Final year internship report

Social Robotics: Human-Robot Interaction Using EEG Signals and Head Motion

Final year project from 9th March to 7th August 2015

Siméon Capy
INSA de Strasbourg
Spécialité mécatronique
Promotion Pierre Richard

Université de Strasbourg
Master sciences, technologies, santé
Mention imagerie, robotique et ingénierie pour le vivant
Spécialité vision, automatique, nanophotonique

Cardiff University
Queen's Buildings
5 The Parade
CF24 3AA Cardiff

Mentors

Rossi Setchi
Director, Mechanics, Materials
and Advanced Manufacturing
Tel.: +44 29 2087 5720
E-Mail: setchi@cardiff.ac.uk

Cecilia Zanni-Merk
Responsable du centre des ressources
pédagogiques en informatique de l'INSA
Tel.: +33 3 88 14 47 82
E-Mail: cecilia.zanni-merk@insa-strasbourg.fr

Institut national des sciences appliquées de Strasbourg

Projet de Fin d'Études

Auteur: Capy Siméon	**Promotion**: 2015
Titre: Social Robotics: Human-Robot Interaction Using EEG Signals and Head Motion	**Soutenance**: Septembre
Structure d'accueil: Cardiff University	
Nb de volume(s): 1 **Nb de pages** : 50	
Nb de références bibliographiques: 42	

Résumé:
Ce stage s'inscrit dans le cadre de l'aide aux personnes âgées menée par l'Université de Cardiff. Il vise à réaliser des interactions sociales au moyen d'un robot (un Turtlebot) et d'un casque électroencéphalographique (Emotiv Epoch).
Dans le cadre d'un système de visioconférence amélioré, imaginé durant le stage, l'algorithme développé s'est concentré sur la détection des clignements des yeux de l'utilisateur dans le but d'être retransmis sur un écran. L'algorithme s'occupe également de détecter les mouvements de la tête de l'utilisateur grâce à l'accéléromètre du casque Emotiv.
L'algorithme a été développé avec un double logiciel client/serveur de communication réseau, afin de permettre à l'ordinateur de l'utilisateur d'envoyer les ordres au robot pour le commander d'après les mouvements de tête de l'utilisateur, sans y être connecté physiquement. Le logiciel sur le robot s'occupe également d'afficher l'image des yeux établie par l'algorithme (avec une précision de 68 %). Ainsi, les personnes qui ont des difficultés à se mouvoir pourraient communiquer presque normalement avec leur entourage. En outre, cela pourrait être utile pour la communication entre personnes très éloignées.

Mots clés: Électroencephalographie, séparation aveugle de sources, reconnaissance de motifs, extraction du clignement des yeux

Traduction:
This internship fits into the context of elderly people health-care conducted at Cardiff University. The purpose of this internship was to realise social interactions with a robot (a Turtlebot) using an electroencephalographic headset (Emotiv Epoch).
As part of an improved visioconferencing system proposed within this project, algorithms have been developed to detect eye blinks of a user. The aim is to display the eye blinks on a screen located on the robot. The developed algorithm also enabled the detection of movements of the user's head thanks to the accelerometer integrated in the Emotiv headset.
A double client/server program was also developed alongside the processing algorithm, in order to enable network communication, with the aim of allowing the user's computer to send orders to the robot without physical connection. The robot finally displays a representation of the user's eye (with an accurancy of 68%) and follows the direction given by the user when moving the head.
Thus, using the developed system people who have difficulties moving themselves would have the possibility to communicate in a more normal way than using nowadays telecommunication systems. In addition, this could be useful in order to improve long distance communication.

Acknowledgement

I would like to thank warmly Professor **Rossi Setchi** and **Cécilia Zanni-Merk** who gave me the opportunity to accomplish this internship, but also **Alex Noyvirt** for his involvement in the project. This internship was an amazing professional experience.

I acknowledge **Bruno Albert** too, who supervised my work, gave me precious advice and showed Cardiff to us.

I would also like to acknowledge the KES and the SRS teams of the Cardiff University which immediately and very kindly made me feel a part of the group.

Then, I would also like to thank my two co-workers **Yann Champetier** and **François de La Bourdonnaye** with whom I shared this internship. They helped me when I needed them.

Then, I thank warmly the OpenVIBE forum members who guided me when I started with the software.

Next, I thank **Julien Baldinot**, my flatmate, who helped me to improve my English and correct some of my mistakes.

Then I would like to acknowledge the members of the *Club robotique de l'INSA de Strasbourg*, through which I learnt a lot about robotics during my three years at INSA.

Finally, I thank all the members of my family who have always supported me in my projects.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This internship is the completion of my studies in mechatronics at INSA of Strasbourg (France) and in the IRIV (robotics and vision) master's degree at Univeristé de Strasbourg. The internship has been achieved at Cardiff University (Wales, United-Kingdom) in the engineering department.

1.1 Motivation

Interactions with robots become more and more important, so just this once won't hurt, let's begin with a definition:

> Social Robotics is the study of robots that interact and communicate among themselves, with humans, and with the environment, within the social and cultural structure attached to their roles.
>
> — International Journal of Social Robotics [2]

According to IJSR, the social robotics brings human-robot interactions **and** robot-robot interactions together. And not only human-robot interaction, as one might think at first sight. However, in this case, this project will focus on the human-robot interactions, especially with elderly people. The ultimate aim, of this branch of study, is to achieve a robot with human reactions. Indeed, it would be easier to communicate with a robot which looks and reacts like an human.

Research conducted at Cardiff University in the Knowledge Engineering Systems group aims at exploring human-robot interactions, in the context of the emerging field of social robotics. Completed a year ago, the SRS project [3] focused on the development and prototyping of remotely-controlled, semi-autonomous robotic solutions in domestic environments to support elderly people. The focus is now made on exploring new ways to interact with a robot using a range of different sensors.

1.2 Project presentation: Aim and objectives

1.2.1 Aim

The aim is to come to the aid of elderly people (or maybe disabled people with the EEG headset, see below). Nowadays, the elderly people population is increasing, and as a result, the demand in elderly nurse. In some country, e.g. Japan, over a quarter of the population is more than 65 years old. In order to have enough nurse for dependent persons, robots can be used. Indeed, to help in everyday life, a robot can be adequate. Bringing some objects, carrying heavy loads or being a companion are feasible tasks for a robot, in order to be a relief for the nurses.

1.2.2 Objectives

In this project, the focus is put on using EEG (electroencephalography) signals and head motion data provided by an Emotiv headset [4] to interact socially with the robot (a Turtlebot [5]).

The human-computer interfaces (HCI) are plethora, from classic buttons to advanced touch-screen. New developing interfaces are the BCIs, or, **brain-computer interfaces**[1]. Their development really began in the 90's [6], with as first goal to restore mobility to paralysed or amputated patients. Indeed, to be as real as possible a prosthesis must be *connected* to the brain. So as to control the prosthesis as a real member, and, with a *duplex* BCI, get sensations.

Two types of BCIs exist, the invasive and non-invasive BCIs. The first category uses electrodes implanted directly inside the skull, at the brain surface with a surgery. This technique gives better results but comes with several risks. Invasive BCIs will not be considered in this work placement.

The second category uses several technologies such as,

- magnetic resonance imaging (MRI) or functional MRI (fMRI);
- electrooculography (EOG);
- electromyography (EMG);
- electroencephalography (EEG).

The last mentioned will be considered in this case. The invasive methods give better results than non-invasive. Because the brain interface is directly on the brain, the skull and scalp don't interfere with the signal. However, the non invasive methods have several advantages. These systems are lighter (except for MRIs), less expensive and easier to use. Moreover people are more inclined to use EEG than to undergo a surgery.

1.3 Outline

Firstly, a short history of the electroencephalography is drawn up, as well as the biological origins of the brain signals. But also, the classification of the brain signal. Then, a state of the art introduces the social interactions with a robot and an EEG headset, and how realised an immersion with this two elements. Next, the material used in this internship is described as well as the used and developed software in order to communicate between the robot and the EEG. The developed algorithm, with the purpose of realising the project charter described in the second part, is detailed afterwards. The penultimate part is the opportunity to see the algorithm results in changing several parameters. Lastly, the conclusion ends this report and summarises the complete work, draws up the limitations and the improvements to realise to perfect the project.

[1] Also called brain-machine interface

Chapter 2

EEG background

Electroencephalography is used in this internship to create social interactions with a robot. These last years, many projects using an electroencephalogram in robotics or computer science have been published. However, the electroencephalography is much older and has a very different objective.

In this chapter a short historic of the electroencephalography is given, as well as the vocabulary specific to this technique ; in order to understand the rest.

2.1 History

The history of electroencephalography (EEG), from the Greek *electro-* (the brain emits electric signals) *-encephalo-* (head in Ancient Greek) and *-graphy* (represents something written), starts in 1875 with the first measure of brain signals with a galvanometer by the Britannic Richard Caton (1842-1926, see Figure 2.1) [7]. Fritsch (1838–1927) and Hitzig (1838–1907) discovered the electrical stimulation of the human brain; then, Napoleon Cybulski (1854–1919) and his assistant, Adolf Beck (1863-1942), conducted several tests on different animal species that consisted of electrically stimulating their brain. Cybulski provided EEG evidence of an epileptic seizure in a dog caused by electrical stimulation.

Figure 2.1: Richard Caton

Pravidch-Neminsky (1879–1952), a Russian physiologist, is one of the first to draw an electroencephalogram of a dog. He calls it *electrocerebrogram*. He noticed several markers and characteristics, especially on the duration and cycle of some patterns. However, Hans Berger (1873–1941) is the first to record a human electroencephalogram in 1924. In 1929, he already presented the alpha rhythm (see below) as a major component of the EEG.

The EEG is mostly used to detect epilepsies or to study the sleep rhythms. Indeed, during an epileptic seizure, some abnormal rhythms appear on the EEG; they are typical of this disease. As described before, over history, EEGs have been mainly used in order to detect diseases or study the brain. This is their first goal. However, nowadays, more and more studies on the utilisation of EEG data to command some devices are published. The brain waves are used to control software, prosthesis, robot.... These first experiments started in the 70s[7]. This internship is aligned with this movement.

Two types of EEG exist: invasive and non-invasive. The non-invasive ones are the most popular, the EEG used in this study is a non-invasive one (see section 4.1.2 page 14). In this type of EEG the electrodes are placed on the surface of the scalp. A saline solution is often used to improve the conductivity, hence reduce the electrical resistance. The advantages of these EEGs are that they are easy to use and safe for the patient but at the cost of a lower accuracy. The second type of EEG is the invasive one. Here, the electrodes are directly implanted in the patient's brain. This method is more dangerous for the patient because this involves a surgery. In compensation, the accuracy is very good.

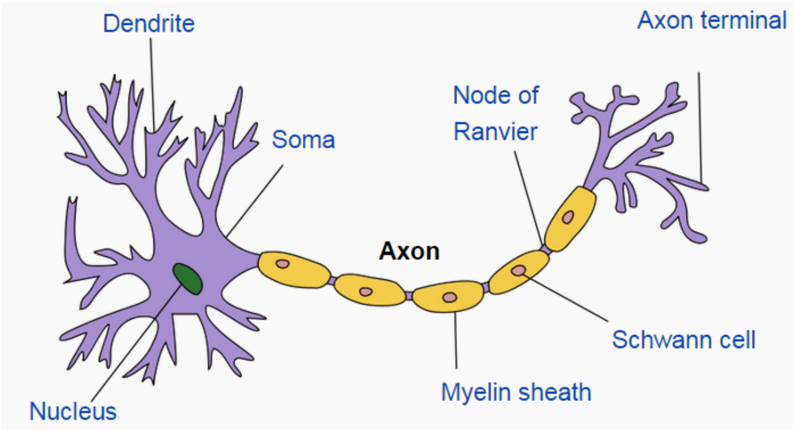


Figure 2.2: A neuron (Quasar Jarosz's picture CC BY-SA 3.0)

2.2 Biology

2.2.1 Signal origin

The neural activity of the human brain starts between the 17th and 23rd week of prenatal development, during the development of the central nervous system (CNS). The cells of the CNS are the glia cells and the neurones, the latter is composed of a nucleus, axons, dendrites, etc. (see Figure 2.2). The axon is a long wire between the nucleus that transmits the brain's electric impulses. The creation of this electric current is mostly due to the difference in concentration of ions, such as Na^+ and K^+. Some synapses, that are placed between the dendrites and the axodendrites, also emit an electrical current.

At rest, an electric potential between −60 and −70 mV exists around the cells' membranes. The variation of this potential over time is recorded by the EEG. However, the measure of the tension is degraded by the multiple layers between the electrodes and the neuron: the scalp, the skull and the brain itself. In addition, noise is generated by the other nerve cells, the muscles, the eyes and the power supply. Fortunately, the power supply band of frequency is very narrow: 50 Hz and can be easily eliminated with a notch filter.

2.2.2 Brain rhythms

Over time, the brain signals have been divided into several frequency bands. Some shared characteristics have been highlighted in each band, as when and where they appear. Bands that are consensus are the bands:

Alpha (between 8 and 13 Hz) Introduced in 1929 by Berger, this band is dominant when people are awake with their eyes closed. Alpha waves appear in the posterior half of the head and are usually found over the occipital region of the brain (see Figure 2.3 for the positions of the skull's bones).

Beta (more than 13 Hz) also discovered by Berger in 1929, they are emitted during normal and intense activities (thinking, solving problems, active attention...). Some pieces of literature give an upper bound of 26 Hz and introduce the gamma band (see below). The amplitude of beta rhythm is normally under 30 µV.

Theta (between 4 and 8 Hz) The notion of a theta wave was introduced by Wolter and Dovey in 1944. Theta waves appear as consciousness slips towards drowsiness and during the learning process. This band has been called theta in presumption of this origin: the thalamus.

