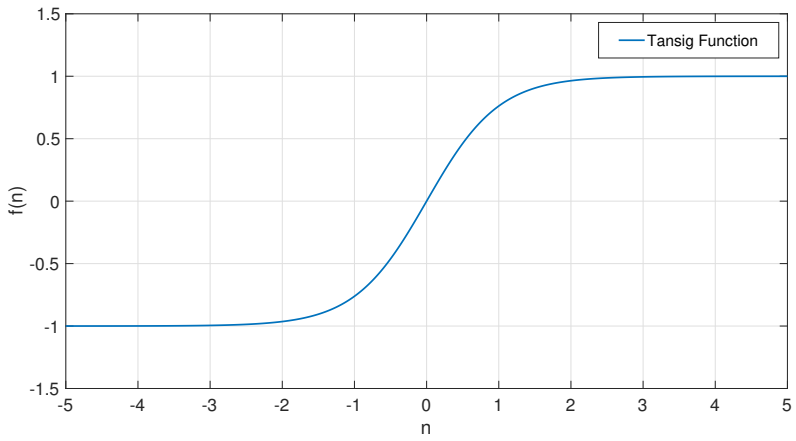


Figure 5.25: Tansig activation function is defined on $\mathbb{R}$ and has, unlike the sigmoïd function, a co-domain equal to (-1,1).

The interest of using a softmax layer as output layer holds in the possibility of expressing the output as a vector that contains the probability of each motor task, each task occupies one

entry of the output vector. The input-output relationship introduces thus a dimension reduction (from 100 to 3). Moreover the MLP neural network definitely acts as a translation stage. Indeed it takes, as input, a 100-dimension features vector that contains several informative features but that can not be understood in terms of motor tasks. It generates, as output, a vector that contains the probability of each of the 3 motor tasks.

The MLP used for the BCI design is depicted in figure 5.27. One may observe the input features vector of 100-dimensions that undergoes a progressive dimension reduction. First its size decreases trough the hidden layer organized in 30 neurons and eventually trough the softmax layer that generates a 3-dimensions vector.

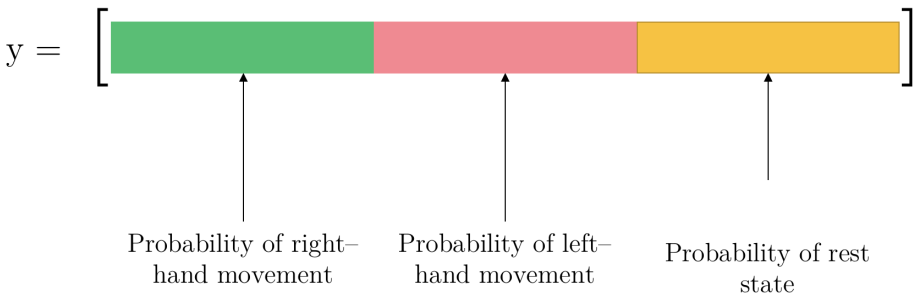


Figure 5.26: Organization of the output vector. Each entry is the probability of a given motor task among the 3 motor tasks that can be recognized in this BCI design (right-hand movement, left-hand movement and rest state).

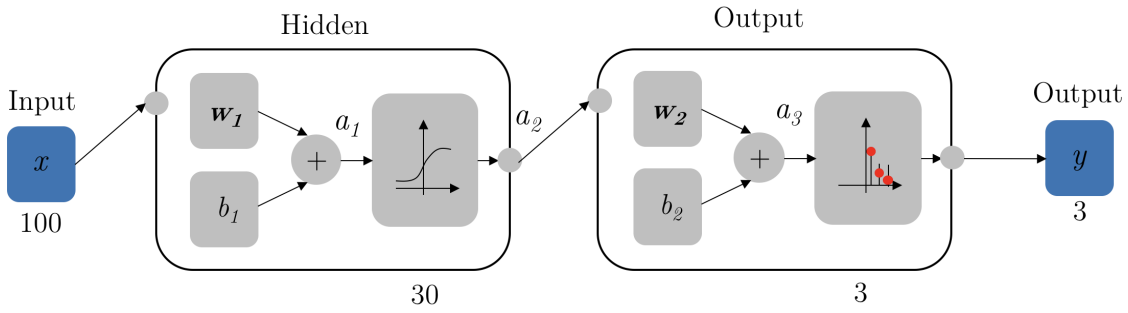


Figure 5.27: Organization of the MLP neural network used in this BCI design.

Mathematically speaking, one has a features vector $x \in \mathbb{R}^{100x1}$ that is combined with a weights matrix W_1 according to :

$$a_1 = W_1 \cdot x + b_1 \qquad \text{with } b_1 \in \mathbb{R}^{30x1} \text{ and } W_1 \in \mathbb{R}^{30x100} \tag{5.18}$$

The second step of the neural network consists in sending a_1 through the activation function that operates element by element the 30 entries of a_1 :

$$a_2 = tansig(a_1) = \frac{2}{1 + \exp(-2 \cdot a_1)} - 1 \qquad \text{with } a_2 \in \mathbb{R}^{30x1} \tag{5.19}$$

Afterwards on may proceed to the second combination with a second weight matrix W_2 according to :

$$a_3 = W_2 \cdot a_2 + b_2 \qquad \text{with } b_2 \in \mathbb{R}^{3x1} \text{ and } W_2 \in \mathbb{R}^{3x30} \tag{5.20}$$

a_3 is therefore a 3-dimensions vector that will go trough the softmax layer in order to be normalized :

$$y = softmax(a_3) = \frac{\exp(a_3 - \max(a_3))}{\sum_{i=1}^{3} \exp(a_{3,i} - \max(a_3))} \tag{5.21}$$

5.4.5.2 Training the MLP

The learning process that occurs in the neural network consists in modifying weights and bias values (W_1, W_2, b_1 and b_2) using data from the training set obtained in chapter 4. One must then evaluate the performance of the network using the test data set obtained in chapter 4 in the same conditions :

$$\{x_{train}\}_{n=1}^{N_1} \in \mathbb{R}^{100 \text{ x } 1} \qquad \{x_{test}\}_{n=1}^{N_2} \in \mathbb{R}^{100 \text{ x } 1} \tag{5.22}$$

As reminder the train data set is made of $N_1 = 1350$ trials, i.e. 450 trials for each motor task. The test set consists of $N_2 = 150$ trials, i.e. 50 trials for each motor task. These two sets form together the whole training data base obtained in chapter 4 ($N_1 + N_2 = N = 1500$) and decomposed here using a random 90%-10% partitioning. As explained in chapter 3, the learning phase is executed by minimizing a cost function. This cost function generally measures the distance between the MLP output and the expected output of the training data set. In this work, the MSE is chosen as cost function. The optimization problem becomes thus :

$$MSE(\{x_{train}\}_{n=1}^{N_1}) = \frac{1}{2N} \sum_{k=1}^{N_1} (y_k - y_{k,\text{train}})^2 \tag{5.23}$$

where y_k represents the output of the MLP and $y_{k,\text{train}}$ is the expected output. As reminder, an expected right-hand corresponds to $y_{k,\text{train}} = (1, 0, 0)$, an expected left-hand corresponds to $y_{k,\text{train}} = (0, 1, 0)$ and an expected rest state corresponds to $y_{k,\text{train}} = (0, 0, 1)$. By developing this equation, one may obtain :

$$MSE(\{x_{train}\}_{n=1}^{N_1}) = \frac{1}{2N} \sum_{k=1}^{N_1} \left\{ softmax(W_2 \cdot \frac{2}{1 + \exp(-2W_1 \cdot x_{k,train} + b_1)} - 1 + b_2) - y_{k,train} \right\}^2 \tag{5.24}$$

The total optimization problem can be expressed as :

$$\begin{aligned} \underset{W_1, b_1, W_2, b_2}{\text{minimize}} \quad & MSE(\{x_{train}\}_{n=1}^{N_1}) \\ \text{subject to} \quad & W_1 \in \mathbb{R}^{30\text{x}100}, b_1 \in \mathbb{R}^{30\text{x}1}, W_2 \in \mathbb{R}^{3\text{x}30}, b_2 \in \mathbb{R}^{3\text{x}1} \end{aligned} \tag{5.25}$$

This problem, highly nonlinear, can be solved with various techniques. Nevertheless, the most popular one is the line-search method based on [1]backpropagation and generally accompanied by a regularizer. The purpose of such an algorithm is to find W_1^*, W_2^*, b_1^* and b_2^* such that $\nabla MSE = 0$. In order to reach that point the algorithm must, in an iterative procedure, follows the direction of descent with a certain step size. After each iteration,∇MSE is re-computed and once a sufficient threshold value (e.g. 10^{-7}) is reached, the algorithm stops iterating. **Notice that the global minimum of such a function is not always reached given the non-convexity of the cost function**. Therefore, choosing relevant initial conditions is crucial for two reasons :

1. One can not be sure to converge towards a minimizer. This could be a saddle point.
2. One can not be sure to converge fast. Generally, the number of iterations is limited and thus having relevant initial conditions allows to go closer to the solution than with bad initial conditions.

[1]the authors have investigated several Newton-based methods such as BFGS algortihm in order to solve the problem. However this method was much less efficient than tools provided by `MATLAB`

Eventually, when the training process is completed, i.e. when W_1^*, W_2^*, b_1^* and b_2^* are known and minimize the MSE after a certain number of iteration, the algorithm is tested with test data set $\{x_{test}\}_{n=1}^{N_2}$. This test simply consists in sending the values $x_{k,test}$ in the MLP and evaluatin the predicted output. One will not go further in the details of the training process, which falls outside the scope.

One will first explain briefly the concept of *confusion matrix*. This is a 3-by-3 matrix (3 motor tasks here) that has two labels: *actual*, also known as target class and *predicted*, also known as output class. $C_{i,j}$ corresponds to the number of trials that have been recognized by the BCI offline system as a motor task of class i but that are in fact motor task of class j. This gives an insight into the error being made and, more importantly, the types of misclassifications. Fig.5.28 depicts three confusion matrices. The left one corresponds to the confusion matrix of the training subset N_1, made of 1350 trials. The confusion matrix at the center is related to the test data set N_2, made of 150 trials. Eventually the right one is the total confusion matrix, defined as the sum of the two previous matrices. By analyzing these three matrices, one may notice that the accuracy (as defined in chapter 4) is **100%** for the training subset. This makes sense because this training subset is used in the optimization problem, leading thus to an optimal classification. The accuracy of the test subset is **94%**. This is good news because it means that the classification is valid and that the MLP using W_1^*, W_2^*, b_1^* and b_2^* performs well on unseen data.

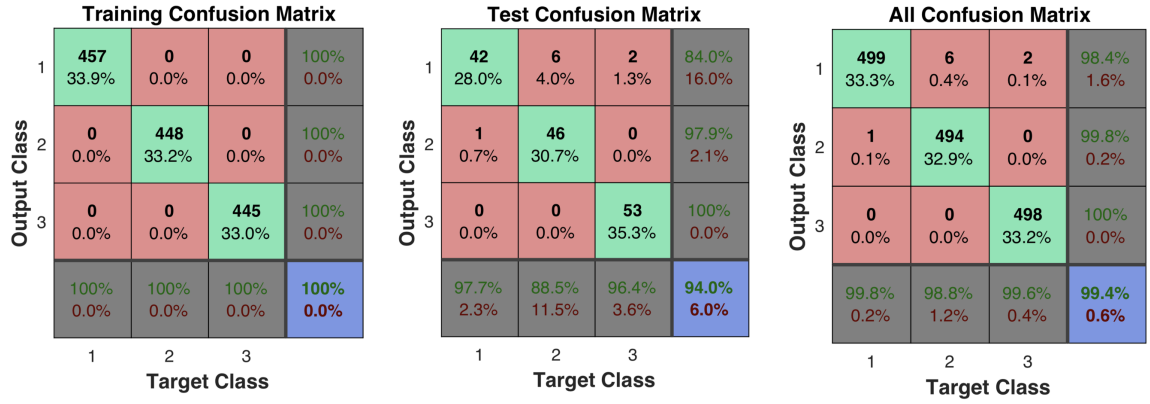


Figure 5.28: Confusion matrix obtained after the training process. The left matrix corresponds to the confusion matrix linked to the training data set ($N_1 = 90\%$ of N), the middle one is obtained with the test data set ($N_2 = 10\%$ of N) and the right one is the total confusion matrix of the whole training database of 1500 trials.

Fig.5.29 and 5.30 plot the evolution of $\nabla MSE(\{x_{train}\}_{n=1}^{N_1})$ and MSE itself in function of the number of iterations, respectively. We can notice that the optimization algorithm has converged after 33 iterations and leads to a gradient value of $1.7e^{-7}$ and and a training MSE value of $2.4e^{-10}$.

Fig.5.31 depicts the error histogram of the training process. It depicts 4500 instances (1500 trials leading to 1500 outputs 3-dimension vectors, leading to 4500 instances) of the error between the predicted output vector and the expected out put vector. For example, a right-hand movement could be predicted with $(0.8, 0.1, 0.1)$ as output vector and is expected with $(1, 0, 0)$ as output vector. This means that the error vector is $(0.2, -0.1, -0.1)$. These three instances would then be depicted on the error histogram. Regarding the histogram depicted in Fig.5.31, it is possible to observe that most of the instances are located in the range (-0.05;0.05) which means that **the MLP predicts accurately the class of a given motor task and this prediction is made with a high probability** leading to very small errors. In fact, 3194 instances exhibit an

error between -0.05 and 0 and 1278 instances exhibit an error between 0 and 0.05. This represent 99.38% of the total number of instances. There remains 28 instances that exhibit a larger error. Among those instances, 27 are related to a wrong classification. This means that when the neural network cannot predict the user intent with a probability larger than 95%, there is a huge chance of wrong classification.

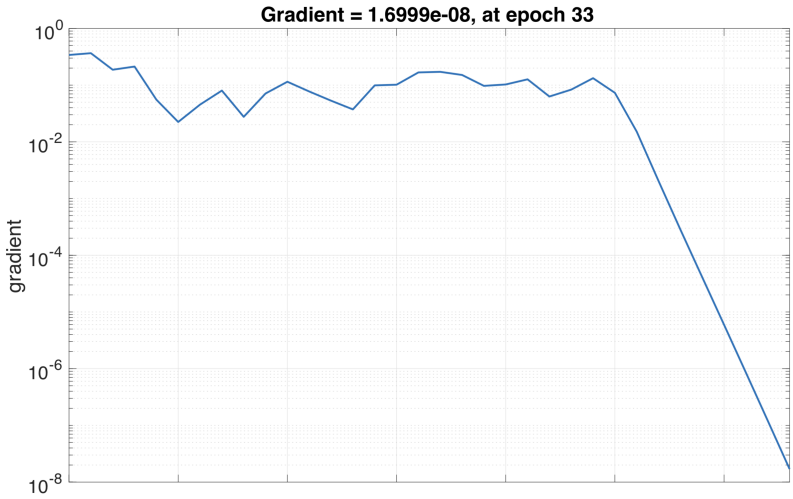


Figure 5.29: Evolution of the gradient of mean-squared error as a function of the number of iterations.

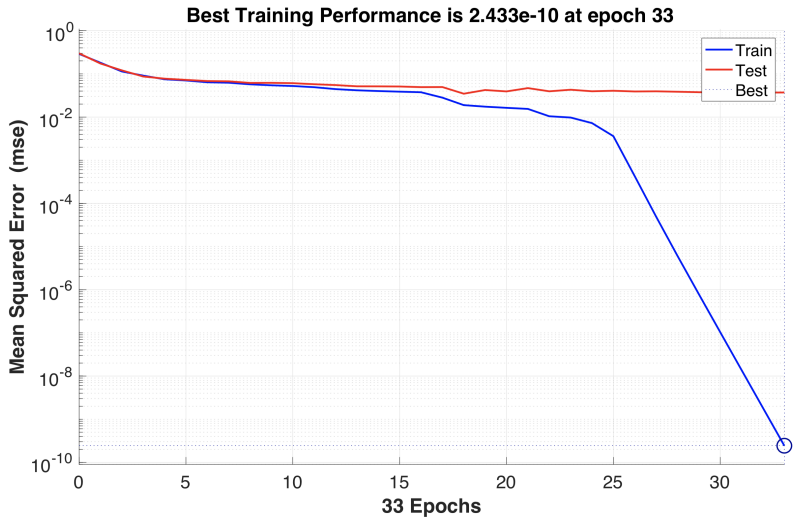


Figure 5.30: Evolution of the mean-squared error as a function of the number of iterations.

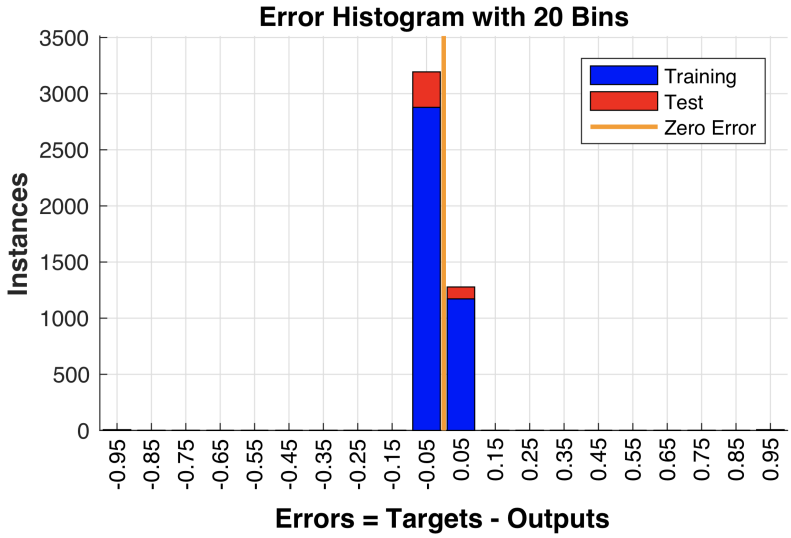


Figure 5.31: Evolution of the mean-squared error as a function of the number of iterations.

5.4.5.3 Software implementation

Once the MLP is trained and tested, one may load W_1^*, W_2^*, b_1^* and b_2^* under a matrix format in `C/C++` on the softcore in the FPGA. One may next program the MLP neural network in `C/C++` using these matrices. This is the purpose of this section. One will describe this implementation which is done using different modules corresponding to those of the block scheme depicted earlier in Fig.5.27. Two modules are implemented: 'tansig' and 'softmax'. This is depicted in algorithm 7.

Algorithm 7 MLP Neural Network - Library.

Require: /
  function y = tansig(x)
  $l \leftarrow$ length(x) //length() is a defined in the sense of `MATLAB`.
  $k \leftarrow 0$
  for $k < l$ **do**
    $y(k) = 2/(1 + \exp(-2 \cdot x(k))) - 1$
    $k \leftarrow k + 1$
  end for
  end function

  function y = softmax(x)
  $l \leftarrow$ length(x) //length() is a defined in the sense of `MATLAB`.
  $k \leftarrow 0$, $sum \leftarrow 0$
  $x_{max} \leftarrow$ max(x) //max() is a defined in the sense of `MATLAB`.
  for $k < l$ **do**
    $y(k) = \exp(x(k) - x_{max})$
    $sum \leftarrow sum + y(k)$
    $k \leftarrow k + 1$
  end for
  $y = y./sum$ //Elementwise division.
  end function

The second part of the implementation of the MLP neural network consists in using the aforementioned modules in order to implement the complete block scheme depicted in Fig.5.27.

Algorithm 8 MLP Neural Network - Implementation.

Require: W_1^*, W_2^*, b_1^*, b_2^* and x

$a_1 \leftarrow W_1 \cdot x + b_1$

$a_2 \leftarrow \text{tansig}(a_1)$

$a_3 \leftarrow W_2 \cdot a_2 + b_2$

$y \leftarrow \text{softmax}(a_3)$

The implementation is realized in algorithm 8. This completes the `C/C++` implementation of the MLP neural network and the whole processing chain. One will now discuss about the processing chain performances.

5.5 Performances

Eventually when the whole processing chain is implemented in `Verilog` and `C/C++` on the FPGA MAX1050 , one may test its performance. Let us distinguish two kinds of performances.

1. The real-time classification rate, which will be discussed in Chapter 5.
2. The system performance in terms of response time, memory management and dissipated power which is analyzed in this section.

5.5.1 Response time

One must distinguish the response time of `Verilog` part and `C/C++` part of the processing chain. The part programmed in `Verilog` will exhibit a better response time than the `C/C++` software part because the `Verilog` part implements frequency filters only. Two delays are denoted when filtering a signal :

1. The delay inherent to the filter. It depends on the sampling period and the order of the filter. This delay must be taken into account.
2. The delay produced by the filtering operation itself that requires several floating-points operations. Given that the filtering operation are executed in parallel with respect to the EEG signals acquisition (filters do not wait for the signal to be fully acquired to start the filtering operation. Indeed each new acquired sample is immediately filtered by both the bandpass filter and the Notch filter), this delay is negligible (order of [μs]).

As reminder, the BCI is **synchronous** and has a working principle in 2 steps as explained in Fig.5.32. One will consider the delay d_{tot} introduced by these filters (bandpass filter and Notch filter).

$$d_{\text{tot}} = d_{\text{FIR}} + d_{\text{Notch}} \tag{5.26}$$

The delay introduced by FIR bandpass filter is constant for all frequency bins given the linear phase of the filter. Regarding the phase of Notch filter (See Fig.5.12), it is close to zero below 30 [Hz] and then increases. The phase is not linear and therefore the delay depends on the frequency. One will assume, however, that for EEG signals of interest (below 30 [Hz]) the phase is 0, and so is the delay. Eq.5.26 can be rewritten as :

$$d_{\text{tot}} \approx \frac{M-1}{2}\frac{1}{F_s} = \frac{17-1}{2}\frac{1}{125} = 64 \text{ [ms]} \tag{5.27}$$

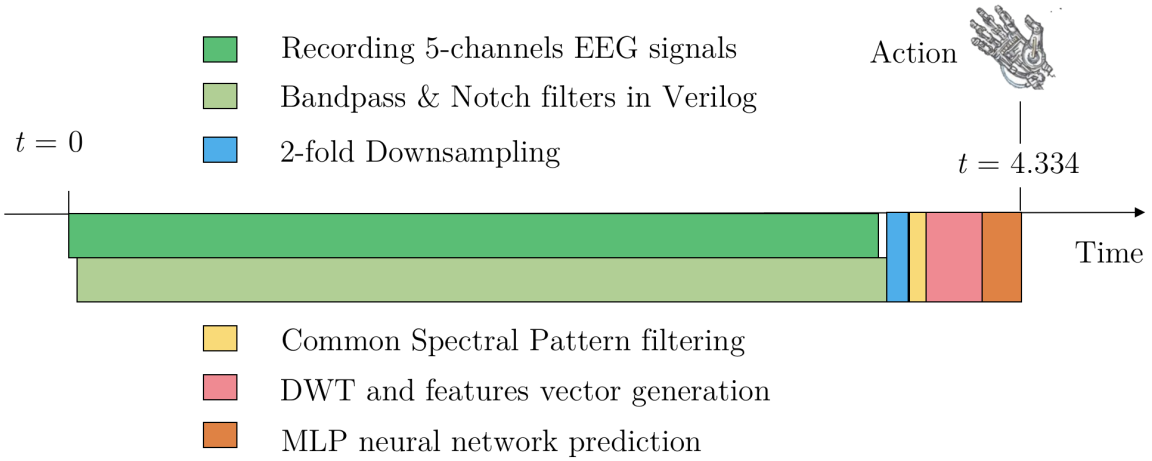


Figure 5.32: Timing sequence of the real-time BCI system

One may also observe in Fig.5.32 that `C/C++` software is executed in a sequential manner. This makes sense because each block needs the result from the previous one in order to be executed. On has measured the different execution times and the total execution time is equal to 334 [ms].