Delta (between 0.5 and 4 Hz) W. Gray Walter is the first, in 1936, to call delta all frequencies below the alpha range before the introduction of theta band. This band appears mainly

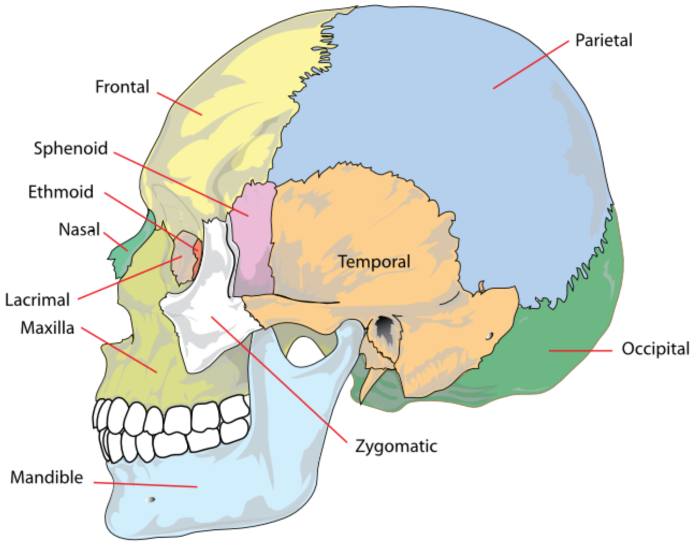


Figure 2.3: The different parts of the human skull (image by LadyofHats Mariana Ruiz Villarreal)

during the deep sleep. The band is also present when one is awake but it is easy to confuse the delta waves with the muscles artefacts.

In some literatures, the spectrum is more divided. Bands are added, some are shared by several literatures, others are more specific:

Gamma (above 30 Hz), also called fast beta waves was introduced in 1938 by Jasper and Andrews.

Phi (less 4 Hz) appears two seconds during eye closure.

Kappa the alpha waves in the anterior temporal region.

Tau the alpha waves in the temporal region.

mu, chi, lambda, etc.

2.3 Summary

Two types of electroencephalogram exist, and this study will only focus on the non-invasive one. There are also four rhythm bands generally accepted: alpha, beta, delta and theta.

Chapter 3

State of the art review

The aim of this project is to have social interactions with a robot and using EEG, in the framework of the SRS project, and so to aid elderly or disabled people.

The chosen way is the immersion *inside* the robot with an EEG headset and an head mounted display (HMD, see below), in order to interact with outside without moving. The system works like a *3D videoconferencing software*: in addition to the sound and the image, the user can move in the space with the robot. His displacements are natural because the robot is steered by the brain and the head's movements. The immersion is reinforced by the HMD, the user sees like the robot. To achieve this system, several devices are needed, for example, for the two main senses: the sight and the hearing. This project fits into this ecosystem.

3.1 Project specifications

The project charter was described in the outlines and defined in the first month, according to Pr Setchi.

The first objective of this project is to control the robot thanks to the Emotiv headset (using EEG and/or gyroscope). In order to achieve social interactions, according to the project outlines. The objective is to realize an immersive robot for disabled or elderly persons, like an improved videoconferencing system. The robot must be an avatar of the user, allowing him to move while staying at home. For this purpose, several things are required, the headset is used to:

- control the robot, forward and backward, with the EEG;
- turn on itself the robot, with the gyroscope;
- display the face movements (winking, frowning...).

It is the main part of the project. The user's facial expressions are used to have social interactions with other people.

Then, to improve the immersion, several components can be added. The first one, is the sight with a stereoscopic vision. The robot films *in 3D* and sends the video to the user who wears a virtual reality head-mounted display (VRHMD). In addition, to improve the immersion, the robot can record the sound and sends it to the user in an ear- or headphone.

All these parts put together, with the robot, constitute what it can be called as an immersive robot or an improved videoconferencing system (IVCS).

3.2 Immersive robot

Some research has been achieved on the control of robot using EEG. Several protocols are needed to acquire and understand the thoughts of the subject. Indeed, reading thoughts is not easy. They are drown in a plethora of signals from the muscles, the eyes, the hearth, normal brain activity, power supply of the measuring device, etc. That is why the software must be trained for each user, see section 3.3.2 p. 9.

3.2.1 Immersive robot control without Brain Computer Interface (BCI)

For controlled robots (most of robots are automatic), in addition to classical the keyboard and mouse combination, joysticks are often used [8]. Joystick utilisation is simple and quite intuitive. Moreover, the gamepads have several buttons which can be used in addition to the joystick [8].

To display information for the driver, the simplest (unless seeing the robot directly, or being on the robot) way is to use a computer or TV screen. However, more immersive solutions using a screen exist (such as immersive chamber (CAVE) [8][9], see Figure 3.1, or an immersive desk [10][11]). User is like in a bubble with the impression to see what the robot sees. Nevertheless the immersion is only visual, this technique is *only* an improved video game, even though the robot is real.

Figure 3.1: Cave automatic virtual environment (picture by Davepape)

3.2.2 Only BCI

The immersion in the robot is generally limited (here, only robots with BCI are considered) to the EEG headset [12]. The interface to control the robot generally uses three types of schemes, which they are called:

- *real thoughts* [12];
- *brain eye-tracking* [13];
- *facial expression* [14].

The first scheme will be used here and described in section 3.3.2. To achieve the second one, the interface displays several buttons (up, down right, left generally), which blink at its own frequency. Thanks to this large panel of frequencies, the user's gaze can be known and found in the EEG graphs. The drawbacks of this technique are the obligation for the user to keep his gaze looking at the screen, and, therefore, the impossibility to see the robot. The third option uses winking or frowning for example. These movements, especially winking and blinking, are easily detectable and can be quickly used. Nevertheless, this interface is not very *user-friendly*, because it is not a natural way for human to interact with their environment. Indeed, winking and frowning are only used for *implicit* communication, and support the speech.

3.2.3 Multi-Human Computer Interface

Other HCI are used, in parallel with BCI, in some projects ; to display information and to improve the control. Keeping in mind, the goal of the project of this report consists in using gyroscope informations, displaying robot's gaze in glasses, and hearing what the robot hears.

Some similar studies have been done, such as in Embry-Riddle Aeronautical University, Daytona Beach, Florida, USA [15]. Their goal is to provide a helpful solution in hazardous environment, like in Fukushima-Daiichi nuclear power plant. Hence, their robot didn't need to be social, but only useful. Moreover, several film-cameras around the chassis are placed and when the user turns his head, only the user's field of vision moves, and not the robot. Therefore, a person in front of the robot cannot know where the *driver* is staring.

3.3 Brain activity

3.3.1 Denoising and signal processing

In order to remove the noise generated by muscles, eyes or other, the signals must be processed. Noise has to be removed from signals before the classification, (see section 3.3.2) and to detect facial expressions. Several algorithms exist each one with their advantages and their drawbacks. The difficulty is to make the difference between valuable data and the noise, especially for the eyes. Indeed, in this application, some eye movements are valuable informations such as blink and frown contrary to eyeball movement. Signal processing, as described in this part, concerns only the facial expression detection. The classification can be considered as signal processing, and is described in the next section.

Several algorithms can be highlighted:

a) Wavelet transform

The wavelet theory implies a decomposition of the signal like in Fourier Transform. However, the wavelet base is not sinusoidal. And, last but not least, the wavelet transform allows an analysis in frequency **and** in time.

Several wavelet decomposition exists, like the discrete wavelet transform (DWT) or stationary wavelet transform (SWT) used in [16]. They use SWT to remove the ocular artefact in the EEG signal; with the advantage of not needing other measures like EOG or EMG. Moreover, the computational requirements are low and allow real-time processing. The method uses, with the SWT, an adaptive thresholding mechanism. At the beginning of the experiment, the thresholding reference is recorded when the subject is still. Then, the algorithm uses the maximum amplitude of EEG signals and the SWT coefficient to detect artefact.

Another team uses also the HAAR wavelet transform (HWT) for denoising [17]. They use the HWT to detect the exact moment when the subject blinks in order to remove it. They use an high order HWT in this case.

b) Independent component analysis (ICA)

What is an usual technique for blind source separation (BSS), especially when several signals are mixed[1] [18]. This algorithm is widely used in the EEG signal processing. It based on the hypothesis that all sources are non-Gaussian. The aim of this algorithm is to find the *separating matrix*. Let s be the valuable signal, A be the *mixing matrix* ($\simeq$ noise) and x be the noisy signal:

$$x(t) = As(t) \tag{3.1}$$

then, the *separating matrix* W is:

$$a(t) = Wx(t) \tag{3.2}$$

The denoising operation is successful if $a(t) = s(t)$ (or as close as possible)[2]. Once separated, the noisy signals (occular or muscle movement) are known to obtain noise-free EEG data. Many different ICA methods exist.

[1] A textbook case is the *cocktail party problem*.
[2] This was the simplest model.

c) Adaptive filter

In the outlines, this filter consists in a linear filter which has a transfer function controlled by variable parameters. Some research use adaptive filter in BCI world [19][20]. The first case is based on a neural network function. A neuron is created for each artefact pattern detected in order to remove it. In the second case, they use the adaptive filter to remove the ocular movements.

d) Regression analysis

works on the assumption that several sources are superposed (brain, eyes, muscles...). In order to remove the perturbation, the regression analysis uses other recordings like gyroscope, eye-tracking, EOG or EMG. The impact of these different sources is estimated and they are removed.

The drawback of this method is the external recording requirement. This can be impossible or very constraining.

e) Spatial distribution of the rhythmicity

uses the principle of neighbourhood [21]. If a perturbation occurs in an EEG channel, then, the closest channels are analysed. If the same perturbation doesn't occur in the other channels, then, it is certainly a perturbation. Indeed, if the *perturbation* comes from the brain, it doesn't affect only one electrode.

This algorithm doesn't work for all perturbations, but it is effective, for example, for electrodes slip.

f) Signal processing

Besides the extraction of the main characteristics of the curve — such as the amplitude, variation rate, sign, etc. — several signal processing methods must be used (Fourier transform, filtering, etc.). These process are mainly used to denoise the signal or to extract some features, *e.g.* pattern recognition.

g) Epoching

In order to achieve real-time signal processing, the signal is regularly cut in several parts. Each part is called an epoch. Thanks to this strategy, the *unreal-time* algorithms, such as Fourier Transform, can be applied on the epoch.

3.3.2 Classification

Interpretation of thoughts is not easy, even for simple orders such as *hand up* or *walk*. It is easier to detect a brain activity of real movement, but, with training, the program can recognise movement thought. Even with a fMRI, that has yet a higher spatial resolution than EEG, the thought detection is difficult. The most used technique with BCI is the classification. However, the training requirement is a drawback of this method. Indeed, the subject needs to *open his mind* to the computer and do several exercises to find a pattern.

For example, the subject is asked to raise his hand several times. The classification algorithm is applied on a feature (see below) of the data to make a pattern out. The features are the EEG properties that are extracted, like amplitude values, band power, Power Spectral Density values, (Adaptive) AutoRegressive parameters, etc. [1].

Several classification methods exist, some of them are less fitted than other. Here is a non-exhaustive list.

a) Linear Discriminant Analysis (LDA)

as its name suggests, is a linear method. It is one of the most popular methods for classification in the BCI world. This method uses an hyperplane to separate the feature vector(s) into two classes (see Figure 3.2) [22][1]. To place the hyperplane, the distance between the two class means should be maximized and the interclass variance minimized. The *one versus the rest* (OVR) strategy is used to separate the data in several classes. The OVR strategy creates two classes, for each iteration: A_i and $\bar{A}_i$, with A_i the hoped class, $\bar{A}_i$ the complement and i the class number.

The LDA advantages are a low computational requirement, the easiness to process and good results. The drawback is a inefficiency for complex non-linear data [23].

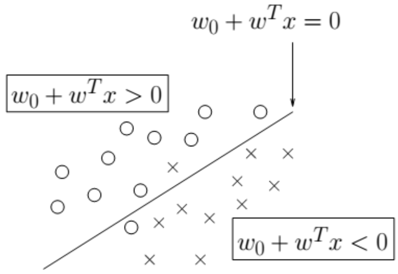


Figure 3.2: LDA: The hyperplane separates the *circles* and the *crosses* into two classes (image from [1])

b) Support Vector Machine (SVM)

is also a linear method which uses a hyperplane. But this time, the plane is chosen to maximize the margins which are the distance from the nearest training points (Figure 3.3) [24][25][26]. The SVM is made for linear systems, but it can be easily changed to non linear problems.

The advantages are to have good generalisation properties, to be insensitive to overtraining and the curse-of-dimensionality[3]. The drawback is a low computational time.

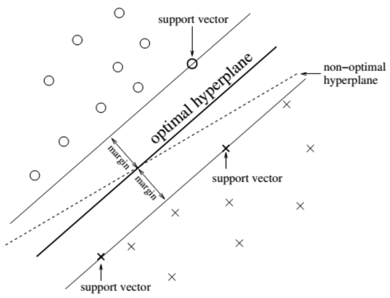


Figure 3.3: SVM: The optimal hyperplane separates the *circles* and the *crosses* in two classes (image from [1])

c) Neural Network (NN)

is an algorithm which mimics the brain, with several *neurons* working together [27]. They are connected together and receive data from one neuron. Then, they send the result of their *work* to another one. Several algorithms using NN are used in BCI like Multilayer Perception, which is one of the most used; the Gaussian classifier, especially made for the BCI; Learning Vector Quantization (LVQ) NN; Fuzzy ARTMAP NN; and so on.

[3]It is a *statistical problem* that appears when the number of training data is small compared to the size of the feature vectors. To avoid this problem, five to ten times more training sample than feature vector number are needed.

d) Nonlinear Bayesian classifiers

like Bayes quadratic algorithm or Hidden Markov Model (HMM) [24]. Use the probability that the feature vector belongs to the class. However, this type of algorithm is not often used in BCI applications.

e) Nearest Neighbour Classifiers (NNC)

as its name suggests, this kind of algorithms uses the nearest neighbour to assign a feature vector to a class. Manhalanobis distance and k Nearest Neighbour can be cited as NNC.

However, these algorithms are not very popular for BCI classification [1].

f) Classifier combination

in order to increase the global algorithm performance, several classifiers can be used according to three strategies:

Boosting which consists in using several classifiers in cascade. The classifiers focus on the errors doing by the previous ones [24].

Voting which consists in on the one hand in using several classifiers in parallel and on the other hand choosing the class that obtains the majority.

Stacking which consists in using several classifiers in parallel and sending their results to a meta-classifier that does the final choice.

3.4 Head motion

The Emotiv gyroscope, it is then in fact an accelerometer (see chapter 4), measures the head acceleration on two axis: vertical and horizontal. Consequently, if the user moves his head with a constant speed, the gyroscope will not detect the movement. It is a limitation of the system.

The goal is to determinate the user's head movements. Not only directions, up and down or right and left, but also acceleration. Thanks to acceleration, the speed of the head movement can be known. This information is useful to adapt the robot angular velocity.