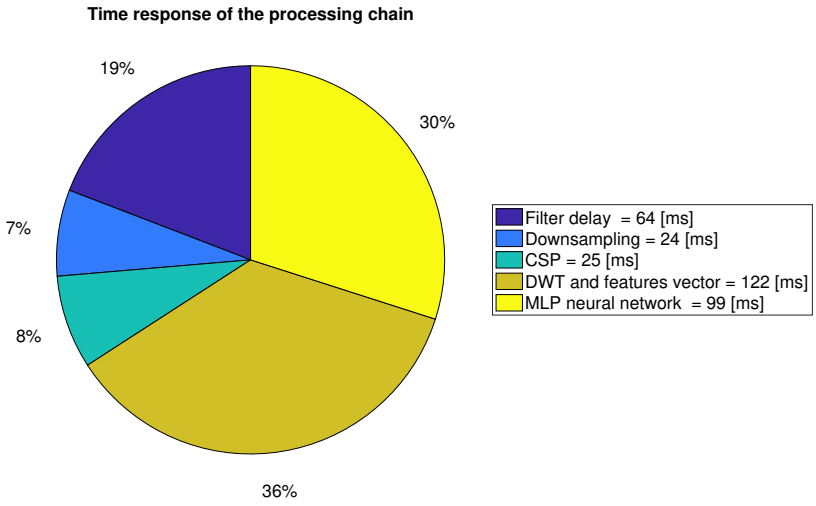


Figure 5.33: Timing performance of the real-time BCI system

One may conclude that timing constraints defined in chapter 4 are respected because the total execution time required in order for the BCI to predict the user's intent is less than 400 [ms].

5.5.1.1 Memory management

Regarding the memory management, one need to distinguish again the `Verilog` part and the software part. The first part corresponds to the size partitioning in terms of logic elements (LEs) while the second part corresponds to the use of the on-chip memory of the softcore NIOS II/f.

5.5.1.1.1 Logic elements.

The module `ADS_SPI_COM` occupies only 619 LEs which (1.2 % of the total size). This is good news given this module is required for interfacing the ADCs and allows the communication with

the FPGA.

Regarding the bandpass filters `my_BPF`, the implementation is quite cumbersome and takes 19349 LEs (38.7 % of the total size) for each of the 5 bandpass filters.

Regarding the Notch filters `my_NOTCH`, the implementation is lighter than `my_BPF`. It takes 5123 LEs (10.2 % of the total size) for each of the 5 Notch filters.

Eventually, the NIOS II/f takes 20931 LEs (41.8 % of the total size). This implementation takes many more LEs than the NIOS II/e implemented in previous work [3] [4]. This is related to the numerous features included in this softcore (hardware multiplication/division/rotation, cache memory management,...).

The total number of used LEs is 46022 for this BCI design on the FPGA MAX1050. Therefore 92.04 % of the total amount of logic elements is used. One can not afford a much more complex processing chain or using an additional electrode with this [2]hardware. **This justifies the choice made in chapter 3 about the number of electrodes. This justifies also the need for implementing the downsampling operation in software**. Fig.5.34 depicts the size partitioning.

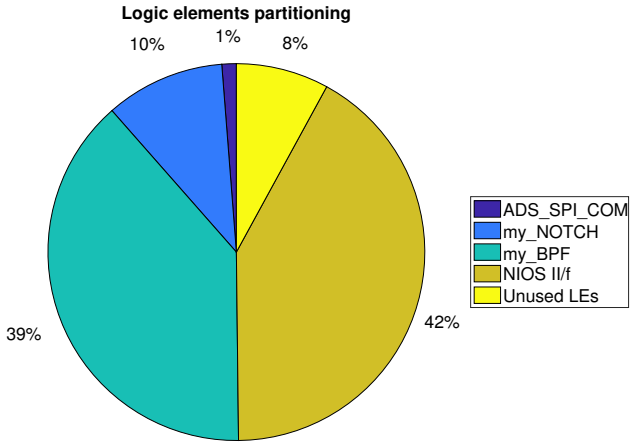


Figure 5.34: Size partitioning of the total 50k logic elements.

Notice also that 95% of the memory bits are used with this BCI design (1,496,721 memory bits used) and that 50% of the embedded 9-bits multipliers are used too.

5.5.1.2 On-chip RAM memory.

The total flash memory available within the MAX1050 corresponds to 120 [kB]. This memory is partitioned into several usage : The data cache, the instruction cache, the flash memory used for downloading the `C/C++` code and the RAM memory required for executing the code. The memory required for executing the code corresponds in fact to the memory required for storing 4 seconds of EEG signals over 5 channels with a sampling rate of 250 [Hz]. The partition is given by table 5.4.

[2]The authors have tried to implement additional modules such as the 2-fold downsampling but the Quartus tool was not able to fit the design on the FPGA.

Type	Usage	Partitioning
Instruction cache	4 [kB]	3.33 % of total on-chip memory
Data cache	4 [kB]	3.33 % of total on-chip memory
C/C++ code size	86 [kB]	71.66 % of total on-chip memory
RAM for acquisition	20 [kB]	16.66 % of total on-chip memory
Unused	6 [kB]	5 % of total on-chip memory

Table 5.4: On-chip RAM memory usage.

5.5.2 Power management

By running the `PowerPlay Analyzer Tool` from `Quartus`, one may estimate the power dissipated by the FPGA and external units. One may also estimate the current that flows trough the FPGA. Table 5.5 gives an overview of these estimations. One may notice that the BCI board has been designed with a power supply of 5 [V] and 500 [mA] which is thus sufficient.

Total thermal power dissipation	2598.55 [mW]
Core dynamic thermal power dissipation	509.03 [mW]
Core static thermal power dissipation	424.43 [mW]
I/O thermal power dissipation	1665.09 [mW]
VCC total current drawn	340.03 [mA]
Dynamic total current drawn	169.75 [mA]
Static total current drawn	170.28[mA]
Minimum power supply	340.03 [mA]]

Table 5.5: Power management of the BCI board.

5.6 Conclusion

This chapter aims at programming the whole processing chain both in `Verilog` and `C/C++` in order to have a functional real-time BCI. The BCI respects thus all the specifications defined in previous chapters, especially the complete sequence of algorithms proposed by the `MATLAB` tool in chapter 4 and the timing constraints defined within the same chapter. The two first stages of the processing chain are implemented in `Verilog` while the remaining part is implemented in `C/C++`. The communication between `Verilog` parts and `C/C++` parts is ensured trough the softcore, the same softcore that executes the `C/C++` codes and communicate the prediction to an host computer used for debugging. One will test the performance of the real-time system in the next chapter and discuss the results.

Chapter 6

Results and discussion

In order to design a BCI, one must established its specification first. The second step consists in building the hardware. Next comes the BCI software programming such that it might work in real-time. Eventually one needs to evaluate perfomance of the BCI system in terms of classification rate. This chapter aims at analyzing classification rate of the real-time BCI system for the three actions that it must be able to identify, as defined in the specifications. At this point let us recall briefly the specifications :

1. BCI general specifications : The way the BCI has to be used. This involves a discussion about the type of BCI that one must design : Invasive/non-invasive, synchronous/asynchronous, real-time/offline, the type of tasks that the BCI must recognize and other elements. This is extensively discussed in chapter 1 and 4.

2. BCI hardware specifications : This concerns the material used for the BCI design (BCI board from [3], new *Ultra Cortex Mark IV* helmet, acquisition over five channels). This is extensively discussed in chapter 3.

2. BCI software specifications : This concerns the way the BCI has to be programmed (CSP spatial filter, discrete wavelet transform, MLP neural network.). An exhaustive comparison of algorithms is done via the `MATLAB` tool. This is a software concern. This is extensively discussed in chapter 4.

4. BCI usage and protocol specifications : the way the real-time software learns to discriminate different motor tasks. One of the major difficulty of BCI design lies in this step and the fact that brain signals are highly time-varying. The training protocol is extensively discussed in chapter 4 and has a major impact in this chapter.

On top of the results analysis, this chapter will also discuss these results and will try to identify which factors may influence classification results. This chapter is organized as follows : First, metrics used for evaluating real-time performance are presented in section 6.1. Next, results in ideal conditions (defined later) are presented in section 6.2. Then, the different factors that influence the results are studied in section 6.3.

6.1 Performance Metrics

In order for the BCI real-time system to be assessed, one must define performance metrics. One will use several metrics among those available : the confusion matrix, the accuracy, the precision, the sensitivity, the specificity, the F1 score and the receiver operating characteristics curve (ROC curve). Notice that as the classification realized in this work is not binary but multiclass, one will use the **One-versus-All** classification, which consists in isolating one class among all and making a group with all other classes. Then one can consider that the group (made by all

but one class) and the isolated class form a binary classification problem. More details about performance metrics can be found in [74] [75].

6.1.1 Confusion matrix.

This corresponds to a table made of C^2 entries where C is the number of classes that the BCI system must recognize. In this work $C = 3$ given that three motor tasks have to be identified by the BCI (right-hand movement, left-hand movement and rest state.). Class 1, labeled $C1$ is assigned to right-hand movement, class 2, labeled $C2$ is assigned to left-hand movement and eventually class 3, labeled $C3$ is assigned to rest state. This matrix has two labels : **Actual**, also known as Target class and **Predicted**, also known as Output class. The organization of a confusion matrix is depicted in Fig.6.1

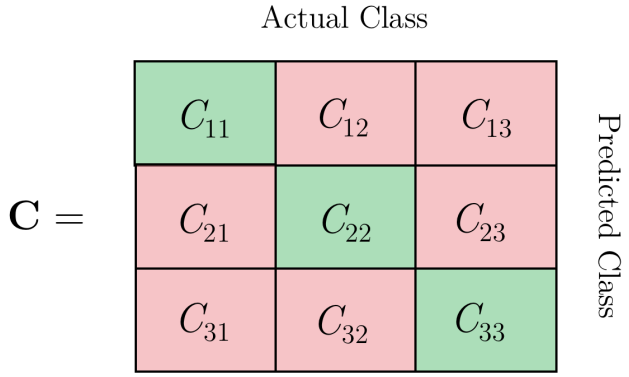


Figure 6.1: Organization of a confusion matrix. Each entry C_{ij} corresponds to the number of trials classified in class i which actually belongs to class j.

The C_{ij} corresponds to the number of trials that have been recognized by the BCI system as a motor task of class i but that are in fact motor tasks of class j. This gives an insight into the error being made and, more importantly, the types of misclassifcations. One may define several terms that will be useful for other metrics :

1. **T1**, this is a true class 1 (right-hand movement), which means that the BCI predicts class 1 and the actual class is 1 too. This corresponds to elements in C_{11}.
2. **T2**, this is a true class 2 (left-hand movement), which means that the BCI predicts class 2 and the actual class is 2 too. This corresponds to elements in C_{22}.
3. **T3**, this is a true class 3 (rest state), which means that the BCI predicts class 3 and the actual class is 3 too. This corresponds to elements in C_{33}.
4. **F1**, this is a false class 1, which means that the BCI predicts class 1 and the actual class is either 2 or 3. This corresponds to elements in C_{12} and C_{13}.
5. **F2**, this is a false class 2, which means that the BCI predicts class 2 and the actual class is either 1 or 3. This corresponds to elements in C_{21} and C_{23}.
5. **F3**, this is a false class 3, which means that the BCI predicts class 3 and the actual class is either 1 or 2. This corresponds to elements in C_{31} and C_{32}.

6.1.2 Accuracy

This metric is defined based on the confusion matrix. This corresponds to the number of correct predictions made by the BCI system over all predictions made. In other words, this is defined as

the sum of diagonal elements of the confusion matrix divided by the sum of all entries :

$$\text{Accuracy} = \frac{\#\text{ Correct classifications}}{\#\text{ Classifications}} = \frac{\text{T1} + \text{T2} + \text{T3}}{\text{T1} + \text{T2} + \text{T3} + \text{F1} + \text{F2} + \text{F3}} \tag{6.1}$$

In order to make a coherent use of the accuracy metric, the number of target elements per class must be nearly balanced, i.e. almost same proportion of right-hand movements, left-hand movements and rest states during the assessment.

6.1.3 Precision

This metric is defined based on the confusion matrix too, but is related to a particular class. It means that there is a precision measurement for each of the three classes. This measure tells us what proportion of motor tasks of one given type is correctly predicted among all predictions for this particular motor task. This is expressed as :

$$\text{P}_{C1} = \frac{\text{T1}}{\text{T1} + \text{F1}} \qquad \text{P}_{C2} = \frac{\text{T2}}{\text{T2} + \text{F2}} \qquad \text{P}_{C3} = \frac{\text{T3}}{\text{T3} + \text{F3}} \tag{6.2}$$

6.1.4 Sensitivity

This metric is defined based on the confusion matrix too and is related to a particular class too. It means that there is a sensitivity measure for each of the three classes. This measure tells us what proportion of actual motor tasks of one given type is correctly predicted among all actual motor tasks of this type. This is expressed as :

$$\text{S}_{C1} = \frac{\text{T1}}{\text{T1} + C_{31} + C_{21}} \qquad \text{S}_{C2} = \frac{\text{T2}}{\text{T2} + C_{12} + C_{32}} \qquad \text{S}_{C3} = \frac{\text{T3}}{\text{T3} + C_{13} + C_{23}} \tag{6.3}$$

6.1.5 Specificity

This metric is defined based on the confusion matrix too and is related to a particular class too. It means that there is a specificity measure for each of the three classes. The measure of specificity for class 1 (e.g.) tells us what proportion of actual motor tasks 2 or 3 are correctly predicted among all actual motor tasks 2 and 3. This is expressed as :

$$\begin{aligned} \text{Sp}_{C1} &= \frac{\text{T2} + \text{T3}}{\text{T2} + \text{T3} + \text{F1} + C_{32} + C_{23}} \\ \text{Sp}_{C2} &= \frac{\text{T1} + \text{T3}}{\text{T1} + \text{T3} + \text{F2} + C_{31} + C_{13}} \\ \text{Sp}_{C3} &= \frac{\text{T1} + \text{T2}}{\text{T1} + \text{T2} + 31 + C_{12} + C_{21}} \end{aligned} \tag{6.4}$$

6.1.6 F1 score

The purpose of this metric is to merge information about both the precision and the sensitivity into a single metric. This metric is defined for a given class too. It is computed as the **harmonic mean** of the precision and the accuracy :

$$\text{F1}_{Ck} = \frac{2}{1/\text{S}_{Ck} + 1/\text{P}_{Ck}} \tag{6.5}$$

6.1.7 Receiver Operating Characteristics

This is a plot that shows the prediction ability of the BCI system when the discrimination threshold varies. This plot has, as y-axis the true positive rate (TPR) which is equivalent to the sensitivity and as x-axis the false positive rate (FPR) which is equivalent to the 1 - specificity. A typical plot of ROC is depicted in Fig.6.2.

One may notice a baseline between (0,0) and (1,1). A classifier in contact with this baseline is equivalent to a random classification. Let's imagine that this ROC is established for class 1. At (0,0) the classifier always predicts either class 2 or 3, It has thus a TPR of 0 %. at (1,1) the classifier always predicts class 1 in any situation. In practice it is impossible to obtain a classifier below this baseline.

The better a classifier is, the closer it is to the (0,1) point which corresponds to 100% classification rate given that the TPR is 1 (predictions for this class is perfect) and the FPR is 0 (no wrong prediction for this class).

A last property must be emphasized : the area under the curve (AUC) which is a good measure for performance of a classifier. This corresponds in fact to the precision of a given class of a classifier. In Fig.6.2 one may notice that AUC = 90% which means a precision of 90% for this class.

A last question arises : How to construct such a plot based on measured data. First notice that one will produce three plots, one for each class using the **One-versus-All** approach. Such an approach allows to treat the classification as a binary classification. Next one will establish a table that computes the TPR and FPR for different threshold values depending on the output y of the MLP. These threshold values goes from 0% to 100%. **Notice that when the threshold value is chosen to be 50%, the TPR and FPR corresponds to the values found in the confusion matrix**. More details in [76] [77].

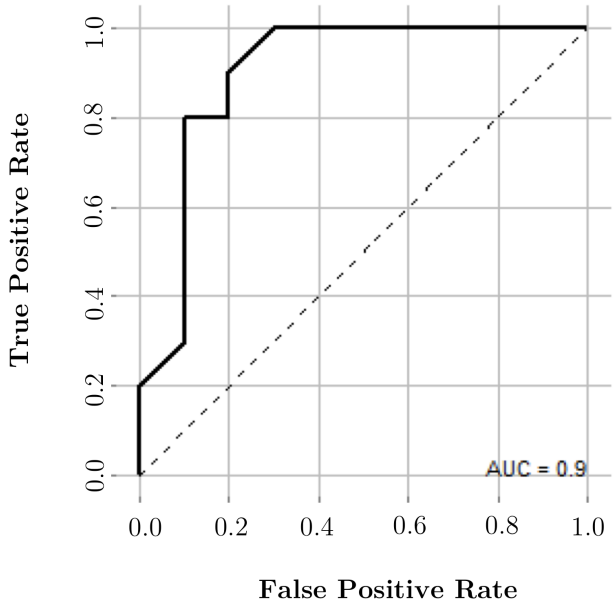


Figure 6.2: Typical classification Receiver Operating Characteristics plot.

To summarize, numerous manners exist for assessing the efficiency of a classifier. One will use several of them for performing the evaluation of the real-time BCI system.

Sampling frequency F_s	125 [Hz] (2-fold downsampling)
Frequency Filtering 1	16-taps FIR Bandpass filter in the μ band
Frequency Filtering 2	2-taps IIR Notch filter for 50 [Hz]
Spatial Filtering	5-channels Common SPectral Pattern
Feature Extraction	5 levels DWT with 4 quantities extracted : variance, maximum, minimum and Willison amplitude
Classification	MLP Neural network with 1 hidden layer of 30 neurons

Table 6.1: Summary of the software specifications.

6.2 Results and protocol

in order for the BCI to be assessed, one must execute the real-time software on the BCI board and perform a real-time assessment. One will evaluate only one software combination in this work. This corresponds to the software specified in chapter 4 and programmed in chapter 5. As reminder, the software specifications are summarized in Table 6.1. One will establish a protocol for assessing the BCI system. This assessment is realized with two people : the subject and the assessor. The subject is place in the same conditions as the ones used for establishing the training data set in chapter 4. The assessor is next to the subject. The protocol is defined as :

1. A sequence of 300 movements is defined by the assessor. **This sequence is arranged in the same manner as the training protocol** defined in chapter 4. The BCI system is turned on.

2. The BCI system is predicting movements. **The subject can not see the prediction made by the BCI system**.

3. The assessor writes down the predictions given by the BCI board. He must write the total output vector y in order to draw the ROC curve afterwards.

3. Steps 2 and 3 are repeated until the subject has executed 300 movements. 100 for each motor task.

4. Performance may be computed.

Here ideal conditions are used. This means that the BCI board is assessed with these 4 requirements :

1. The test user is the same person as the one used for establishing the training database.

2. The assessment is realized immediately after the record of the training database.

3. The user behaves in the same way as for establishing the training database (similar sequence of movements). This means that the user must perform two times the same movement in series in case of training protocol 2. In case of protocol 1, the user must execute 10 times the same movement in series as in the training protocol 1. This requirement allows to have an equivalent movement sequence between the testing phase and training phase.

4. The environment where the assessment is realized is the same as the one for recording the training database.

Assessment results (mean over 5 assessments) in these conditions are given in Fig.6.3 and Fig.6.4.

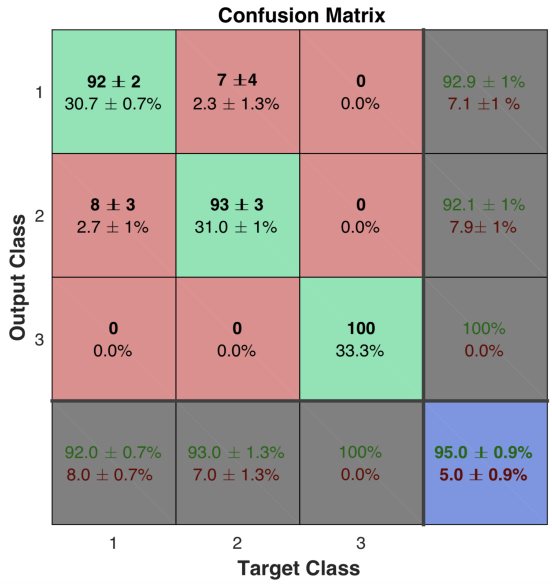


Figure 6.3: Confusion matrix for 5 assessments of the BCI real-time system.

This matrix has to be interpreted as follows : the mean result of each entry, over the 5 assessments, is depicted in the matrix. The standard deviation is added to the mean value. Each entry of the confusion is expressed as : mean ± standard deviation. Target class corresponds to actual class and output class corresponds to predicted class.

These results are characterized by an **accuracy of 95 %**. The precision of each class is respectively $P_{C1} = \mathbf{92.9\%}$, $P_{C2} = \mathbf{92.1\%}$ and $P_{C3} = \mathbf{100\%}$. Regarding the sensitivity of each class, one has respectively $S_{C1} = \mathbf{92\%}$, $S_{C2} = \mathbf{93\%}$ and $S_{C3} = \mathbf{100\%}$. Eventually, one may compute the F1 score (even if it is not really meaningful in this case given precision results and sensitivity results are close). One has respectively $F1_{C1} = \mathbf{92.447\%}$, $F1_{C2} = \mathbf{92.547\%}$ and $F1_{C3} = \mathbf{100\%}$.

On may analyze these results into several directions. First one may notice that no over-fitting problem seems to occurs. Indeed the test confusion matrix is rather close from the one obtained during the training process. **Results are however slightly better during the training phase but this makes sense because training data always produces better results**. Indeed results obtained in chapter 5 depict an accuracy of 99.4%, which is better than those obtained in real-time but nothing catastrophic neither. **The BCI real-time system works as expected.**

Next let's remark that the precision and the sensitivity of class 3 (rest state) is better than for other classes. **This basically means that the BCI real-time system can very accurately predict the presence of a movement**. However, it commits several errors for interpreting the movement.

By comparing precision and sensitivity one may see that these values are close, which means that the BCI real-system does not over-fit one movement with respect to another which is good news. During early development in this work, the authors have experimented such a situation where the BCI was recognizing only one category of movements. This means that for example class 1 was predicted when either a right-hand or left-hand movement was executed. Hopefully this issue has been solved.

Eventually let's recall that the confusion matrix has been built using 5 successive assessments in order to emphasize results variations. One may notice that these variations are rather weak, leading to standard deviation of about 1%. Therefore one may conclude that these results are rather reliable.

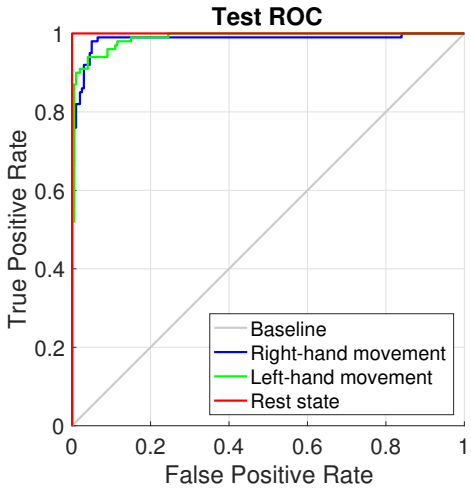


Figure 6.4: ROC plot for 5 assessment of the BCI real-time system.

Regarding the ROC plot one may notice that, once again, the classification of class 3 (rest state) is excellent. For class 1 and 2, ROC plots are close to (0,1) point. The classification is thus accurate too. One may notice that green curve in Fig.6.4 crosses the blue one around 5% of false positive rate (FPR). The green curve has a higher true positive rate than the blue curve between 0 and 5 % of FPR. Beyond 5% of FPR, the blue curve exhibits better TPR. This means that the BCI real-time system is more sensitive to left-hand movement for low FPR but this trend seems to reverse for a larger FPR, where the BCI real-time system is more sensitive to right-hand movement.

To summarize this section, the BCI real-time system produces good results with an accuracy around 95%, a good sensitivity and precision for each motor task. **However these results are obtained for specific conditions (4 requirements) described earlier and called ideal conditions.** One will remove progressively these ideal conditions in the next section in order to study their influence on the classification results.

6.3 Results in non-ideal conditions

In order to study the robustness of the real-time BCI system, one will remove each ideal conditions requirement. One will first remove the assumption that the user is the same person as the one who established the training database. Next one will remove the assumption that the assessment is performed with a training database established immediately prior. Then, the influence of the training protocol is going to be studied and eventually, the effect of the external environment is studied too. Notice that only one ideal condition is removed at a time.

6.3.1 Variation of the BCI user

For studying the influence of this parameter, one will assess the BCI system with two users, none of them has established the training database. All other ideal conditions requirements are preserved. The assessment procedure is the same as the one described earlier. The only difference lies in the fact that the BCI real-time system is not trained for these users. Results

are presented in Fig.6.5 and Fig.6.6.