The extraction of characteristics from gyroscope signals is similar to EEG ones. Fortunately, the gyroscope signals are less noisy than EEG ones. The main valuable feature is the amplitude of the first variation. The sign is strongly important to determine the direction.

Gyroscope is often used as an internal sensor. Indeed, it is used in control systems engineering, especially in the feedback loop. It is a simple sensor to know the robot orientation, in addition other sensors. For instance, a study has been made on a cleaning robot [28]. The cheapest robot uses only an odometer as feedback sensor. However, this method is unreliable, odometer informations can be quite erroneous when the robot follows a curved path. The gyroscope is used with the odometer and a constrained Kalman filter. Gyroscope is often used in biped walking robot to stabilise them [29][30]. Like the vestibular system in the human body, the gyroscope allows the robot to be stable.

3.5 The sight

To reinforce the immersion *in* the robot, the sight is essential, even more with stereoscopic vision. VRHMD works generally with a side-by-side video. To do that, different configurations are used:

- use one video-camera and use algorithm(s) to recreate the 3D (the two images);
- use two cameras and put the videos together;
- use a 3D camera, which does internally the second solution.

Obviously, the third solution is the easiest, but the 3D cameras are expensive, around €1,000. Some cheaper cameras exist, but their functionalities are poor.

The first solution needs a fast algorithm to achieve a 2D-3D conversion in a real-time basis [31]. This type of algorithm is often used in 3D television [32]. The advantage is the requirement of only one camera. But the drawbacks are numerous. Firstly, the computed 3D model cannot be better than the real 3D model (2nd and 3rd methods). Secondly, these algorithms have difficulties to work with static images [31]. Thirdly, some depth distortions can appear (like first plan becoming second plan), and, finally, delay the image processing. This last point can be crippling, indeed, the real-time can vanish and make the user feel sick.

The second solution is a tradeoff between the two others. The cost of two separate film-cameras is generally lower, *2D camera* is a reliable technology. However, an issue is raised: the temporal simultaneity. Indeed, the two cameras must be synchronised in order to avoid delays between left and right images.

3.6 The hearing

Sound is an important sense, moreover with spatial perception. Indeed, if the user can know where the sounds are coming, the immersion is total. Several sound formats exist like 5.1, 6.1, etc. These formats use several loudspeakers, it is restrictive and need an equipped room. These equipments cannot be easily carried and each person who uses the IVCS would need a room for himself. Indeed, they could not be in the same location. Moreover, the recording needs more than two microphones to receive the different sound sources. But a simpler solution exists: binaural recording.

This technique uses two microphones surrounded by auricle [33]. The *artificial ears* must be placed as they would be on a real human head (distance, position, orientation...). This technique uses the brain capacity to rebuild a sound spatialisation.

Three parameters are used for that [34]:

Intensity variation: the intensity of the sound source allows to know if the source is near or far. The intensity is inversely proportional to the distance.

Time delay: the sound doesn't arrive in the same time in each ear. Thanks to this delay, the brain knows if the sound comes from the left or the right.

Spectral variation: the auricle shape add some spectral variation in the sound. Since the earliest childhood, the brain learnt this variation. This extra datum informs the brain of the vertical position of the source.

The two channels must be heard separately by each ear. The simplest way is to use an ear- or headphone.

3.7 Summary

In the time allotted for this internship, it was chosen to focus on the detection of user's facial expressions (blinks) using EEG data; and on the robot control with the Emotiv's accelerometer. The developed algorithms to that purpose are explained in the chapter 5. That is the main achievement of this internship with the communication program and the robot control one.

The proposed solution fits into the social robotics. Indeed, this immersive robot allows elderly people, far persons or disabled people to communicate as if they are there. Moreover the proposed scenario brings a new approach for the communication. Nowadays, in the videoconference, the point of view is fixed. If something happens behind the camera, the user is blind. Thanks to the robot, the user will be able to turn *his* head. And not only turn his head, but *walk* too.

To put it in a nutshell, this system allows persons with reduced mobility to be close of other people. Moreover, Figure 3.4 summarises the principle.

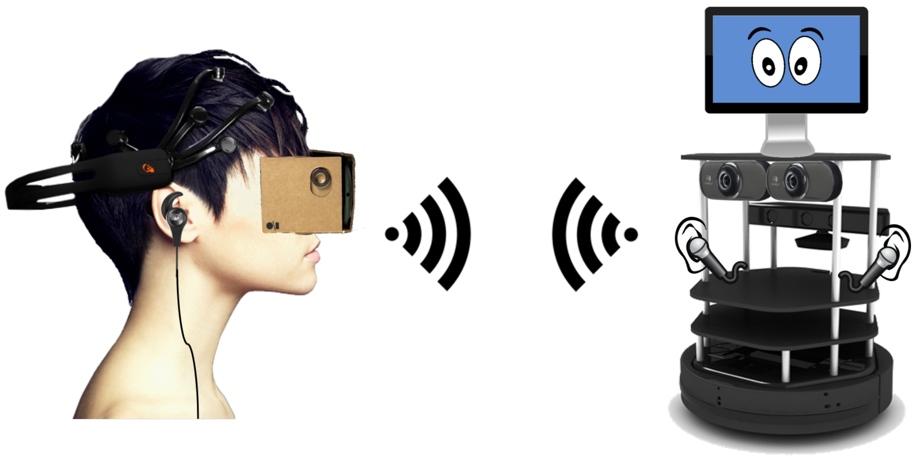


Figure 3.4: General principle

Chapter 4

Material and communication

To realise the project described section 3.1, several tools are used , in this part, each one will be detailed as well as the developed program to communicate between the robot and the EEG headset.

4.1 Material

4.1.1 The robot

The project uses a Turtlebot (see Figure 4.1) [5]. The robot is managed by the Open Source Robotics Foundation, Inc., however, it is an open-hardware and open-source project, everyone can participate. The philosophy is to build a low-cost robot for everyone. The robot's base is a Kobuki base that looks like a robotic vacuum cleaner. The robot is planar, it can move only on two axis (x and y) thanks to two wheels. The robot also has several security devices :

- Bumper switches ;
- Lifting sensors.

Figure 4.1: Turtlebot (image from Turtlebot's website)

The robot is made to work with the Microsoft Kinect sensor but it is not an obligation. The Kinect and the robot are connected to a laptop. This computer is the robot's brain and contains the software.

The turtlebot's operating system is ROS (for Robot Operating System)[35], a free OS[1] specially built for robots. ROS is managed by the same foundation as the Turtlebot.

4.1.2 The EEG headset

The aim of the project is to get the EEG data from the brain. For this purpose, the University provides an EPOC EEG headset from Emotiv society. It is an affordable and multi-platform EEG headset [36]. The headset has 14 data electrodes and 2 reference electrodes. The location of the electrodes is based on the 10-20 system, but the choice of the electrode is specific to Emotiv. Indeed, some electrodes are present only in 10-10 system, see the Figure 4.2.

In addition EEG, the headset has a two-axis *gyroscope*. The system is described as such in the datasheet, whereas in reality, it is more of a two-axis accelerometer. The data are sent by Bluetooth to the computer. In order to improve the electrode conductivity, they need to be humidified by a saline solution.

The software development kit (SDK) bought by the University works on Linux. Several utility softwares are supplied. The most useful allows the user to see the different brain signal graphs.

[1] Under BSD licence

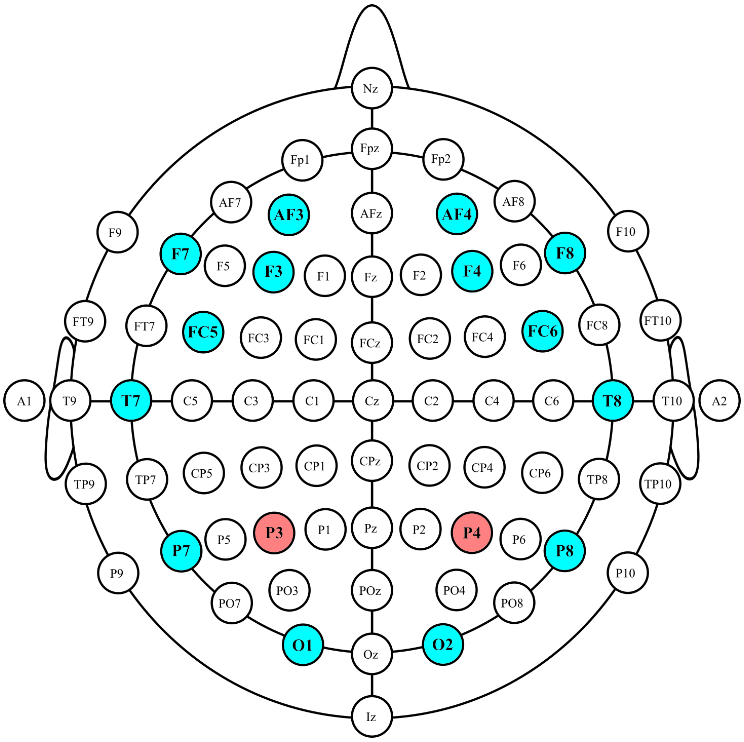


Figure 4.2: Electrode localisation with Emotiv headset (adapted from Marius 't Hart's picture CC BY-SA 3.0)

All the following reasons tip the scales in Emotiv headset favour, its price, its ease-to-use, and the combination of the EEG and gyroscope sensors. Moreover, the OpenVIBE software (see below) can work with Emotiv.

More specifications of the Emotiv headset are detailed on the annexes page 41.

4.1.3 OpenVIBE

OpenVIBE [37] is an open-source freeware made by INRIA, the French Institute for Research in Computer Science and Automation. This software is especially made for BCI signal processing. The interface looks like the Matlab's toolbox Simulink. Each function is represented by a box, and the user links the boxes together to build the algorithms (see Figure 4.3). The software can read the data from the Emotiv headset directly, and from the gyroscope thanks to the SDK. Consequently, the Linux version of OpenVIBE is used.

Some boxes exist to realize signal processing, or classifications such as LDA, SVM or Fourier transform. Besides, the software is open-source, hence, a nonexistent box can be coded. The processing can be achieve in real-time and data can be sent to another software thanks to VRPN library.

OpenVIBE is used to get the Emotiv's data and display the raw signal in order to check the process. Simple process are achieved by OpenVIBE, the more complex process are done by the **TurtlebotCommunication Client** (see below) that has been created during this internship. Indeed, although OpenVIBE allows to create boxes, some problems have been encountered in interfacing mathematical libraries. These problems prevented the signal processing, that's why the raw signal is directly sent to the **TurtlebotCommunication Client** program.

4.2 Communication

The project is made in two parts in the C++ programming language:

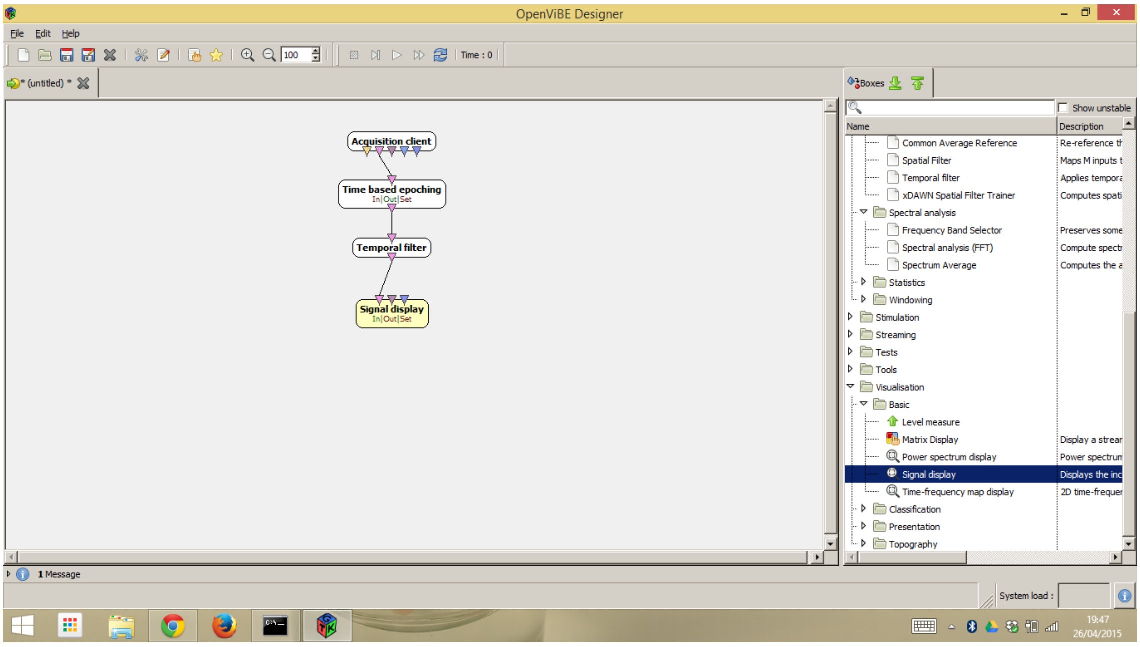


Figure 4.3: Window of the designer of OpenVIBE

- The first one is on the turtlebot's computer which receives the command and leads the robot.
- The second one is on the user's computer. In our study, the computer is connected to an ad hoc network with the first computer. However, in the final product, both have to be connected to the Internet. This computer receives the EEG data. These data are processed then the commands are created and sent to the first computer.

Figure 4.4: Qt's logo (© The Qt Company)

François, Yann and I worked together on the communication with the robot and its control, these two points are the shared part of the internship. It consists on the realization of a communication software (see below), and the control, as for it, is ROS software. It commands the robot thanks to an home-made PID. The program receives a point (or a trajectory) and with its position and the PID, it computes the motor commands.

In order to communicate between the two computers, two home-made programs have been built. The **TurtlebotCommunication Server** on the turtlebot's computer and the **TurtlebotCommunication Client** on the user's computer. Theses programs use the Qt library (more precisely, a framework) to lay the GUI (Graphical User Interface) out[38]. Qt facilitates the window construction. Indeed, the interface is standard for each operating system without modification of the source code. The programmer has only to work on the program's core. For the signal processing, the Armadillo library is used[39]. It is a free linear algebra library, its syntax is deliberately similar to Matlab.

The programs communicate together thanks to the TCP (Transmission Control Protocol) protocol. It is, with UDP (User Datagram Protocol), one of the most used low-level protocol. However, contrary to UDP, the packets (the data are cut in small parts to facilitate the communication) come in order. In addition, TCP handles errors, in case of mistransmission the lost packets are sent again. That is why this protocol has been chosen. Qt has a TCP library at its disposal, and the same philosophy as the GUI is applied, which makes things easier. Programmer only have to deal with transmitting the message. The message is built as follows (see also Figure 4.5):

1. The 16 first bits are kept for the command size. It is the size in bit and not in number of character.
2. The command is the second part of the message. It is the valuable data, and the command

Figure 4.5: The sent message by TCP

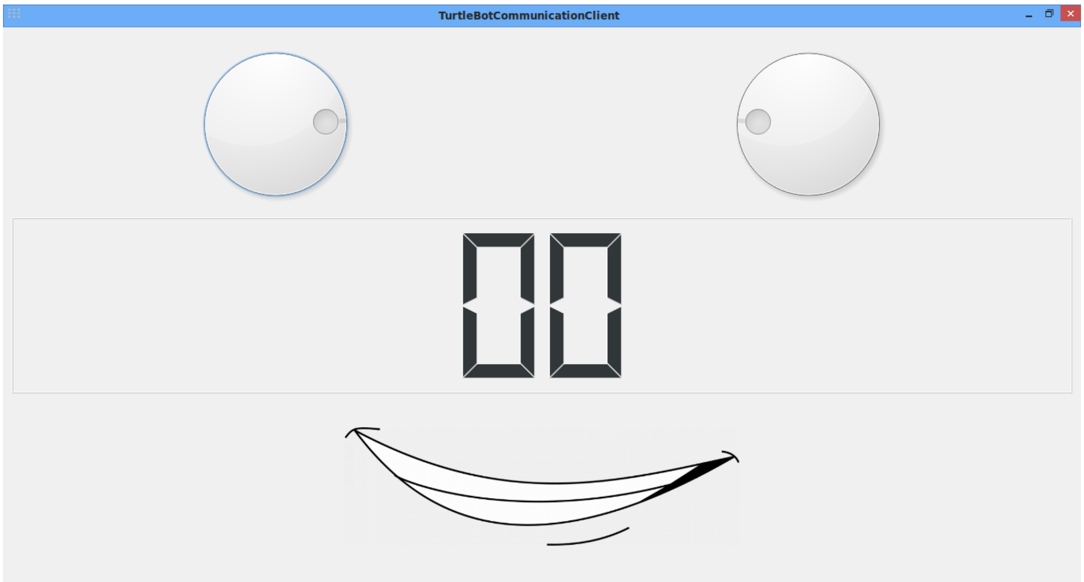


Figure 4.6: The eye interface (on Linux)

is always built like this `CODE_DATA`. The code is used by the program in order to know the data's purpose and how to manage them; the code works like a railroad switch.

4.2.1 The client

The EEG data are sent to OpenVIBE which uses the Emotiv driver. Some preprocessing is applied (low-pass filter) on the signal. Then the OpenVIBE VRPN (Virtual-Reality Peripheral Network) library is used to send the data to the client. Next, the Client applies the algorithm described in Chapter 5. Finally, the commands are created and transmitted to the Server.

Some informations are displayed to the user: a terminal displays the sent command in order to check the process. For the same purpose, a graph displays the data after the processing. Finally the eyes are displayed (see below). See Figure B.1 in the appendices page 41.

4.2.2 The server

Once data are received, they are steered thanks to the code then the data are processed. In order to manage the robot, a shared memory is implemented between the Server and the ROS program. It works like a shared variable. The shared memory works as a simplex communication: the Server changes the variable values and the ROS program reads them; except for one bit which is an acknowledgement bit.

As for the client program, a terminal is displayed, but the most important is the eyes interface. Indeed, this interface (see Figure 4.6) is displayed in full screen on the Turtlebot's computer for the user's interlocutor.

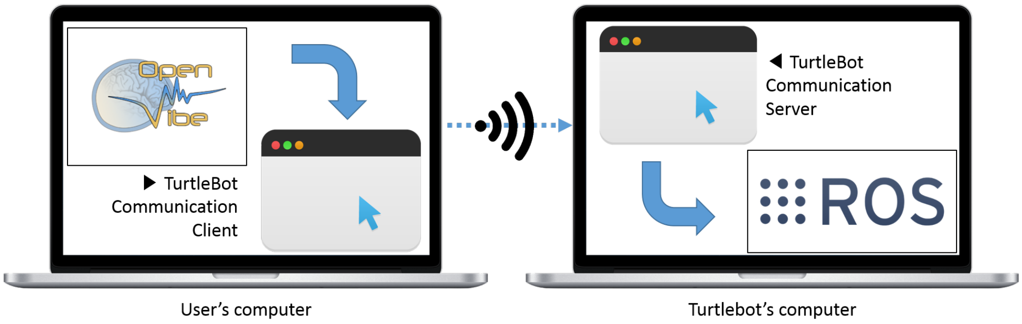


Figure 4.7: The project's communication

Three bands can be outlined, the first one, with the two dials, represents the eyebrows. When the user frowns, the dials (or knobs) move to mimic the eyebrows. The left one dials to the maximum and the right one dials to the minimum.[2] The second one, with a two digit seven-segment display, represents the eyes. When the eyes are opened, 00 is displayed. If the user blinks, - - is displayed. The third one, with the smile, has for only purpose to create a more user-friendly interface.

4.3 Summary

To put it in a nutshell, two computers are used, one for the user and one for the robot. The EEG headset is connected to the user's computer with the OpenVIBE software installed. The robot's computer leads the robot's movements. Moreover different software have been developed, two for the communication and one to control the robot. See Figure 4.7 that sums up the process.

[2]The Microsoft Windows' dial style is closer to eyebrow than the Linux style. In order to be more realist, the window style cloud be changed.

Chapter 5

Data extraction

The data sent by the EEG are very noisy and have to be processed in order to extract useful informations. In this chapter, the created algorithm to extract the blinks will be described. The algorithm for the extraction of the head's movements will also be described.

5.1 Gyroscope

A double accelerometer is inside the Emotiv headset (called gyroscope in the documentation) that allows to detect the head's movements according to the x and y axis.

According to the project charter introduced section 3.1, this sensor is used to measure the head's horizontal movement (at first) in order to move the robot in the same way. The purpose is to simulate the fact to be in the robot thanks to the HMD.

The gyroscope sends a very clean signal without noise (see Figure 5.1). Because it is an accelerometer, to generate a signal, the user has to move his head with a jerk. With a constant speed, the signal is null. The signal's amplitude gives an information on the angle: if the user turns his head strongly, the robot has to turn with a bigger angle. Moreover, the direction of variation gives a indication on the movement's direction (left or right).

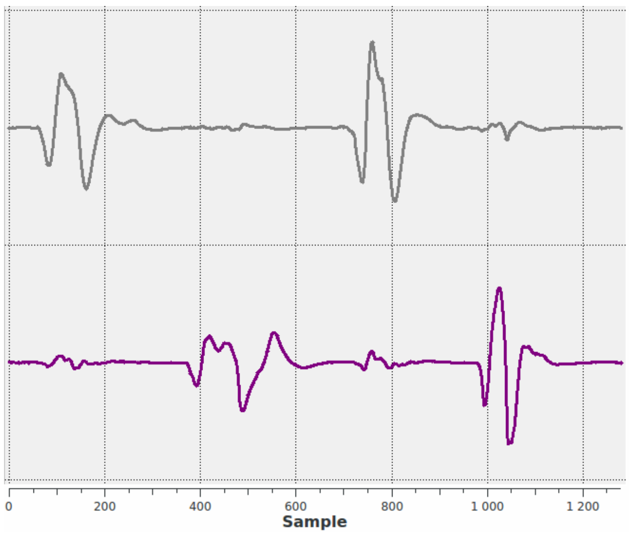


Figure 5.1: The gyroscope's signals (horizontal and vertical)

A simple method has been chosen in order to detect the movement, the sign of the biggest amplitude gives the movement's direction. And the value of this maximum gives the rotation speed. In order to get the exact moment to move the robot and avoid multiple commands (because of the several peeks, see Figure 5.1), a Schmitt trigger is used. This component is well known by

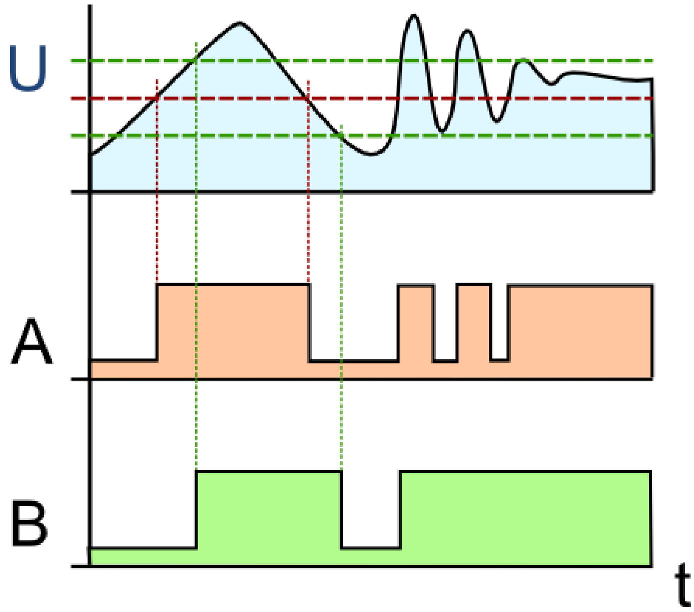


Figure 5.2: Comparison between a simple comparator and a Schmitt trigger (image by FDominec CC BY-SA 3.0)

electricians. It is a double threshold comparator or an hysteresis comparator (see Figure 5.2). This component avoids toggling too quickly and at an untimely moment. Thanks to that, the robot receives only one command for each head movement. The threshold value is adjusted to have, approximately, the same angle between the robot and the head.

The Schmitt trigger functioning is very simple. The output signal of the trigger is binary (in this case, 0 or 1). The 0 value is chosen as initial value because it is equivalent to “no movement”; the 1 value is equivalent to “move”. Their are two thresholds, an “high threshold” and a “low threshold”, symmetrical around 0, for instance 200 and -200. When the signal goes over the high threshold, the output becomes 1 and the robot turns. The output is equal to 1 while the signal is taller than the low threshold. Thus, there may be fluctuations around the high threshold, the output will not fluctuate, (watch the difference between the red output (simple comparator) and the green one (Schmitt trigger) on the Figure 5.2).

The Schmitt trigger benefit is its very simple functioning, it can be easily programmed (with two conditions), hence, no delays are added in the command chain. This is very important in this application. To be the more immervise, the robot has to react immediately.

Unfortunately, the Turtlebot has a delay of one second between the command sending and the movement execution, consequently, the movement is not a real time movement.

The keyboard is used to move the robot forward and backward. The keyboard is used instead of the EEG because it was decided to focus on the processing of eye movement. Indeed, in some pieces of literatures, it is notices that extract order from the brain take lot of time and ask lot of training. The internship length was only 21 weeks, thus it focus only on the eye movement detection, in order to get results at the end of the internship.

5.2 EEG

5.2.1 Introduction

The aim of the EEG data extraction is to display the user’s eye actions (blinking, frowning, movements...) on the robot. The eyes are a full-fledged social *device*. Indeed, nearly all emotion use the eye-eyebrow couple, see Figure 5.3 to identify few of them.

The eyebrow position and the eye opening are the main characteristics of expressions: for

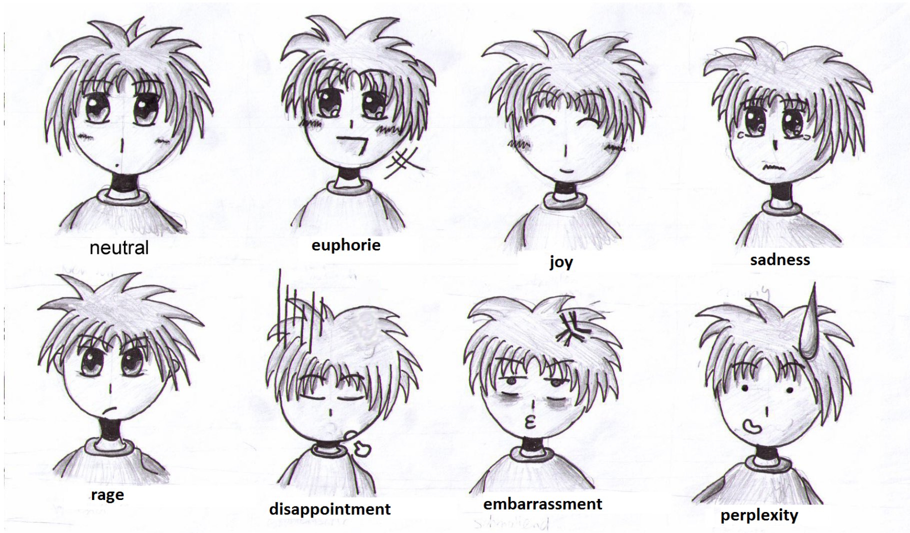


Figure 5.3: Several facial expressions (adapted from Sympho's picture CC BY-SA 3.0)

example, when people are angry, they half close their eyes and they frown. If they are surprised, they open big eyes and they raise eyebrows. To show one's connivance with someone or to share a secret, the most common expression is winking; once again, eyes are used. That's why this project will focus on the winking and the frowning. It may be noticed that another *parameter* could be taken into account: the pupil's position (and by extension the glaze). However the Emotiv's is not accurate enough for this purpose.

In order to improve the algorithm's accuracy, only the six electrodes closest of the eyes will be chosen, *i.e.* F7, F3, AF3, AF4, F4 and F8 (see Figure 4.2).

One can be noticed that OpenVIBE does not really work in real-time. The values are not sent point by point, but in a packet called epoch. In this project, the epoch is composed of 128 points. Each epoch is processed after the previous one.

5.2.2 The developed algorithm

The developed algorithm during this internship has for goal to detect blinking and frowning. Several operation are executed like filtering, *unmixing* and matching. The Figure 5.4 shows the algorithm with a block diagram.

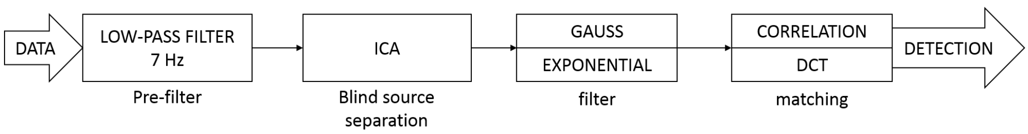


Figure 5.4: The blink and frown detection algorithm

Before any operation, a low-pass filter (a simple 1st order Butterworth filter) with a cutoff frequency of 7 Hz is used. Indeed, eye movement signals are mostly located in the delta and theta bands (see section 2.2.2 page 4), and the range of these bands is [1; 7 Hz] [40].