One may analyze these results. The first element to notice is the important loss in accuracy. The other user 1 is characterized by an **accuracy of 79.3 %** while the other user 2 is characterized by an **accuracy of 78.7 %**. In both cases the results are about 15% worse that those obtained in ideal conditions. This emphasizes the dependency of the classification algorithm to a particular training data set and poor generalization behavior for multiple users.

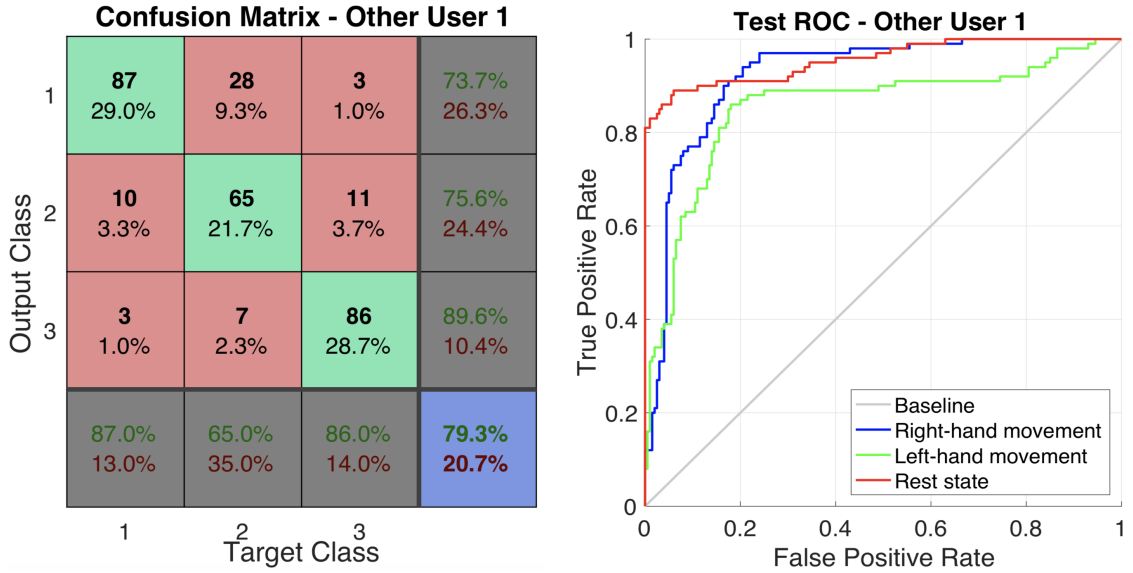


Figure 6.5: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system by a user who has not trained the system. Case 1.

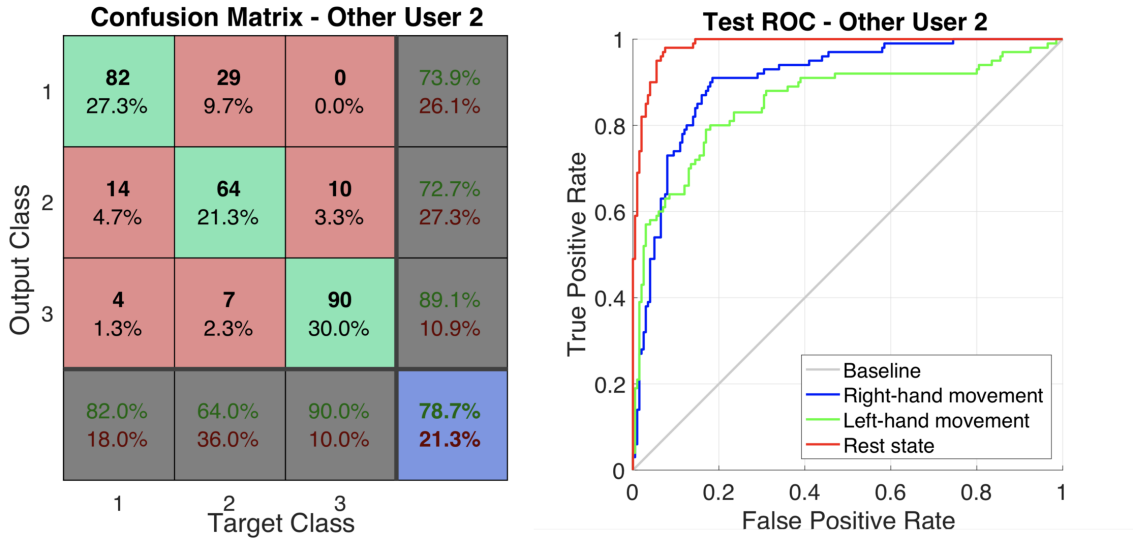


Figure 6.6: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system by a user who has not trained the system. Case 2.

Several reasons can justify this difference :

1. Brain signals patterns are similar for all human beings. However some slight differences

may appear depending on the brain anatomy of each human being. These differences are however very difficult to analyze and quantify.

2. The way the movement is executed can slightly vary from one person to another even if each movement is shown and taught to each subject.

3. Head morphological differences between the different users are probably a huge source of variations. Indeed electrodes are perhaps in contact with slightly different head areas. This is in fact the biggest drawback of the helmet given it cannot ensures an exact localization of the electrodes.

One may also notice that the precision and sensitivity of class 3 (rest state) and for each other user is better than precision and sensitivity of other classes. One again this means the BCI real-time system is able to identify the presence of a movement quite accurately but exhibits difficulties to recognize its nature (right-hand or left-hand movement).

One may remark that for both other users, the left-hand movement has always a lower sensitivity. The sensitivity of each class for other user 1 is respectively $S_{C1} = \mathbf{87\%}$, $S_{C2} = \mathbf{65\%}$ and $S_{C3} = \mathbf{86\%}$ while the sensitivity of each class for other user 2 is respectively $S_{C1} = \mathbf{82\%}$, $S_{C2} = \mathbf{64\%}$ and $S_{C3} = \mathbf{90\%}$. A bias is introduced when switching the user leading to over-fitting problems for the right-hand movement.

Eventually one remarks, by analyzing the ROC plot for other user 1, that the sensitivity of class 3 (rest state) is lower than the sensitivity of class 1 (right-hand movement) between 18% and 55% of FPR. Therefore actual right-movement are more precisely predicted than actual rest state within this part of the plot for other user 1.

To summarize, training the BCI real-time system with one subject and testing it with another leads to bad classification results. Three possible reasons have been mentioned in this paragraph : The brain signals, the movements execution manner and the head morphology. One will now study performance degradation related to the time laps between the assessment and the training database establishment.

6.3.2 Training database 'age' variation

For studying the influence of this parameter, one will assess the performance of the BCI real-time system 2 weeks after the establishment of the training database and compare with the results obtained in ideal conditions. All other ideal conditions requirements are preserved. The assessment procedure is the same as the one described earlier. The only difference lies in the fact that the BCI real-time system is assessed with a trained model that is 2 weeks old. Results are presented in Fig.6.7.

One may analyze these results. The first element to notice is the loss in accuracy of about 9%. This shows the importance of having an updated model of the BCI system for obtaining good results. In this work the training process of the classifier is realized offline. Such a training phase is easier to implement but exhibits a major problem : the training process is out-of-date after a certain period and needs to be retrained.

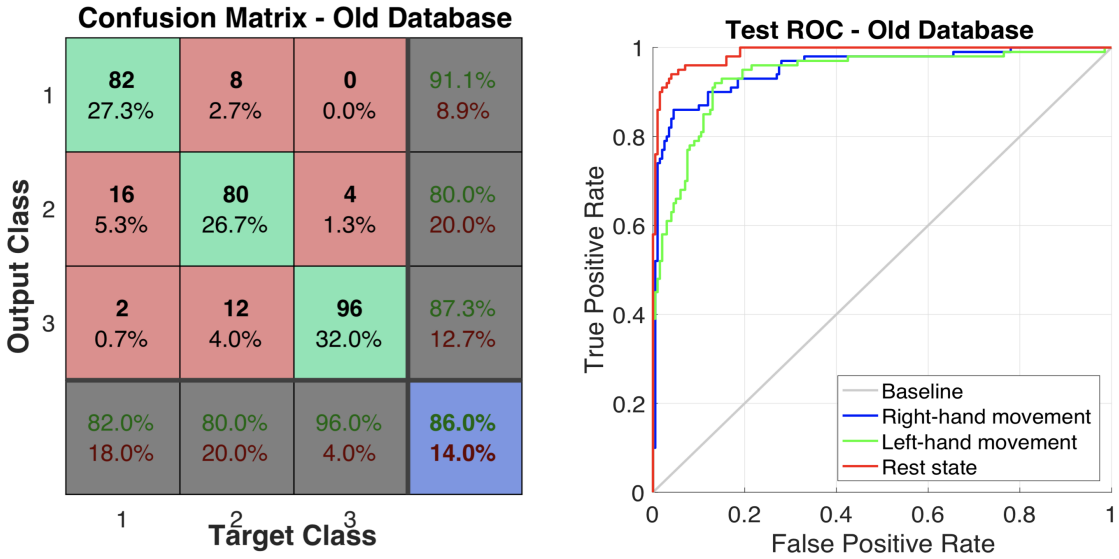


Figure 6.7: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system when a 2-weekds old database is used.

Several reasons can justify such differences :

1. The user's state, both physiological and psychological, evolves with time leading to very different conditions when the training database is 'old' compared to moment when the test is performed.

2. Some physical properties have an influence too. For example in this case, the haircut length has changed between the moment when the training database is established and the moment when the assessment is done.

3. Long-terms habituation effects, called here habit effects can modify the results too. In this case, the user that established the training database was assessing intensively the BCI system during this period. This means that he was used to operate the system. For the assessment described in this section, the same user had not used the BCI system since 2 weeks. He was not used to operate it anymore.

To summarize, the age of the training database has an impact on the classification accuracy. The older the database is, the worse the classification accuracy will be. Three possible reasons have been mentioned in this paragraph : The user's state, the user's physical changes (haircut length) and the usage rate of the BCI system. One will now study performance degradation related to the training protocol.

6.3.3 Training database protocol variation

For studying the influence of this parameter, one will perform two assessments. First one will assess the performance of the BCI real-time system that uses a training protocol of type 1 or 3 (as described in chapter 4) and tested with these protocols. In a second time, one will assess performance of the BCI real-time system that uses a training protocol of type 1 or 3 and that operates during the assessment with a sequence of movements similar to the one defined in protocol 2. The different protocols are described in Table 4.2, as reminder :

1. Protocol 1 : Sequence is defined as 3 series of 10 movements, e.g. : 10 left-hands followed by 10 rest states and then 10 right-hands, etc.

2. Protocol 2 : Sequence is defined as 15 series of 2 movements, e.g. : 2 left-hands followed by 2 rest states, then 2 right-hands followed by 2 rest sates again, etc.

3. Protocol 3 : Sequence is defined as 30 series of 1 movement in a random order, e.g. : 1 left-hand followed by 2 rest state, then 1 right-hand followed by 2 left- hand again, etc.

For the first assessment, all ideal conditions requirements are preserved. The only difference lies in the fact the system has been trained with a different training protocols. Results for training protocol of type 1 and type 3 are respectively presented in Fig.6.8 and Fig.6.9.

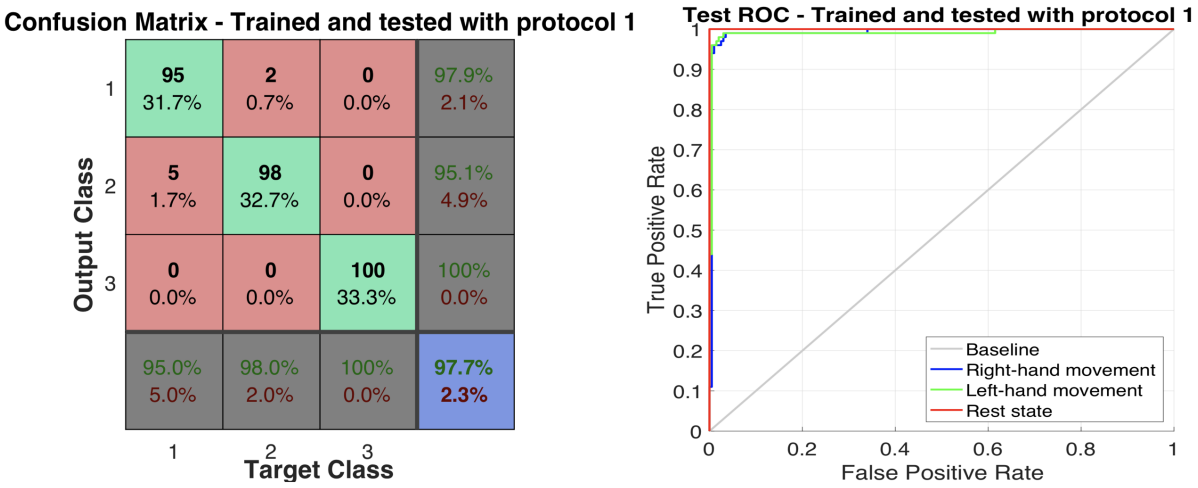


Figure 6.8: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 1.

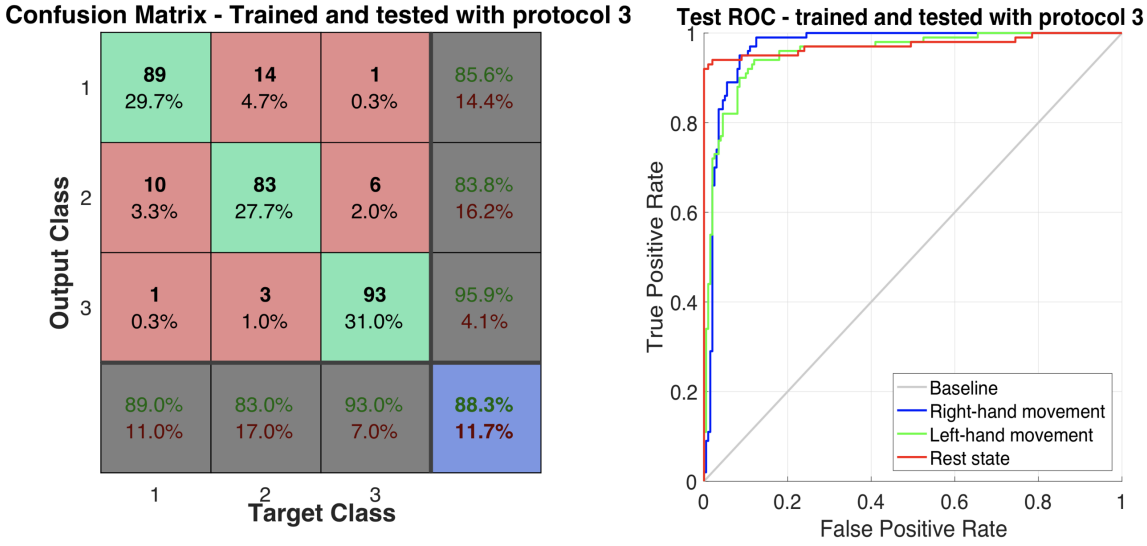


Figure 6.9: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 3.

One may analyze these results. The first element to notice is that the accuracy obtained during the assessment of protocol 1 (**97.7%**) is better than the one from protocol 2 (**95%**) being itself better than the accuracy obtained with protocol 3 (**88.3%**). This highlights the importance

of the training database and protocol for a BCI design. Nevertheless one must notice that the protocol 3 is the most natural one given that the user performs only once the desired movement. In the protocol 2, the user is told to execute two times the desired movement in series in order to obtain the announced classification rate. In the protocol 1, the user is told to execute 10 times in a row the same movement to obtain such a classification rate. **Therefore authors only use the second and third protocol, the first one being unrealistic for a daily usage of the BCI real-time system even if it exhibits a good classification rate. The second protocol seems to be a good compromise because it offers a classification rate of 95% for only two repetitions of the desired movement (8 [s] in total).**

Several reasons can justify such differences between protocols :

1. The habit effect still plays an important role here. When the BCI real-time system is trained with protocol 1, the different movements are repeated numerous times in a row. The human body undergoes this repetition and an habit effect appears. This creates a voltage drift in the repeated signals (that can be seen as a slow cortical potential on top of the ERP). The training database is thus biased, leading to a real-time BCI system that gives accurate results only when a given movement is repeated several time.

2. The habit effect is not present when the BCI real-time system is trained with protocol 3 because movements are different for each iteration. Eliminating this habit effect leads to results worse than those obtained with protocol 2. **This means that, even if it is not desired by the authors, the habit effect helps to improve the classification rate**. The EEG-based BCI system designed in this work makes indirectly the most of the habit effect.

Let's consider the second assessment. All other ideal conditions requirements are preserved. The assessment procedure is the same as the one described earlier. The only difference lies in the fact that the BCI real-time system is assessed with a user that performs series of two movements (as in protocol 2) with a trained model that uses alternative protocols (1 and 3). Assessment results, using training database with protocol 1 and 3 and evaluated with movements sequence from protocol 2, are respectively presented in Fig.6.10 and 6.11.

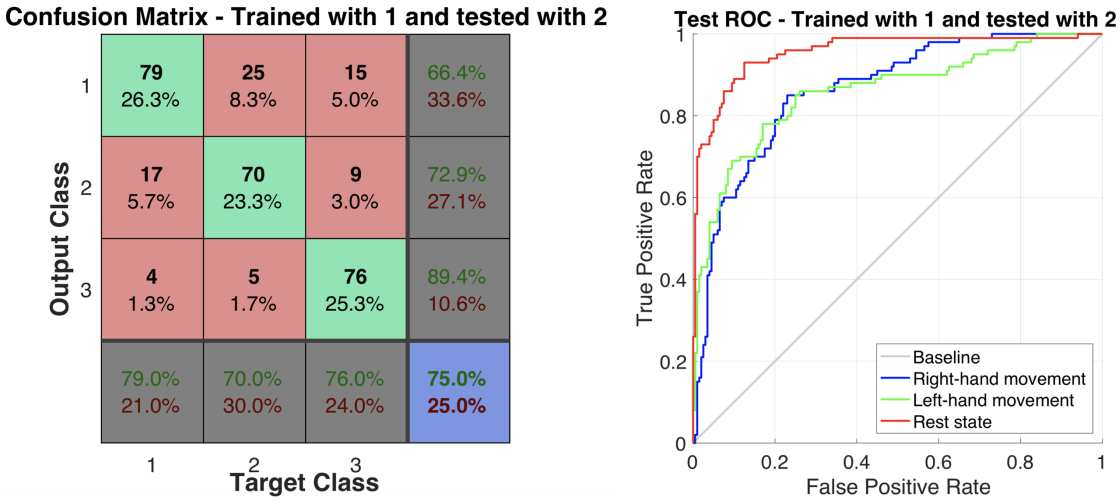


Figure 6.10: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 1 and assessed based on a sequence of movements similar to protocol 2.

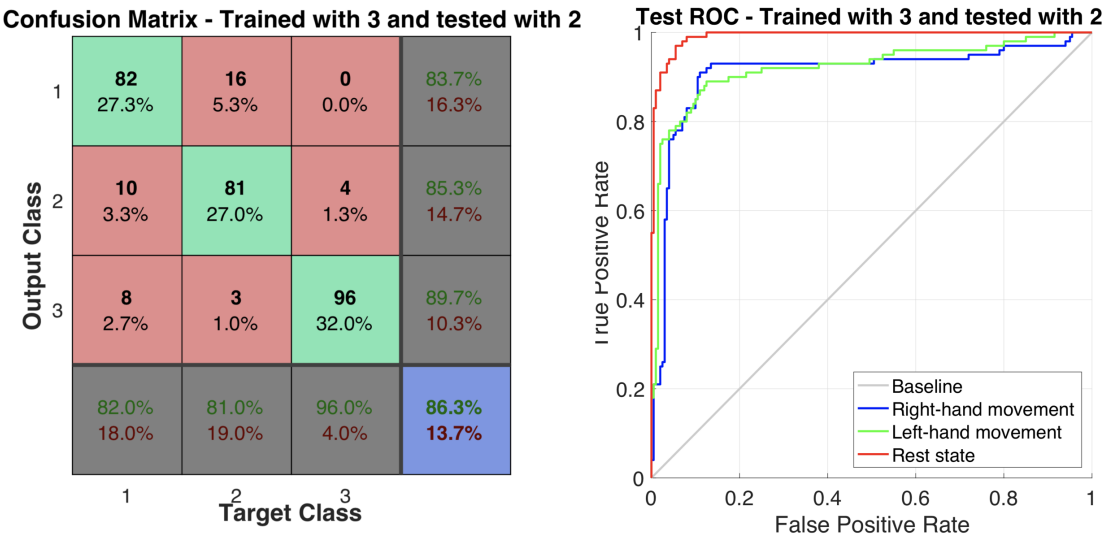


Figure 6.11: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired using protocol 3 and assessed based on a sequence of movements similar to protocol 2.

One may analyze these results. The first element to notice is that, in any case, the classification accuracy has decreased.

1. In the first case, the decrease is **large**. Indeed the accuracy evolves from **97.7%** to **75%** when the user goes from a sequence of movements equivalent to the training database with protocol 1 (10 same movements in a row) to a sequence of movements similar to protocol 2 (2 same movements in a row).

2. In the second case, the decrease is **moderate**. Indeed the accuracy evolves from **88.3%** to **86.3%** when the user goes from a sequence of movements equivalent to the training database with protocol 3 (1 movement at a time, random order, no movement repetition) to a sequence of movements similar to protocol 2 (2 same movements in a row).

One may also remark that in both cases the precision of the class 3 is better than other classes which means that the BCI remains good at predicting the presence of a movement. However the precision in the first case is given, respectively for class 1 and 2, by P_{C1} = **66.4%**, P_{C2} = **72.9%**. This means that the system encounters difficulties for distinguishing a right-hand movement from a left-hand movement.

Eventually the fact that **the second case is almost not influenced by the change of protocol** can be justified as follows : The protocol 3 is the most general and natural one. It follows the quote : "Who can do more can do less". Indeed given its classification ability for very eclectic sequences of movements, the system can give good results when the sequence is better arranged.

To summarize, the type of training database and protocol has an impact on the classification accuracy. The training protocol must be similar to the expected usage of the BCI real-time system. Furthermore, even the type of protocol has an impact on the classification accuracy.**One must very careful when claiming that he/she has designed a BCI that exhibits a good classification rate. Results can fall over when the BCI is used differently from its expected use.** One will now study performance degradation related to the environment variation.

6.3.4 Environment variation

For studying the influence of this parameter, one will assess the performance of the BCI real-time system that has been trained in a given room and tested in another. Here are some characteristics of these two rooms

1. Room 1 : 77 $[m^2]$ Computer room. Neon lights and lots of electronic devices. Training room. Nobody else in the room except the user.

2. Room 2: 20 $[m^2]$ Sleeping room. Filament lamps and no electronic devices (other than the BCI system itself and the computer connected to the system). Testing room. Nobody else in the room except the user.

All other ideal conditions requirements are preserved. The assessment procedure is the same as the one described earlier. The only difference lies in the fact that the BCI real-time system is assessed in room 2 with a trained model obtained in room 1 (same room as the one used for all other assessments). Results are presented in Fig.6.12

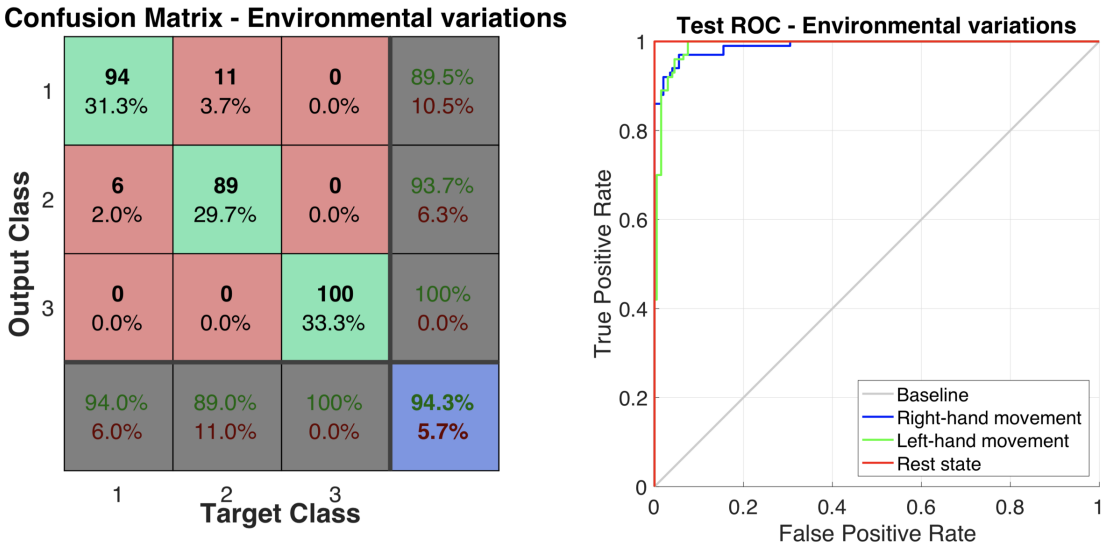


Figure 6.12: Confusion matrix and ROC plot for 1 assessment of the BCI real-time system with a training database acquired in a given environment and assessed in another environment.