5.2.3 Independent source analysis

In order to extract the eye actions (blinking, frown, movement...), the ICA algorithm (more exactly *fastICA*) is used to *sort* the data. More specifically, to realise a blind source separation (BSS). Indeed, the EEG measures a mixed signal from some part of the brain but also from the muscles,

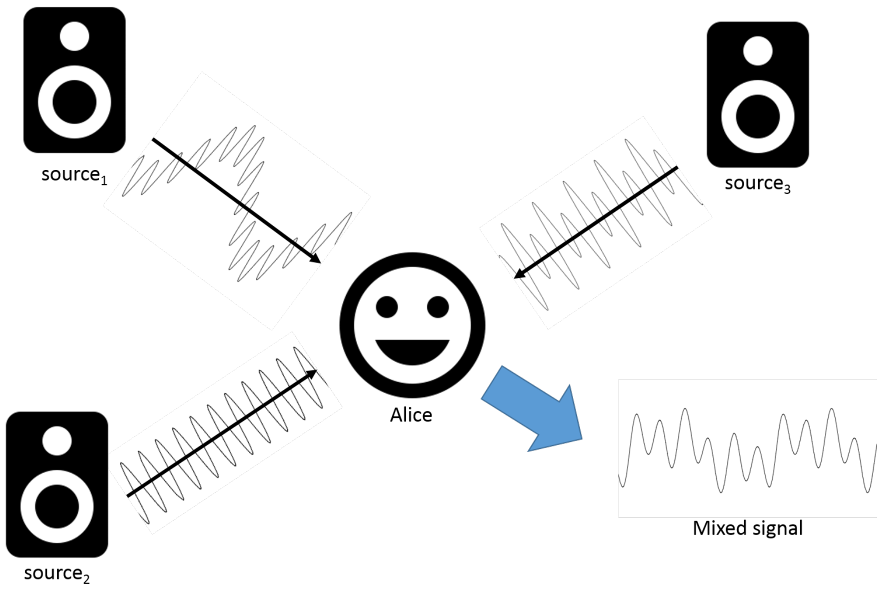


Figure 5.5: Cocktail party problem

and from the eyes. This last source is valuable in our case, contrary to all EEG studies. In order to get the eye signals (EOG) from the raw data, the ICA will be useful.

The most famous BSS textbook case is the cocktail party problem. In a cocktail party, each person is a sound source and without brain concentration, people hear a brouhaha: all sources are mixed. In the Figure 5.5 the sources are represented by a sound speaker and Alice is a cocktail party's member. She hears a mix of each source with a weighting :

$$Alice \quad = \alpha source_1 + \beta source_2 + \gamma source_3 \tag{5.1}$$

To work, the BSS needs at least as much sources as measures. So, let us add two more peoples at the party: Bob and Carol. They have to be at different positions, to change the source weightings else, one would have three identical equations[1]. Now, the equations are:

$$Alice \quad = \alpha source_1 + \beta source_2 + \gamma source_3 \tag{5.2}$$
$$Bob \quad = \delta source_1 + \varepsilon source_2 + \zeta source_3 \tag{5.3}$$
$$Carol \quad = \eta source_1 + \theta source_2 + \lambda source_3 \tag{5.4}$$

And, with a thorough notation:

$$\underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} \alpha & \beta & \gamma \\ \delta & \varepsilon & \zeta \\ \eta & \theta & \lambda \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}}_{S} \tag{5.5}$$

The aim is to find the $(s_1\ s_2\ s_3)^T$ values from $(x_1\ x_2\ x_3)^T$, hence the $W = A^{-1}$ matrix has to be found.

A parallel can easily be drawn between the cocktail party and the brain: the sound speakers represent a part of the brain, the eyes or the muscles and Alice, Bob or Carol represent an electrode from the headset. For this project, only one source is valuable: the EOG. The goal of the ICA algorithm is to extract the left eye and right eye signals and leave all the brain and muscle waves in one signal. That's why, in this study the S vector has 3 rows.

[1] The weighting value can be easily understandable: if the person is close to the source, the coefficient is close to one. On the contrary, if the person is far of the source, the coefficient is close to zero

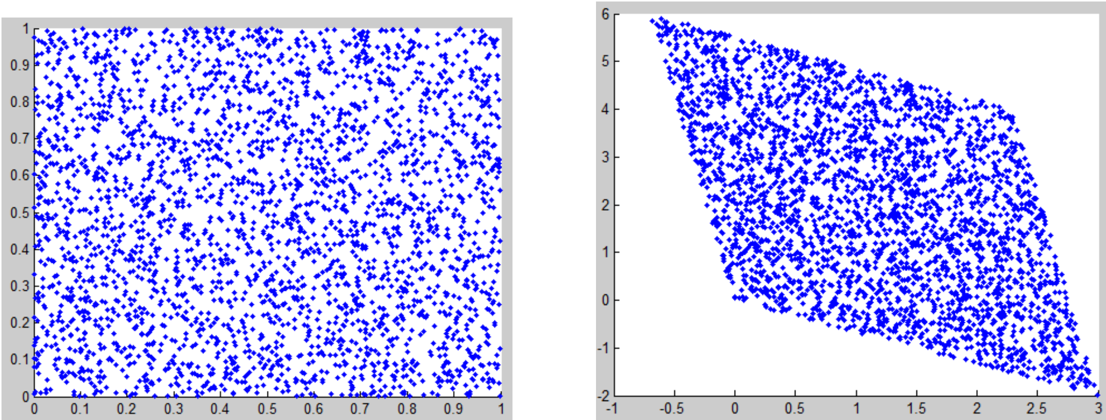
The ICA is a BSS method using the non-Gaussianity principle. For this reason, only non-gaussian sources can be used with this method. Furthermore, some pre-processing has to be done.

Let S be the sources matrix and X be the mixed signals matrix. To illustrate let s_1 and s_2 be two random vectors representing the sources and x_1 and x_2 the mixed signals such as

$$x_1 = 3s_1 - 0.7s_2 \tag{5.6}$$

$$x_2 = -2s_1 + 6s_2 \tag{5.7}$$

See Figure 5.6 for the illustration. On the Figure 5.6a, one can see that the data are non-gaussian, contrary to Figure 5.6b which is closer to gaussian distribution.



(a) $s_2 = f(s_1)$ (the sources) (b) $x_2 = f(x_1)$ (the mixed signals)

Figure 5.6: Illustration of the BSS

a) Averaging

The first step of preprocessing before the ICA is the averaging. The signal should be mean-zero, in order to accelerate the algorithm. if that is not the case, the signal mean must be subtracted to the signal:

$$X_m = X - \overline{X^T} \cdot \vec{1}_M^T \tag{5.8}$$

with X the $(N \cdot M)$ raw data matrix. N is the number of electrodes and M the number of samples. $\vec{1}_K$ is a column vector (of M rows) full of 1.

b) Whitening

The second step is the whitening in order to uncorrelate the signals. The algorithm proposed by Hyvärinen *et al.* [41] (the fastICA algorithm creators) uses the eigenvalues and eigenvectors for this purpose:

$$X_w = ED^{-1/2}E^T \cdot X_m \tag{5.9}$$

with E the eigenvector matrix and D the diagonal matrix of the eigenvalues.

After these two steps, the example becomes the Figure 5.7. The result is squarer, and closer to the sources. Then one *only* needs to turn the square to restore the original. The criterion to stop the *rotation* is the gaussianity. Indeed, the assumption was the non-gaussianity of the source and to obtain the source signals from the whiten signals, the gaussianity has to be reduced. One can notice that the proportions are kept but not the values. But in this project, only the proportions are important.

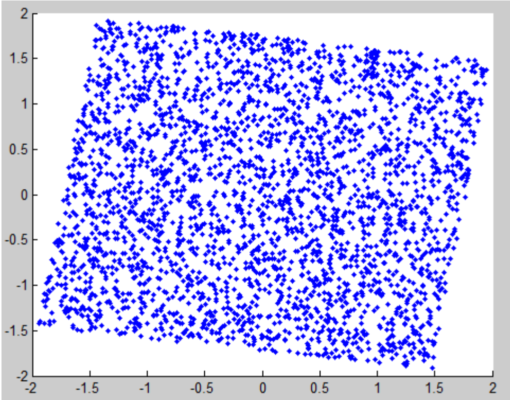


Figure 5.7: $x_{w2} = f(x_{w1})$

c) ICA

After the preprocessing, the ICA algorithm can be used. As explained previously, the the aim of ICA is the *ungaussianisation*. Two criteria may be used to measure the gaussianity of the signal: kurtosis and neguentropy. An approximation of the second one will be used.

In order to find the unmixed signals S (the sources) from the mixed measures X, the mixing matrix A should be found: $X = AS$. In the case of unmixing the signal, it is the unmixing matrix A^{-1} (called W) that should be found:

$$S \approx Y = WX \tag{5.10}$$

with Y the approximate matrix of the sources.

The ICA algorithm has to be used as much times as the number of sources, in this case three times. The algorithm finds a unit vector $\mathbf{w}$ such that the projection $\mathbf{w}^T\mathbf{x}$ maximizes the nongausianity. Then, the variance $\mathbf{w}^T\mathbf{x}$ should be constrained to unity, with withened data, this is equivalent to constraining the norm of $\mathbf{w}$ to be unity. These operations are repeated until convergence, *i.e.* when the dot product between the old and new values is equal to 1 (plus or minus epsilon). The algorithm in pseudocode is given below.

for p = 1 to numberOfSource
  $w_p \longleftarrow$ random(1, numberOfSample)
  do
    $w_pOld \longleftarrow w_p$
    $w_p = \mathrm{E}\left[X_w g\left(w_p^T X_w\right)^T\right] - \mathrm{E}\left[g'\left(w_p^T X_w\right) w_p\right]$
    $w_p = w_p - \sum_{j=1}^{p-1} w_j^T w_p^T w_j^T$
    $w_p = \dfrac{w_p}{\|w_p\|}$
  while $||\langle w_p \cdot w_pOld\rangle| - 1| \leq \varepsilon$
  $W(p,:) = w_p$
for end
S=W*X

Figure 5.8: FastICA algorithm

With g' the derivative of g; g is the derivative of a nonquadratic function. For example g can be equal to:

$$g_1(u) = \tanh(a_1 u) \tag{5.11}$$

$$g_2(u) = u\exp(-u^2/2) \tag{5.12}$$

where $a_1 \in [1;2]$, the most common value is 1. In this study $g = g_2$. E represents the mean function.

Line 6 should be used when there are more than one source, to avoid that all vectors converge to the same value; that is a decorrelation operation.

Now, the S matrix contains three vectors which correspond, theoretically, to the left eye, to the right eye and to the rest. In order to be sure to have the two eye signals and to identify them, a correlation is applied between the two electrodes closest to the eyes. AF3 for the left and AF4 for the right. After this step, six coefficients are gotten, for example:

	Left	Right
Source #1	0.9	0.3
Source #2	0.4	0.01
Source #3	0.5	0.7

Here, the first source is probably the left eye, the second corresponding to the rest and the third one to the right eye. If the maximum coefficient for each eye is for a same source (*e.g.* the first one), then the eye with the greatest coefficient (*e.g.* the left one) will correspond to this source (here, the first one). The other eye (in this example, the right one) will take the source with the second greatest coefficient (*e.g.* the second source). For example, with these values:

	Left	Right
Source #1	0.9	0.7
Source #2	0.4	0.3
Source #3	0.5	0.1

The left eye is the first source and the right eye will be the second one. Thanks to this method the probability to make a good separation is higher. In addition, to improve once again the method, the W matrix is kept if and only if the higher correlation coefficients are superior or equal to a threshold. The higher the threshold is, and the closer to the reality the model is. The W matrix is recomputed until the threshold is reached.

5.2.4 Filtering

Then, the data have to be filtered to erase the noise and keep a good signal shape. For this purpose, two filters have been developed and they are compared in the Chapter 6: a Gauss and an exponential filter. Two low-pass filters. Some others filters have been tested, as a shape detection filter or a first order low-pass filter, but these filters prove to be less efficient than the Gaussian filter or the exponential one. Gaussian and exponential filters have been kept in order to compare them.

a) The Gaussian filter

The Gaussian filter is a filter impulsing response is a Gaussian function (or an approximation to it). It is considered the ideal time domain filter, just as the cardinal sine function is the ideal frequency domain filter [42]. The Gaussian filter modifies the input signal by convolution with a Gaussian function. In the case of a digital filter (as here), the Gaussian function is approximated in computing several values and in storing them in an array.

Two variables define the filter, σ (standard deviation) that represents the bell's width (therefore the height) and x the filter's size (the number of value). This filter makes an *average* with a Gaussian weighting. Indeed, the number of value is odd and the central value corresponds to the Gaussian function's peek, *i.e.* the highest value. The around values have a lowest weighting hence an average due to weaker impact of the around values. The Gaussian filter slides on the input sigma until the output signal is complete (*i.e.* they have the same size), see Figure 5.9.

In addition, side effects have to be managed, there are two ways in this algorithm:

- for the first epoch, zeros are added before the signal ($x - 1$ zeros);
- for the other epochs, the last values of the previous one are added before the signal ($x - 1$ zeros);

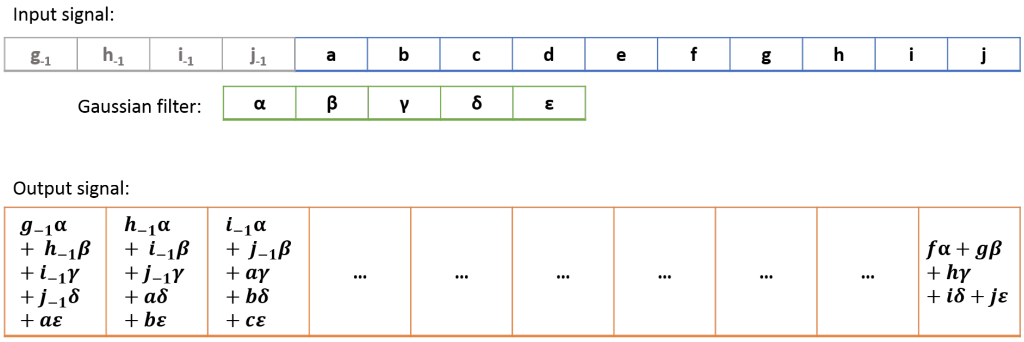


Figure 5.9: Gauss filter principle (gray cells represent the previous values)

Thanks to the side effect management, all the signal values are processed in the same way.