One may analyze these results. One must notice that the accuracy is roughly (**94%** vs **95%**) the same as in ideal conditions. This difference is negligible and one may assume that results obtained in these two rooms are equivalent. This is good news because **it means that environment does not affect the classification accuracy, as long as the environment does not influence the user's state itself.** This conclusion makes sense because most of the environmental noise is high-frequency noise which is filtered by lowpass filters and Notch filters within the processing chain.

6.4 Conclusion

Within this chapter the authors have assessed the performance of the BCI real-time system that has been developed in this work from chapter 1 to chapter 5. The purpose of such an assessment is confirming results obtained in chapter 5 during the classification stage.

In order for this assessment to be realized, the authors have reproduced exactly the training database acquisition conditions. This is defined by what the authors have called : **ideal conditions requirements**. These four conditions are : same user for training and assessment, system trained with an updated training database, same acquisition protocol for training and assessment, same environment for training and assessment. **With these four conditions, the BCI real-time system produces predictions with an accuracy of 95 %. One may therefore conclude that the BCI real-time system works as expected under ideal conditions.**

In a second time, the authors have evaluated the **robustness** of the BCI real-time system by dropping each ideal condition (only one condition is removed during an assessment).

1. When the user is different during the training process and the evaluation process, the results can be drastically worse. One has noticed an **accuracy drop of about 15%**. Three possible reasons have been mentioned : The difference in terms of brain signals between two people, the movements execution manner and the difference in head morphology.

2. When the training database is old, results can be degraded too. One has noticed an **accuracy drop of about 9%**. Three possible reasons have been mentioned : The user's state, the user's physical changes (haircut length) and the usage rate of the BCI system.

3. When the acquisition protocol is different for the training process and the assessment, results may drastically decrease. One has noticed an **accuracy drop up to 20%**. Several reasons related to the **habituation effect** have been mentioned.

4. When the environment is different for the training process and the assessment, **results are not really influenced**.

The robustness of a BCI real-time system is crucial. A BCI is always used under assumptions and thus the authors have not investigated the worst case scenario. **One may conclude that the BCI performances are rapidly degraded when ideal conditions are not satisfied. Therefore the BCI system is not as robust as the authors would have desired. Nevertheless, if the BCI user takes care and uses it following the author's instructions, he/she may execute three different actions with an accuracy of 95%.**

Conclusion

Conclusion and main results

A *Brain Machine Interface* is able to record signals coming from the brain, translate them into commands that depict the user's intent and provide feedback to them. This kind of device is particularly useful for tetraplegic patients who have lost their ability to move their limbs but still have a functional brain. With the increasing performances of new technologies and electronics, the field of BMI is booming and more and more studies are made to develop such devices.

In this work, the development and implementation of a complete BMI on an experimental board was investigated. The aim of our work, in collaboration with *Synergia Medical*, was to continue the work done in the previous Master Thesis [3] [4] in order to come closer to a system that could potentially be put on the market. To reach this objective, different steps in our work have been followed.

First, it was important to present the context and the goal of developing a BMI. Therefore, the disease of tetraplegia was discussed. Some important concepts concerning brain anatomy, bio-electric signals and acquisition methods, among others, were also explained in order to understand the topic and its challenges. The specifications for the type of BMI to implement were also listed and it was established that it should be based on EEG signals leading to*non-invasive* system, meaning that it will record signal from the scalp. It should also be *synchronous* and *active* since it should record the user's intention and not its reaction to external stimuli. The BMI should also record event-related potentials related to SMR rythms.

A BMI consists of a processing chain so the design can be separated in a few different steps. Chapter 2 presented a state of the art of the processing techniques, spatial filters types and characteristics,feature extraction and classification algorithms. We saw that lots of techniques exist and that a selection had to be made.

In 2016, an experimental board was developed, a helmet was printed in 3D and three electrodes were integrated to this helmet in order to have a system able to record EEG signals. In Chapter 3, a review of the board and its main characteristics was presented. The FPGA is the central component of the board that executes the processing chain, but components such as ADCs and filters are also detailed, among others. One of the objectives of our work was to improve the material developed in 2016 and the improvements made were presented. The helmet was updated to its last version, the *Ultra Cortex Mark IV*, and new electrodes were added to be able to go from three to five recording electrodes. The hardware specifications were also established at this end of this chapter and it was stated that the BMI should comprise the updated version of the helmet and include the five electrodes C3, Cz, C4, CP1 and CP2.

In Chapter 4, a `MATLAB` application was developed. We called it *Algorithm exploration tool*. In this tool, the whole processing chain presented in Chapter 2 was implemented offline. Various pro-

cessing options were implemented, and also different feature extraction and feature classification methods. This tool allowed us to decompose the processing chain step by step, explore various algorithms and their combinations for finding the ones giving the best classification accuracy. After having collected lots of results, the algorithms choices and specifications were established by observing the most promising combinations. Indeed, it was chosen that the BMI should feature a FIR filter of order 16. The BMI should also be developed by using the *CSP* spatial filter, which seems to give the best results among other filters. The feature extraction method should be the *Discrete Wavelets decomposition* method with a 5 levels and extract the following features for each decomposition level: Willison amplitude, variance, maximum and minimum. The classifier should be implemented using a *Multilayer Perceptron Neural Network* with 30 neurons and one hidden layer. **This combination gave the very good offline result of 99.4 % as a classification accuracy and is therefore a great improvement compared to the results obtained in the previous years, with a classification almost 30% more accurate.**

After having defined all the specifications that the BMI should complete, the real time application could be started. First, a precise protocol for the experiments was explained. Indeed, a clear protocol had to be established. An overview of the real-time processing chain was also explained in details to completely understand the challenges of the implementation on the FPGA. The `Verilog` implementation and the performances of the filters was then detailed. After this, the software implementation on the integrated microprocessor of the FPGA was detailed. Indeed, the major part of the processing chain was implemented using `C/C++` languages. For every step, the algorithm was detailed in pseudocode. First, the translation of the CSP spatial filter algorithm was explained in details. The model of the CSP is trained offline on `MATLAB` before being passed to the FPGA. The same procedure goes with feature translation algorithm using the Wavelets transform, where the algorithm was explained in details. The feature classification step is then described. The classifier is based on the *MLP Neural Network* algorithm. Some performances are then evaluated. The timing constraints are evaluated and proved to respect the specifications. The memory management and power management are finally briefly investigated.

In Chapter 6, the results of our BMI real-time system were presented. First, the performance metrics were established. It was explained that the confusion matrix is an important metric to quantify the accuracy of the classifier and thus, the BMI real-time system. We also saw that there exist other different ways of evaluating the performances of a BMI, such as the F1 score, the specificity, sensitivity and the Receiver Operating Characteristics curve. Then an assessment procedure was presented. This procedure also introduced some ideal conditions requirements, four in total : Same user for training and assessment, system trained with an updated training database, same acquisition protocol for training and assessment, same environment for training and assessment. The main result obtained is that the BCI real-time system can very accurately predict the presence of a movement. However, it can commit several errors when interpreting if it is a left or right arm movement in ideal conditions. **Its global real-time accuracy is about 95% in ideal conditions.** The results for different (non-ideal) cases were also collected. Indeed, it was decided that varying some parameters in the acquisition protocol would be a great opportunity to figure out which parameters have a great influence on the results. These results in non-ideal conditions allowed us to understand how sensitive the acquisition protocol may be. First, the BCI user was changed. It was noticed that changing from one subject to another could lead to worse classification results. Then, we explored the time difficulties that can occur when recording a database and using it for classification a certain period of time afterwards. We saw that the older the database was, the worse the classification accuracy could be, while using the BCI just after having recorded the database gives better results. The protocol was also changed to see if it had an influence. We saw that changing the protocol for the database acquisition could potentially lead to different results since we observed something called the *habituation*

effect (or *habit effect*) : **when the subject repeats the movement a certain number of time, the BCI seems to continue detecting the same movement even if the subject changed it.** Finally, the last parameter that was changed was the environment and we saw that after having changed the environment, the BCI roughly gave the same results.

Further improvements and possibilities

Even if we are satisfied with our work, some improvements could be done to enhance the quality of the results and the possibilities of the BMI system. Some of the main ideas for improvements are listed here after.

1. More extensive use of the FieldTrip toolbox

In Chapter 4, the FieldTrip toolbox was presented. In our work, it has been used for preprocessing stages and more particularly for frequency filtering. The function *ft_preprocessing* made it easier to deal with all kinds of parameters for the filters such as the filter type, the filter order, the type of window, etc. However, this toolbox is enormous and could be used for other stages of the BCI implementation. Among others, we could cite the statistical analysis with various functions. A better exploration of this toolbox could open doors for new strategies and better results for EEG signal processing and BCI implementation.

2. Motor Imagery and more movements exploration

In our work, we decided to detect and classify real movements. However, the final goal of the system is to detect imaginary movements to help tetraplegic patients retrieve their ability. This is called *Motor Imagery*. To implement Motor Imagery, a strong database containing lots of trials would be needed. A certain period of time would have to be consecrated for the training as it may be difficult for some people to imagine a movement strongly enough such that EEG signals present the characteristics of a movement. Due to a lack of time, imaginary movements were not tested but it certainly is one of the improvements that could be made in the continuity of our work.

Besides Motor Imagery, we could also imagine taking more movements such as feet movement into account. This would add complexity for the classifier because the number of possible movements would go from two to four and more states would thus be needed.

3. Material improvements

The previous Master Thesis [3],[4] used the three electrodes C3, C4, Cz integrated to the helmet to record EEG signals from the scalp. This year, one of our goal was to improve the material provided from the last years. We went from these three electrodes to five by updating the version of the helmet and adding the electrodes Cp1 and Cp2. The addition of these two electrodes enabled to collect more signals around the sensory-motor cortex and apply better spatial filters. By doing this, we obtained better a better classification accuracy than in the previous years. The new version of the helmet can incorporate more electrodes than five, but we were limited to this number because of the limited memory and number of logic elements in the FPGA, as explained in Chapter 5. An improvement would thus be to either find a way to manage the memory such that more electrodes could be added, or change the FPGA into a better one with more logic elements. Having more logic elements available would also allow to implement more efficient frequency filters for the preprocessing of EEG signals. Moreover, the features vectors could be larger and contain more information and the Neural Network could be implemented with more hidden layers and neurons. Last but not least, an *asynchronous* BCI

where the signal acquisition and the classification are made in parallel could be implemented. All these extra features could lead to even better classification results.

4. Database

For the spatial filter and classifier model training, a database was established. For this purpose, lots of trials were recorded for every possible state (right arm, left arm, rest). The database has now 500 trials for each state and was quite convenient for our implementation. However, the trials were recorded on only person that provided better EEG signals, maybe due to the form of the head that was more adapted to the helmet or the length of the hair. One improvement would be to do a cross person database with trials recorded on more than one person. This would require more time and investment but could help building a very strong database, adaptive to different people. Indeed, if only one person is recorded, the BCI will tend to work better on this particular person, and will probably give inferior results for others.

5. Online training and classification

In our real time BCI, the model training of the classifier is first performed offline on `MATLAB` with the database previously recorded and only then passed to the FPGA for online classification. One possible improvement, which would probably also require to update the material and the FPGA, would be to perform online training of the model. This technique would not only have the advantage of being adaptive to everyone, since a small training phase would be needed before starting the real use of the BCI, but the system would also be completely independent from offline training, making it more portable and insensitive to non-idealities such as database age variations and user's state variations, as explained in Chapter 6.

6. MATLAB tool: EEG simulation

At the beginning of our work, we had the idea of creating a `MATLAB` tool that would allow us to explore all algorithms concerning BCI design and implementation. In this tool, we also wanted to include a simulation tool that would be able to simulate EEG signals where a movement occurs in order to test rapidly the different algorithms, without having to record real trials on patients. After having developed a simple EEG generation model, we integrated it to the `MATLAB` tool and used it during a while. However, after having collected some real trials on patients, we rapidly came to the conclusion that the EEG signals generated were not realistic. Having realized that we already had other major points to develop during our work and that this EEG generation tool is outside the scope of our Master Thesis, we decided to leave it for further improvements. The integration in the `MATLAB` tool is already done but is rather basic and needs more consideration. To be validated, the signals generated should be extensively compared to real signals. However, we think that we had a good starting idea and that it should be investigated in the future years, as EEG signal generation could be really helpful in the field of BCI design.

Bibliography

[1] Wolpaw J., Wolpaw E.,*"Brain Computer Interface : Principles and Practice"*, 2014

[2] Hassanien A. , Taher Azar A.,*"Brain Computer Interface : Current trends and application"*, 2015

[3] Schramme M., Delhaye B.,*"Design of a BMI for tetraplegic patients"*, June 2016

[4] Govaerts J., Parez R.,*"Design of a BMI for tetraplegic patients"*, June 2017

[5] Jolivet R. , Magistretti P.J., Weber B., *Deciphering neuron-glia compartmentalization in cortical energy metabolism.*,Front Neuroenerg, 2009

[6] Cherry S. , Phelps M., *Imaging brain function with positron emission tomography, In Brain Mapping: Th e Methods.*, Toga AW, Academic Press, 2002

[7] spinalcord.com, `https://www.spinalcord.com/quadriplegia-tetraplegia`, May 2018

[8] Dum R.P. , Strick P.L. *The origin of corticospinal projections from the premotor areas in the frontal lobe.*, J Neurosci 11, 1991

[9] Fu Q.G. , Flament D/ , Coltz J.D. , Ebner T.J., *Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons.*, J Neurophysiol 73, 1995

[10] Hutchins K.D. , Martino A.M. , Strick P.L., *Corticospinal projections from the medial wall of the hemisphere.* Exp Brain Res 71, 1988

[11] Brunner P. , Bianchi L. , Guger C. , Cincotti F. and Schalk G, *Current trends in hardware and soft ware for brain–computer interfaces (BCIs).*, J Neural Eng, 2011

[12] Roadmap Horizon 2020, `http://bnci-horizon-2020.eu`, April 2015

[13] Synergia Medical website `http://www.synergia-medical.com`

[14] Picton T.W. , Bentin S. , Berg P. , Donchin E. , Hillyard S.A. , Johnson R. Jr , Miller G.A. , Ritter W. , Ruchkin D.S. , Rugg M.D. , and Taylor M.J., *Guidelines for using human event-related potentials to study cognition.*, Psychophysiology 37, 2000 .

[15] Pilgreen K.L., *Physiologic, medical and cognitive correlates of electroencephalography.*, Oxford University Press, 1995

[16] di Pellegrino G. , Fadiga L. , Fogassi L. , Gallese V. , Rizzolatti G. *Understanding motor events: A neurophysiological study.* Exp Brain Res 91, 1992

[17] Cabel D.W. , Cisek P. , Scott S.H. , *Neural activity in primary motor cortex related to mechanical loads applied to the shoulder and elbow during a postural task.* J Neurophysiol 86, 2001

[18] Crammond D.J. , Kalaska J.F., *Prior information in motor and premotor cortex: Activity during the delay period and effect on pre-movement activity.*, J Neurophysiol 84, 2000

[19] Huettel S. , McCarthy G., *What is odd about the odd-ball task? Prefrontal cortex is activated by dynamic changes in response strategy*, Neuropsychologia 42, 2004

[20] Thompson R. , Spencer W. , *Habituation: a model phenomenon for the study of neuronal substrates of behavior.*, Psychological Review. No.1.73, 1966

[21] Thompson R.F., *Habituation: A history.*, Neurobiology of Learning and Memory, 2009

[22] Kandel E. , Schwartz J. , Jessel T. , *Principles of Neural Science.* Norwalk, CT : Appleton & Lange, 1991

[23] Kamen G., E*lectromyographic Kinesiology.*, Human Kinetics Publ., 2004

[24] Hibbs A.E., Thompson K.G., French D.N., Hodgson D., Spears I.R. *Peak and average rectified EMG measures: Which method of data reduction should be used for assessing core training exercises?*, Journal of Electromyography and Kinesiology, 2011

[25] Anderson D.J., *Penetrating multichannel stimulation and recording electrodes in auditory prosthesis research.*, Hear Res 242, 2008

[26] Rousche P.J., Normann R.A ., *Chronic recording capability of the Utah intracortical electrode array in cat sensory cortex.*, J Neurosci Methods, 1998

[27] Grill W. M., Mortimer J.T., E*lectrical properties of implant encapsulation tissue.*, J Biomed Eng, 1994

[28] Chestek C. A., Cunningham J. P., Gilja V., Nuyujukian P., Ryu S.I., Shenoy K.V., *Neural prosthetic systems: current problems and future directions.* , Conf Proc IEEE Eng Med Biol Soc, 2009

[29] Ferree T.C. *Spherical splines and average referencing in scalp EEG.*, Brain Topogr. 19, 2006

[30] Ferree T.C , Luu P. , Russell G.S., Tucker D.M. , *Scalp electrode impedance, infection risk, and EEG data quality.* Clin Neurophysiol. 112, 2001

[31] Hashiguchi K., Morioka T., Yoshida F., Miyagi Y., et al., *Correlation between scalp-recorded electroencephalographic and electrocorticographic activities during ictal period.*, Seizure. 16, 2007

[32] Schuh L., Drury I., *Intraoperative electrocorticography and direct cortical electrical stimulation.*, Seminars in Anesthesia, 1996

[33] Miller K.J.; deNijs M., Shenoy P., et al., *Real-time functional brain mapping using electrocorticography.*, NeuroImage. 37, 2007

[34] Wise K.D., Angell J.B., Starr A., *An integrated-circuit approach to extracellular microelectrodes.*, IEEE Trans Biomed Eng, 1970

[35] Subbaroyan J., Martin D.C., Kipke D.R., *A finite-element model of the mechanical effects of implantable microelectrodes in the cerebral cortex.*, J Neural Eng, 2, 2005

[36] Pfurtscheller G. , da Silva F.H.L. ,*"Event-related EEG/MEG synchronization and desynchronization: basic principles."*, Clin Neurophysiol, 1999.

[37] da Silva F.H.L. ,*" Neural mechanisms underlying brain waves: from neural membranes to networks."*, Electroencephalogr Clin Neurophysiol, 1991.

[38] Open BCI `www.openbci.com`

[39] FieldTrip Toolbox and tutorials `http://www.fieldtriptoolbox.org`

[40] Suleiman A., Fatehi T.,*"Feature extraction techniques of EEG signal for BCI applications"*

[41] Amjed S. Al-Fahoum, Ausilah A. Al-Fraihat,*Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains*

[42] Schlögl A.,*The electroencephalogram and the adaptative autoregressive model: theory and applications*, April 2000

[43] Kumar M.,Kumar D. *Wavelet Based Feature Extraction Scheme of Electroencephalogram*, February 2014

[44] CNN article `https://edition.cnn.com/2017/04/12/health/brain-computer-interface-partner/index.html`

[45] Müller K.-R, Krauledat M., Dornhege G., Curio G., Blankertz B. *Machine learning techniques for brain-computer interfaces*

[46] Lotte F., Congedo M., Lécuyer A., Lamarche F., Arnaldi B. *A review of classification algorithms for EEG-based brain–computer interfaces*

[47] Sridhar G.V. *An improved MultiLayer Pereptron for BCI with Evolutionary Optimization*

[48] Günter S.,Bunke H. *Fast Feature Selection in an HMM-based Multiple Classifier System for Handwriting Recognition*

[49] FieldTrip toolbox and tutorials `http://www.fieldtriptoolbox.org`

[50] Whitmarsh S. *FieldTrip for Dummies*

[51] `https://github.com/fieldtrip/fieldtrip`

[52] Texas Instruments, *ADS 1299 datasheet*, `http://www.ti.com/lit/ds/symlink/ads1299.pdf`, May 2018

[53] Atera, *FPGA MAX 1050 datasheet*, `https://www.altera.com/en_US/pdfs/literature/hb/max-10/m10_datasheet.pdf`, May 2018

[54] Texas Instruments, *CC430 microcontroller datasheet*, `http://www.ti.com/lit/ml/slat124a/slat124a.pdf`, May 2018

[55] Spüler M. et al, *A high-speed brain-computer interface (BCI) using dry EEG electrodes.*,PLoS ONE journal, 2017

[56] FLorida Research Instruments, *Dry and wet electrodes package*, `https://fri-fl-shop.com/`, May 2018

[57] Speier W. et al, *A Method for Optimizing EEG Electrode Number and Configuration for Signal Acquisition in P300 Speller Systems*,Clin Neurophysiol, 2015

[58] Popescu F. , Fazli S. , Badower Y. et al, *Single trial classification of motor imagination using 6 dry EEG electrodes.*, PLoS One journal, 2007

[59] Offner , F. F. *The EEG as potential mapping: The value of the average monopolar reference*, Electroencephalogr Clin Neurophysiol, 1950

[60] Georgopoulos A. , Schwartz A. , Kettner R., *Neuronal population coding of movement direction*, Science 233, 1986

[61] Lu X. , Ashe J., *Anticipatory activity in primary motor cortex codes memorized movement sequences*, Neuron 45, 2005

[62] Pfurtscheller G. et al, *Separability of EEG Signals Recorded During Right and Left Motor Imagery Using Adaptive Auto-regressive Parameters*, IEEE transactions on rehabilitation engineering, 1998

[63] Schalk G ., *Effective brain-computer interfacing using BCI2000*, Conference IEEE Biomedical Engineering, 2009 .

[64] Noorzi M.I., Faye I., *A review of EEG signal simulation methods*, Neural information processing 23rd conference, 2016

[65] Perera H., Fairuz M., Wong K. W., *A review of electroencephalogram-Based Analysis and Classification Frameworks for Dyslexia*, Neural information processing 23rd conference, 2016

[66] Vaz F., Oliveira P.G., Principe J.C., *A study on the best order for autoregressive EEG modeling*, Int. Journal of Bio-Med computations, 1987

[67] McClellan J. , Schafer R. , and Yoder M., *Signal Processing First*, Pearson, 2003

[68] Qin L. and He B., *A wavelet-based time-frequency analysis approach for classification of motor imagery for brain–computer interface applications*, J. Neural Eng. 2, 2005

[69] Wolpaw J., David S., McCane L., McFarland D., *Spatial filter selection for EEG-based communication*, 1997

[70] Babiloni F. et al., *Mahalanobis Distance-Based Classifiers are able to recognize EEG Patterns by using few EEG electrodes*, 2001

[71] Oana Diana Eva, *Comparison of classifiers and statistical analysis for EEG signals used in Brain Computer Interface motor task paradigm*, 2015

[72] Konar A. et al., *K-Nearest Neighbor Classification of Left-Right Limb Movement Using EEG Data*, 2014

[73] Hoffman U. et a., *Classification of EEG Signals Using Dempster Shafer Theory and a K-Nearest Neighbor Classifier*, 2009

[74] Powers D., *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation.*, Journal of Machine Learning Technologies. 2 (1): 37–63, 2011

[75] Betz M.A., Gabriel K.R., *Type IV Errors and Analysis of Simple Effects*, Journal of Educational Statistics, Vol.3, 1978

[76] Lasko T.A., Bhagwat K.H., Ohno-Machado L. *The use of receiver operating characteristic curves in biomedical informatics.*, Journal of Biomedical Informatics, 38(5):404–415, 2005

[77] Gonen M., *Analyzing Receiver Operating Characteristic Curves Using SAS*, SAS Press, 2007

[78] Meyer Y., *Wavelets and Operators.*, Cambridge: Cambridge University Press, 1992

[79] Chui C. K., *An Introduction to Wavelets.* San Diego: Academic Press,1992

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**