After several tests with a file (therefore not in real-time), it emerged that the sigma value must be high (between 20 and 30). The results is almost without noise (see Figure 5.10) and each eye movements can be easily extracted.

b) The exponential filter

Also called exponential smoothing, can be easily set up and does not require lot of processing resources; this filter uses only a recursive relation. This filter executes a weighted average with decreasing exponential weight over time.

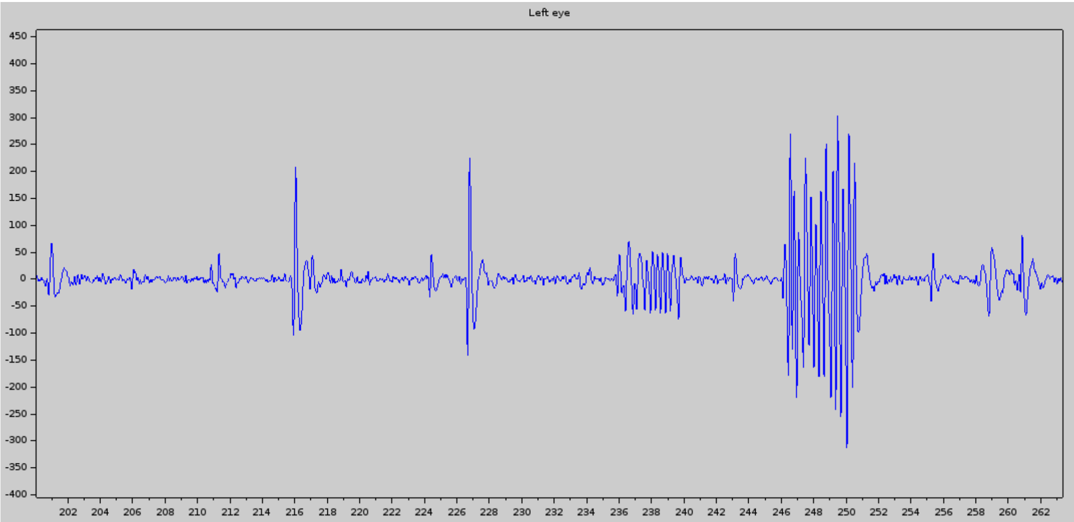
This filter is commonly used to smooth data and act as a low-pass filter in removing the high frequencies. The exponential smoothing is part of the moving average family (under the name of exponentially weighted moving average (EWMA) or Autoregressive integrated moving average (ARIMA) (0,1,1) model with no constant term). The exponential filter has some advantages compared to others MA, as simple moving average (SMA). This filter makes a simple average on the k last values of the signal (the choice of k influences on the filtering, if k is great, the output will be smoother, and *vice versa*). One disadvantage of this technique is that it cannot be used on the first $k-1$ terms of the time series without the addition of values created by some other means. It also introduces a phase shift into the data of half the window length. For example if the data were all the same except for one high data point, the peak in the smoothed data would appear half a window length later than when it actually occurred. Another filter filter is the weighted moving average (WMA), it is the same as SMA except with a linear weighting. The drawbacks are more or less the same as for SMA.

The first description of the exponential window function given by Siméon Denis Poisson in the 17th century. Exponential smoothing was first suggested in the statistical literature by Robert Goodell Brown in 1956, and then expanded by Charles C. Holt in 1957. The filter is simply defined by this recurrence relation:

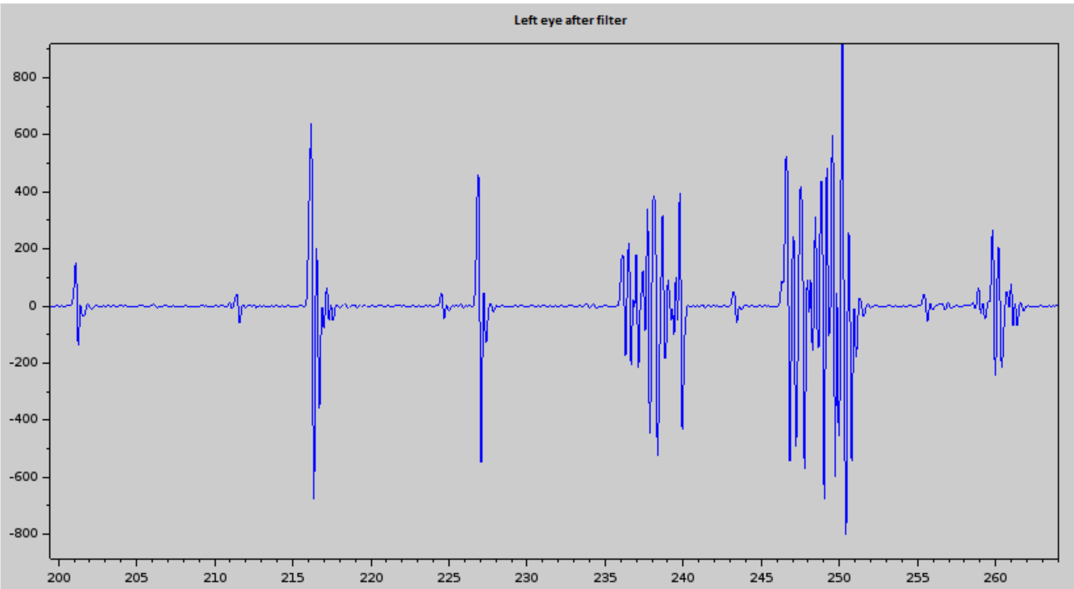
$$s_0 = x_0 \tag{5.13}$$

$$s_t = \alpha x_t + (1-\alpha) s_{t-1}, \; t > 0 \tag{5.14}$$

with α the smoothing factor ($\alpha \in]0;1[$). This filter is a weighted average between the current value (x_t) and the previous smoothed value (s_{t-1}). For the values of α close to 1, the output signal is less smoothed (and for $\alpha = 1$ the output is equal to the input) and for the values close to 0, the signal is strongly smoothed. Simple exponential smoothing is easily applied, and it produces a smoothed statistic as soon as two observations are available, contrary to the SMA or WMA. However the accuracy of the filter is better when some values have been processed. In practice a *good average* will not be achieved until several samples have been averaged together.



(a) Signal from ICA before Gaussian filter



(b) Signal from ICA after Gaussian filter

Figure 5.10: The Gaussian filter effect

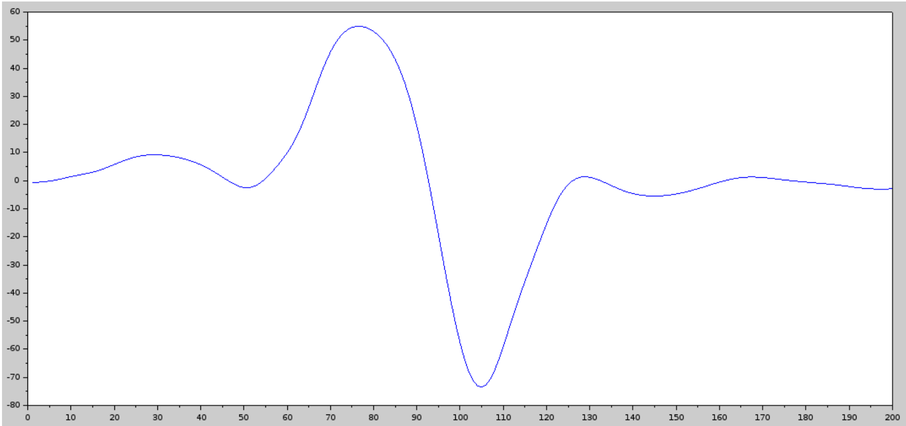


Figure 5.11: The blink signal

This filter is called *exponential* because an exponential window is used during convolution:

$$\begin{aligned} s_t &= \alpha x_t + (1-\alpha)s_{t-1} \\ &= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2 s_{t-2} \\ &= \alpha \left[x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + (1-\alpha)^3 x_{t-3} + \cdots + (1-\alpha)^{t-1} x_1\right] + (1-\alpha)^t x_0 \end{aligned}$$

Over time, the smoothed signal s_t becomes the average of a greater and greater number of past observations. The weight becomes closer and closer to the geometric progression (equivalent to the discrete version) of the exponential function: $1, (1-\alpha), (1-\alpha)^2, (1-\alpha)^3, ...$ That's why this filter is called exponential.

Double and triple exponential filters exist but, the double is used when a trend is in the data and the triple when their are a periodic signal. There are no trend and no periodic signal in the data, consequently, a simple exponential filter is enough.

The exponential filter has been chosen because it is a really fast filter and very efficient to smooth signal, that is the aim of the filtering.

5.2.5 Matching

The aim of this part is to detect the moment where the user blinks or frowns. After the ICA, the signal is composed of two channels, one for each eye, thus one can get the blink (or frown) for each eye. The blink is easily recognizable on the filtered signal even with naked eye. The blink's shape signal is always identical to the Figure 5.11. It is interesting to notice that the frown signal is also the same, except for the value of the amplitude which is greater (twice or thrice). In order to find the user's blink, two methods have been chosen and will be compared: the correlation and the DCT.

a) Correlation

More exactly, cross-correlation, is a measure of similarity between two signals thus a pattern recognition. For discrete signal, as here, the formula is:

$$(f \star g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m]\, g[m+n] \tag{5.15}$$

with f^* the complex conjugate of f.

The cross-correlation is also called *sliding dot product*, indeed, the Figure 5.12 illustrates this name. If the f function is the signal and the g function the pattern. The pattern is *slides* on the signal in order to compute the dot product and get the $(f \star g)$ value, *i.e.* the correlation coefficient.

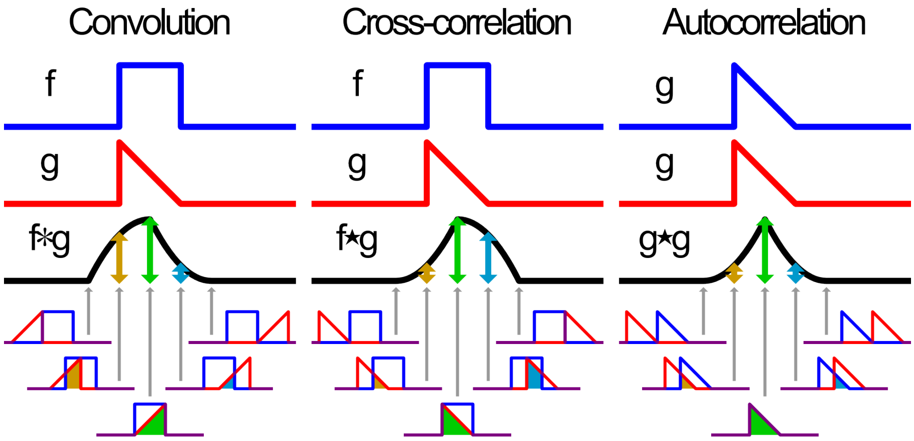


Figure 5.12: Visual comparison of convolution, cross-correlation and autocorrelation (image by Cmglee CC BY-SA 3.0)

In the case of pattern recognition, if it is found in the signal, the cross-correlation is closer to an autocorrelation (correlation between the signal and itself). Closer the signal is to the pattern, and closer the correlation coefficient is to 1.

The cross-correlation is a very simple way for patter recognition, and it is a low computational-complexity algorithm. That is why, it has been chosen. For the test, the correlation coefficient threshold is a parameter, the aim is to find its best value. The correlation coefficient threshold is the value from which the software detects a blink.

b) DCT

The Discrete Cosine Transform (DCT), is a transform near the discrete Fourier transform (DFT). The DCT creates from a signal a set of real coefficients, thanks to a cosine sum. The DCT is very used for lossy compression (*e.g.* for audio signal). Indeed, instead of keeping all the points of a signal, one keeps only a handful of coefficient. They are enough to rebuild the signal (an approximation). More one has coefficient and one can be closer to the original signal. This is the compromise between quality and economization of resources (space and computing performance).

In this case, the DCT is used to compare two signals. The pattern to be recognized (*e.g.* blinking signal) is computed by the DCT in order to get its coefficients, eight in this project. They are stored in a file, which avoids calculate these coefficients each time. Then, the signal itself is computed by the DCT (in real-time) and the signal's coefficients are compared to the pattern ones (with an euclidean norm).

the formula for the DCT is as follows:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \tag{5.16}$$

With N the number of point, n the current point and k the coefficient index. Thus, it is possible to choose the number of coefficients. This value is the first parameter tested in the chapter 6. Then, the euclidean norm is apply to the coefficient:

$$s = \sqrt{(X_1+P_1)^2+(X_2+P_2)^2+\cdots+(X_k+P_k)^2} \tag{5.17}$$

With P_i the ith DCT coefficient of the pattern. The second parameter is the value of the threshold (that is compared to the euclidean norm, the s value), that determines when a blink occurs.

This method is also ungreedy and quick to implement, especially as half the calculation is not realized in real time.

5.3 Summary

There are two algorithms, one for the head's movement and the other one for the brain signal.On the one hand, the accelerometer signal is processed with a Schmitt trigger in order to know when to command the Turtlebot and how (direction and speed). On the other hand, the brain signals are processed by several *blocks*:

1. A low-pass filter with a cutoff frequency of 7 Hz;
2. The ICA algorithm that give the left and right eye signals, only when the algorithm converged;
3. The filter (Gauss or exponential) in order to erase the noise;
4. The matching (correlation or DCT) in order to extract the blinking.

The two filters and the two matching methods are compared. The purpose is to know which combination is the best and which parameters have to be used.

Chapter 6

Results

6.1 Experimental protocol

In order to test the algorithm results, the following protocol have been achieved. Several brain records have been recorded by the user. Theses records last 1 minutes. During this time, the user has to blink each 5 seconds (10 times per record). The records last only one minute because it is weary for the user to blink deliberately, avoiding his natural blinking. Several files have been recorded, by two users.

The aim is to count the blinks detected by the algorithm in order to find the better configuration. One chosen to fix the filter values in order to avoid having too many parameters to manage. Therefore, the values of the Gaussian filter are $\sigma = 20$ and $x = 59$, the value for the exponential filter is $\alpha = 0.01$. These values have bee chosen after several experiments on non-real-time file on Scilab. They was the best values for filtering.

Then, for each filter, the following parameters have been tested:

Correlation threshold: 0.601 ; 0.614 ; 0.624 ; 0.65 ; 0.702 ; 0.75

DCT (3 coefficients): 200 ; 250 ; 260 ; 280 ; 300

DCT (5 coefficients): 200 ; 250 ; 260 ; 280 ; 300

Consequently, their are 32 experiments per file.

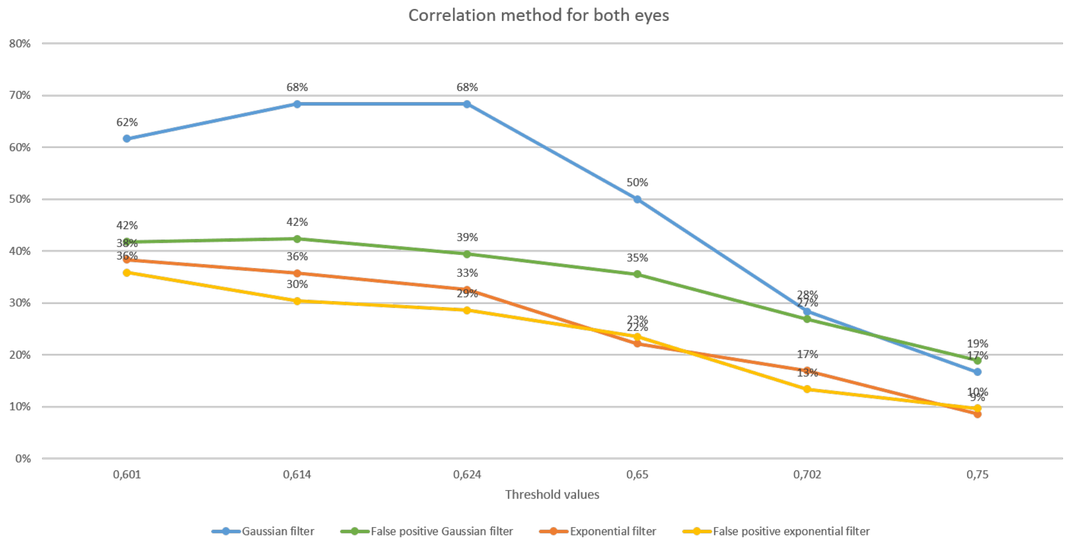
It can be noticed, the results are less good than in normal situation. Indeed, here, the W matrix is not computed until the correlation coefficients are close to one, the first computed matrix is used. Because this operation can take several seconds, even a minute. That is too much for the records. The matrix is not necessarily the best, but remained the same for tests on the same file.

6.2 Experiment

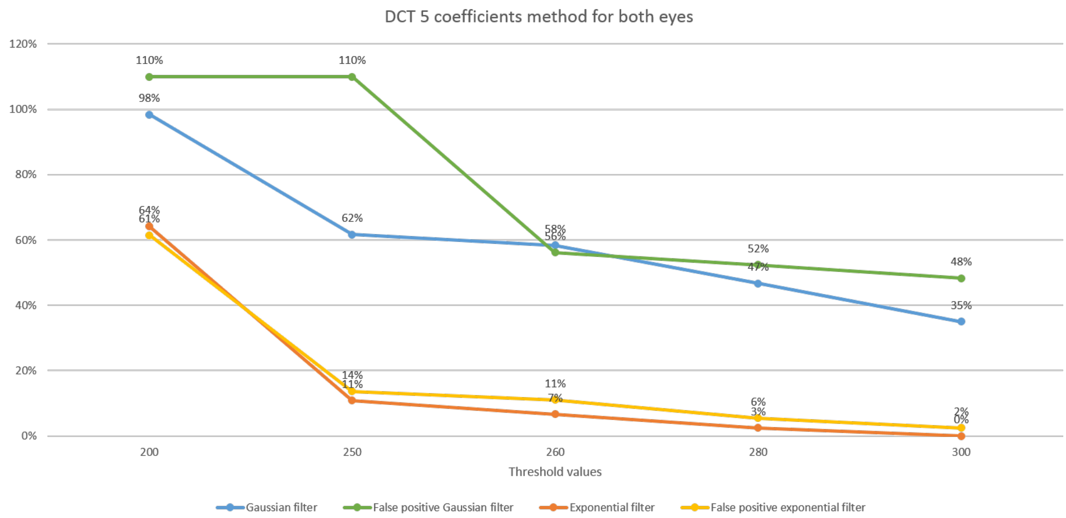
6.2.1 Filter comparison

Firstly, the filters are compared on each matching method (correlation and DCT) for the both eyes. In this case, the blink is taken into consideration as detected if at least one eye blinked. For the *false positive* (a blink has been detected, but the user didn't blinked his eyes), if the value is 110%, so there was too many of them to be countable. The DCT with 3 coefficients has been ruled out because the algorithm detects no blink. The results are given in the Figure 6.1. The cold colours represent the Gaussian filter and the warm ones the exponential filter. For the blue and orange curves, the percentage represents the number of blink detected. For the green and yellow ones, the percentage represents the proportion of *false positive*.

Excluding the first two Gaussian filter values with DCT, which have too many *false positive*. One observed Gaussian filter get better results than the exponential one. Indeed, the Gaussian filter maximum value is 68% (with correlation method and a threshold of 0.624). The exponential filter gets its maximum value, 61%, with the DCT algorithm and a threshold of 200. Although



(a) Comparison with the correlation method



(b) Comparison with the DCT (5 coefficients) method

Figure 6.1: Comparison of the two filters on the both eyes

these two values are close, the number of *false positive* is no comparison between the Gaussian filter and the exponential one: 39% for the first one and 64% for the second.

This first step of comparing allowed to choose the Gaussian filter to the rest.

6.2.2 Matching method comparison

The matching method are compared with the same filter, the Gaussian one. For this test, the eyes have been considered separately. During the recording, the user had to blink his **both** eyes every 5 seconds, thus two blinks should be detected. The *false positive* rate is displayed again ; see Figure 6.2. The cold colours represent the left eye and the warm ones represent the right one.

Once again, a point has too many *false positive*, it is dismissed (left eye in DCT and a threshold of 200). After an examination of the graphs, one can observe that the correlation method gives better results than DCT. The positive rate is greater for the correlation generally. When the DCT's rate is high, it is a the cost of a high rate of *false positive*. It can be noticed that the left eye is globally better detected as the right one. In light of the tests, the better configuration is the

Gaussian filter with the correlation and a threshold of 0.601.

It can be noticed that the *false positive* rate remains quite high, between 40 and 60%, to have good detection rate (greater than 60%). If the eyes are not distinguished; the better configuration is the correlation with a threshold of 0.624 (always with the Gaussian filter). The *false positive* rate is "only" of 39%. Keeping in mind that the W matrix is not ideal and can increase the *false positive* rate. It can also be an explication of the difference of detection between the left and right eye. The correction of this defect is a part of the future works, see section 7.1.

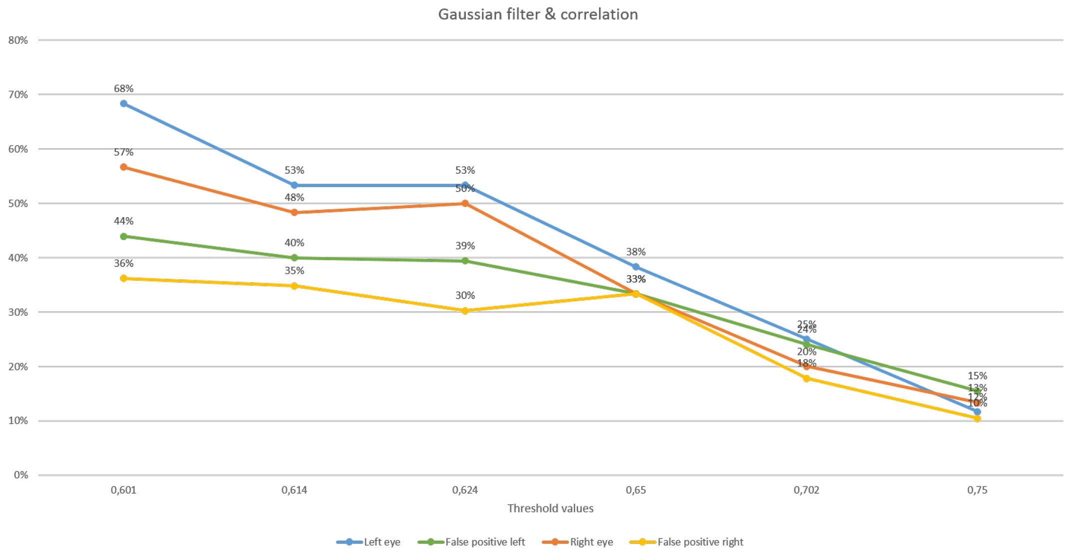
6.3 Pictures of the results

The Figure 6.3 shows the control of the robot by the Emotiv's accelerometer. The computer is used to move the robot forward and backward. The Figure 6.4 shows the eye interface on the robot, when the eyes are opens and when they are closed (a blink is detected).

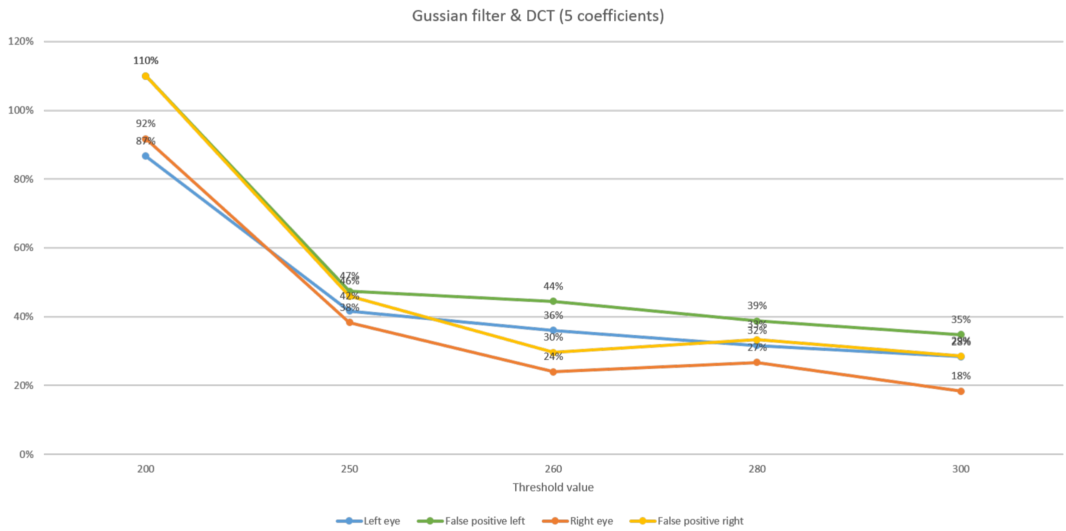
6.4 Summary

The Gaussian filter seems to be the best compared to the exponential one. The Gaussian filter's *positive* rates are high and the *false positive* ones are lower than for the exponential filter. The correlation, with a threshold around 0.600, seems to be better than DCT.

However, the *false positive* rates remain quite high and have to be reduced. A solution could be the combination of the two matching methods, in cascade or in parallel.



(a) Left and right eyes with correlation and Gaussian filter



(b) Left and right eyes with DCT and Gaussian filter

Figure 6.2: Comparison of the two matching method on each eye

Figure 6.3: Test of the control by the accelerometer

(a) The robot with open eyes

(b) The robot with closed eyes

Figure 6.4: The program in action

Chapter 7

Conclusion

The part on the detection of eye movements, as well as the control of the robot by the Emotiv's accelerometer have been achieved during this internship. The project answers to the *social robotics* problematic of the subject: **Social Robotics: Human-Robot Interaction Using EEG Signals and Head Motion**. Indeed, detecting the eye movements allows to transcribe the user emotions again. The robot and the EEG are used for this purpose, the robot have to become *a second body*.

The developed algorithm demonstrated that it could detect the majority of the blinks, with a relatively low rate of *false positive*. So, the announced goal of displaying the user's blinks on the robot's screen is achieved. This assessment could be qualified. Indeed, the *false positive* rate stills high to reproduce the user's mimics faithfully. A way to correct this problem is to use several matching methods, in parallel or in cascade.

This method is not enough efficient to detect precisely the pupil movements, and get the user's gaze position. Only *big* movements as blinks and frowns can be detected. Moreover the signal quality from the Emotiv is not very good and, contrary to a hat EEG, the electrodes' position is less accurate.

The purpose of this algorithm, and this internship, is to fit into the improved videoconferencing system (IVCS) invented section 3.1. The IVCS could allow disabled or elderly people to have the impression of moving while sitting.

7.1 Future work

Several things can be executed to improve the system. Like making the differentiation between the right and the left eye better, or improve the distinction between frown and blink.

Furthermore, hardware could be upgraded, especially by using a real gyroscope instead of an accelerometer. Because, with an accelerometer the user has to shake his head to move the robot instead of only turn his head. Moreover, another robot could be used in order to suppress the Turtlebot's delay (1 second). The utilisation of an humanoid robot could have a stronger immersion, whether for the user or his interlocutor. The integration of the binaural microphones would also be better on a humanoid robot.

Change the EEG headset could allowed to use the EGG as an basic eye-tracking system and display this information on the robot. Indeed, the gaze is an important social interaction. Knowing where people are steering is useful. For example to know who is the *speech recipient*.

It would have been interesting with more time and resources to develop the rest of the ecosystem, like the sight and the hearing. But also the control of the robot with brain.

Bibliography

[1] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and . A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4, 2007.

[2] ShuzhiSam Ge and MajaJ. Matarić. Preface. *International Journal of Social Robotics*, 1(1):1–2, 2009.

[3] R. Qiu, Z. Ji, A. Noyvirt, A. Soroka, R. Setchi, D.T. Pham, S. Xu, N. Shivarov, L. Pigini, G. Arbeiter, F. Weisshardt, B. Graf, M. Mast, L. Blasi, D. Facal, M. Rooker, R. Lopez, D. Li, B. Liu, G. Kronreif, and P. Smrz. Towards robust personal assistant robots: Experience gained in the srs project. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1651–1657, Oct 2012.

[4] Emotiv. http://emotiv.com/. Accessed: 2015-04-29.

[5] Turtlebot. http://www.turtlebot.com/. Accessed: 2015-04-29.

[6] Mikhail A. Lebedev and Miguel A.L. Nicolelis. Brain–machine interfaces: past, present and future. *Trends in Neurosciences*, 29(9):536 – 546, 2006.

[7] Saeid Sanei. *EEG Signal Processing*. Wiley-Interscience, 1st edition, 2013.

[8] J. Fournier, M. Mokhtari, and B. Ricard. Immersive virtual environment for mobile platform remote operation and exploration. In *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, pages 37–42, Sept 2011.

[9] A. Sandygulova, A.G. Campbell, M. Dragone, and G.M.P. O'Hare. Immersive human-robot interaction. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pages 227–228, March 2012.

[10] Jinki Moon, Youngwan Son, Shinsuk Park, and Jinwook Kim. Development of immersive augmented reality interface for construction robotic system. In *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pages 1192–1197, Oct 2007.

[11] N.P. Jouppi, S. Iyer, Wayne Mack, A. Slayden, and S. Thomas. A first generation mutually-immersive mobile telepresence surrogate with automatic backtracking. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1670–1675 Vol.2, April 2004.

[12] Jd.R. Millan, F. Renkens, J. Mourino, and W. Gerstner. Noninvasive brain-actuated control of a mobile robot by human eeg. *Biomedical Engineering, IEEE Transactions on*, 51(6):1026–1033, June 2004.

[13] Brain computer interface used to control the movement and actions of an android robot. http://www.diginfo.tv/v/12-0199-d-en.php, 2012. Accessed: 2015-04-13.

[14] E.J. Rechy-Ramirez and Huosheng Hu. Bi-modal human machine interface for controlling an intelligent wheelchair. In *Emerging Security Technologies (EST), 2013 Fourth International Conference on*, pages 66–70, Sept 2013.

[15] J. Fletcher, C. Fountain, C. Kerley, J. Torres, W.C. Barott, S. Salamah, and R. Stansbury. Immersive audio-visual system for an unmanned robotic vehicle in a senior design project. In *Southeastcon, 2012 Proceedings of IEEE*, pages 1–5, March 2012.

[16] Xinyi Yong, Mehrdad Fatourechi, Rabab Ward, and Gary Birch. Automatic artefact removal in a self-paced hybrid brain- computer interface system. *Journal of NeuroEngineering and Rehabilitation*, 9(1):50, 2012.

[17] N.V. Kalpakam and J.S. Sahambi. A novel wavelet based technique for detection and denoising of ocular artifact in normal and epileptic electroencephalogram. In *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, volume 2, pages 1027–1031 Vol.2, June 2004.

[18] Stephen Roberts and Richard Everson. *Independent Component Analysis: Principles and Pratice*. Cambridge University Press, 2001.

[19] Aysa Jafarifarmand and Mohammad Ali Badamchizadeh. Artifacts removal in {EEG} signal using a new neural network enhanced adaptive filter. *Neurocomputing*, 103(0):222 – 231, 2013.

[20] Md. Khademul Islam Molla, Md. Rabiul Islam, Toshihisa Tanaka, and Tomasz M. Rutkowski. Artifact suppression from {EEG} signals using data adaptive time domain filtering. *Neurocomputing*, 97(0):297 – 308, 2012.

[21] A.M. Skupch, P. Dollfuß, F. Fürbaß, M. Hartmann, H. Perko, E. Pataraia, G. Lindinger, and T. Kluge. {EEG} artifact detection using spatial distribution of rhythmicity. *{APCBEE} Procedia*, 7(0):16 – 20, 2013. The 3rd International Conference on Biomedical Engineering and Technology - {ICBET} 2013.

[22] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, Inc., second edition, 1990.

[23] G.N. Garcia, T. Ebrahimi, and J. Vesin. Support vector eeg classification in the fourier and time-frequency correlation domains. In *Neural Engineering, 2003. Conference Proceedings. First International IEEE EMBS Conference on*, pages 591–594, March 2003.

[24] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., second edition, 2001.

[25] ChristopherJ.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[26] Kristin P. Bennett and Colin Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explor. Newsl.*, 2(2):1–13, December 2000.

[27] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. CLARENDON PRESS - OXFORD, 1995.

[28] Hyun Myung, Hyoung-Ki Lee, Kiwan Choi, Seok-Won Bang, Yong-Beom Lee, and Sang-Ryoung Kim. Constrained kalman filter for mobile robot localization with gyroscope. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 442–447, Oct 2006.

[29] Y. Okuyama, A. Yabu, and F. Takemori. Attitude control of a biped walking robot model with circular arced soles using a gyroscope. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1379–1384 vol.2, May 1998.

[30] T. Sugihara and Y. Nakamura. Gravity compensation on humanoid robot control with robust joint servo and non-integrated rate-gyroscope. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 194–199, Dec 2006.

[31] T. Iinuma, H. Murata, S. Yamashita, and K. Oyamada. 54.2: Natural stereo depth creation methodology for a real-time 2d-to-3d image conversion. *SID Symposium Digest of Technical Papers*, 31(1):1212–1215, 2000.

[32] L. Zhang, C. Vazquez, and S. Knorr. 3d-tv content creation: Automatic 2d-to-3d video conversion. *Broadcasting, IEEE Transactions on*, 57(2):372–383, June 2011.

[33] Raphaël Greff. *Binaural holophony - Sound spatialization over circumaural transducer arrays.* Theses, Université Pierre et Marie Curie - Paris VI, November 2008.

[34] Erwan Lecomte. L'incroyable expérience du son 3d. http://www.sciencesetavenir.fr/high-tech/20130430.OBS7693/l-incroyable-experience-du-son-3d.html, 2013. Accessed: 2015-04-24.

[35] Robot operating system. http://www.ros.org/. Accessed: 2015-04-29.

[36] Emotiv epoc brain computer interface and scientific contextual eeg. http://emotiv.com/product-specs/Emotiv2014. Accessed: 2015-04-14.

[37] Y Renard, F Lotte, G Gibert, M Congedo, E Maby, V Delannoy, O Bertrand, and A Lécuyer. Openvibe: An open-source software platform to design, test, and use brain computer interfaces in real and virtual environments. *Presence*, 19(1):35–53, Feb 2010.

[38] Qt library. http://www.qt.io/developers/. Accessed: 2015-08-29.

[39] Armadillo library. http://arma.sourceforge.net/. Accessed: 2015-08-29.

[40] Chi-Hsuan Hsieh, Hao-Ping Chu, and Yuan-Hao Huang. An hmm-based eye movement detection system using eeg brain-computer interface. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 662–665, June 2014.

[41] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411 – 430, 2000.

[42] Herman Blinchikoff and Helen Krause. *Filtering in the time and frequency domains.* The Institution of Engineering and Technology, Stevenage, 2001.

Appendix A

Carddiff University

This internship took place in the Cardiff University (*Prifysgol Caerdydd* in Welsh), more precisely, in the Mechanics, Materials and Advanced Manufacturing department (MMAM). The University is located in the center of the Wales' capital.

The University was founded in 1883 as The University College of South Wales and Monmouthshire. The University is one of the top university in the United-Kingdom and in 200 best universities in the world.

The MMAM department is led by Prof Rossi Setchi who supervised this internship. The MMAM's research theme incorporates cutting edge research, which fosters innovation and sustainability, supports social and economic development, and contributes to improvements in health and quality of life. Composed of over 30 researchers, the institute is recognised as being one of the top engineering research centres in the UK. Research is conducted within a vibrant environment, which includes world-class laboratories in additive layer manufacturing, micro/nano manufacturing, metrology, tribology and structural performance.

Figure A.1: Cardiff University, the Queen's Building

Appendix B

Material and communication

B.1 Material

B.1.1 Emotiv characteristics

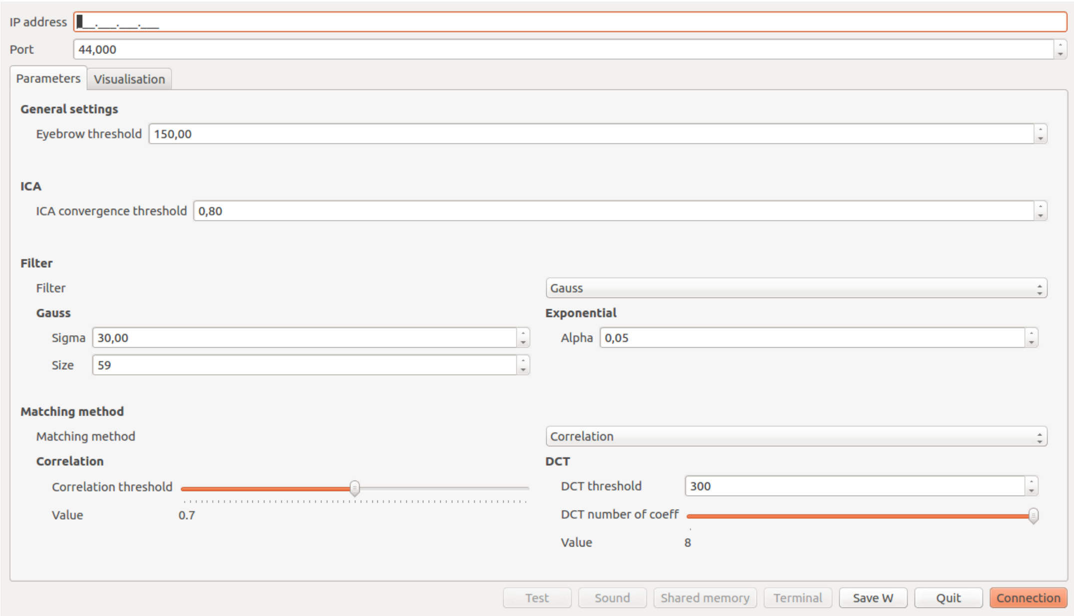
Characteristics	Value
Number of channels	14 (plus CMS/DRL references, P3/P4 locations)
Channel names (International 10-20 locations)	AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4
Sampling method	Sequential sampling. Single ADC
Sampling rate	128 SPS (2,048 Hz internal)
Resolution	14 bits 1 LSB = 0.51 µV (16 bit ADC, 2 bits instrumental noise floor discarded)
Bandwidth	0.2 - 45 Hz, digital notch filters at 50 Hz and 60 Hz
Filtering	Built in digital 5th order Sinc filter
Dynamic range (input referred)	8,400 µV (pp)
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4 GHz band
Power	LiPoly
Battery life (typical)	12 hours
Impedance Measurement	Real-time contact quality using patented system

Table B.1: Emotiv headset characteristics

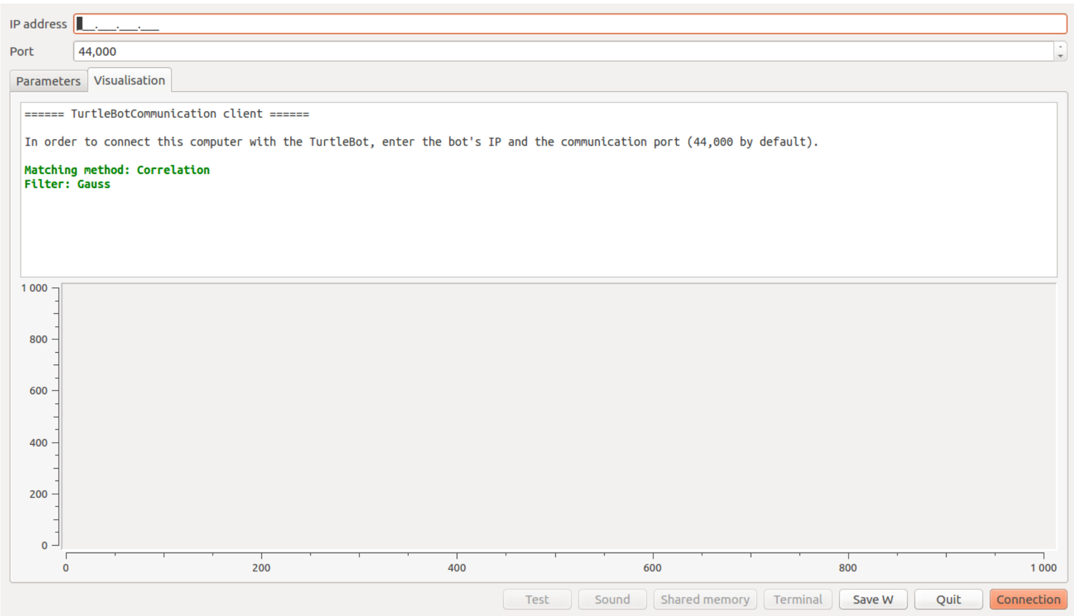
B.2 Communication

B.2.1 The client

On the client software, the user can change the algorithm configuration. He can choose the filter and their values, as well as the matching method. Once the algorithm configured, the *Visualisation tab* displays the sent command to the Turtlebot. The curves after the filter are also displayed.



(a) Parametres tab



(b) Visualisation tab

Figure B.1: TurtlebotCommunication client

Cardiff School of Engineering
Director of School Professor P J Bowen BSc PhD CEng MIMechE FInstP
Research Office
Deputy Director of School-Research Professor A Porch MA PhD CPhys MInstP MIET
Ysgol Beirianneg Caerdydd
Cyfarwyddwr yr Ysgol Yr Athro P J Bowen BSc PhD CEng MIMechE FInstP
Swyddfa Ymchwil
Dirprwy Gyfarwyddwr yr Ysgol-Ymchwil Yr Athro A Porch MA PhD CPhys MInstP MIET

CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Cardiff University
Queen's Buildings
The Parade
Cardiff CF24 3AA
Wales UK

Tel *Ffôn* +44(0)29 2087 0076
Fax *Ffacs* +44(0)29 2087 4939
Email *E-bost*
ENGINResearch@cardiff.ac.uk
www.cardiff.ac.uk/engin/

Prifysgol Caerdydd
Adeiladau'r Frenhines
The Parade
Caerdydd CF24 3AA
Cymru DU

07 August 2015

TO WHOM IT MAY CONCERN

Re: Internship for Mr Siméon Capy

This is to certify that Mr Siméon Capy has been a visiting student here at Cardiff School of Engineering, Cardiff University under the supervision of Professor Rossi Setchi. Mr Capy arrived in Cardiff University on 9th March 2015 and will be departing today, 7th August 2015.

Should you have any further queries please do not hesitate to contact me.

Yours faithfully,

Mrs Chris Lee
Deputy Research Office Manager
Cardiff School of Engineering

Tell: +44 (0)29 20874825 Fax: +44 (0)29 20874939 E-Mail: LeeC3@Cardiff.ac.uk