UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Master Degree in Biomedical Engineering

EEG-BCI Signal Processing and Analysis to Avoid Trajectory Planning Mistakes of a Smart Wheelchair

Master Thesis by:

FRANCESCO VELLA

Supervisor:

Prof. **ANDREA MONTERIÙ**

First Co-Supervisor:

Prof. **SABRINA IARLORI**

Second Co-Supervisor:

Prof. **FRANCESCO FERRACUTI**

Academic Year 2020-2021

Acknowledgments

First, I would like to thank my supervisor Prof. Monteriù for giving me the opportunity to work on this challenging but exciting topic, and my co-supervisors, Prof. Iarlori and Prof. Ferracuti, for their kindness, patience, availability, and the guidance they offered me throughout this journey, that culminates with this manuscript.

Second, I would also like to thank PhD student Karameldeen Omer, for his assistance, kindness, support and availability for confrontation, whenever there was the need.

Third, I would like to thank, not less important, my friends Veronica, Mara, Sarah, Sara and Matteo, marvellous people with whom I had the pleasure and honour to work and share precious moments of my life with (and to whom I own much more that they will ever know).

A thanks also goes to the other colleagues which I had the pleasure and the honour to meet (but sometimes not to be able to work with), but nonetheless they too have enriched my experience and my journey here at UNIVPM, during these last 2 years of master's degree course.

Finally, I would like to thank my mother and my father, and distant relatives too, who are still supporting me to this day, and never lost their faith in me, even when I did in "darkest times", always pushing me forward.

Index

Introduction

BCI (Brain Computer Interface) is a device that allows the recording of brain activity, either in an invasive (intra-cortical potentials recording with arrays of implanted micro-electrodes) or more frequently non-invasive way (scalp recording via an electrode cap of the Electroencephalogram, or EEG), and translation of such signals in an appropriate interpretable command for an external device (namely, a robot or a computer) to carry out a specific action.

BCIs find wide application in the fields of biomedical engineering, neuroengineering, and in particular rehabilitation, or as a supportive tool for patients having a disability either insurged after a traumatic event (e.g., stroke, paralysis) or as an ultimate result of neurodegenerative disease (e.g., amyotrophic lateral sclerosis).

Regarding the last category of patients, an interesting BCI application is found when coupled with assistive robots such as "smart" wheelchairs: wheelchairs embedded with on-board sensors (encoders, laser rangefinders etc.) that enable semi-autonomous indoor navigation functionalities (self-localization, mapping, trajectory planning). Such coupling of devices can be defined as "Brain-Controlled Wheelchairs" (BCW). Over the years, several BCW approaches were presented in different research area, which efficiently solved the navigation problem, but they have one issue in common: the lack of a fail-safe system in case of trajectory planning mistakes, or wrong sensor readings, thus failing to detect either small obstacles which can enter in collision course with the moving wheelchair, or unexpected dangerous situations for the user.

A solution to this problem has been proposed in the last few years: to include human input in the wheelchair control feedback loop by exploiting a peculiar EEG potential generated spontaneously from the subject under certain conditions, the Error Related Potential (ErrP), and its detection to be used as a feedback signal to prompt changes in trajectory. Although attempts to integrate this solution at a software level have been successful, this has only happened in an unrelated context, since the software integrations were tested with data obtained following an experimental paradigm that does not mirror the conditions experienced with a real BCW setup.

The aim of this thesis is to propose an EEG acquisition and processing protocol, defined in a setting as close as possible to the real-time experience, that ultimately would yield Event-Related Potential (ERP) signals suitable as feedback for obstacle/danger avoidance. The thesis consists of six chapters, where each chapter is devoted to a particular aspect of the analysis and research activities developed for reaching the aim; in detail:

- **Chapter 1**: Discussion on the EEG physiology, the concept of Event-Related Potential (ERP), its generation, and a few examples of the most common ERPs used in BCI applications.
- **Chapter 2**: a brief overview of the BCI system and its applications with particular focus on the BCW concept and some significant implementations.
- **Chapter 3**: Description of ErrP approach implemented by Ferracuti et al. **[1]**; the adopted smart wheelchair setup and its control system, brief description of its modelling and simulation in the Robotic Operating System (ROS) environment, and integration of an ErrP-BCI processing module in the control loop, promoting human-machine shared control.
- **Chapter 4**: Description of the proposed experimental paradigm, acquisition protocol and processing pipeline, from data acquisition to epoch classification.
- **Chapter 5**: Discussion of classification performance.
- **Conclusions**: Summary of the proposed work and possible future directions.

Chapter 1

Mentions about Electroencephalography, background physiology, and Event-Related Potential Generation

1.1 Physiological concepts

Most common BCI approaches rely on the acquisition of brain activity non-invasively recorded by placing electrodes on the scalp, in the form of electric potentials; these recordings constitute the electroencephalographic (EEG) signal. The functional units responsible for such activity are a specific type of cells called **neurons**, mostly present in the Central Nervous System (CNS), specifically those found in the outer layers (or cortex) of the brain**.**

1.1.1 Neurons and their electrical signals

Like all biological cells in complex organisms, apart from a nucleus and an extracellular membrane as protection from the external environment, what characterizes the neurons the most is their metabolic activity and their interconnectivity. Such activity is described by inward and outward ion fluxes across the cellular membrane via dedicated ion channels, generating electrical currents and stimuli that are transmitted among the cells via special connections called synapses. Most of the neurons have three sections: the soma, which contains the nucleus of the cell, and it is the processing unit, integrating the incoming electrical stimuli and generating its own as a response; the axons and the dendrites, which represent the outward and inward pathways for the electrical currents. (Figure 1).

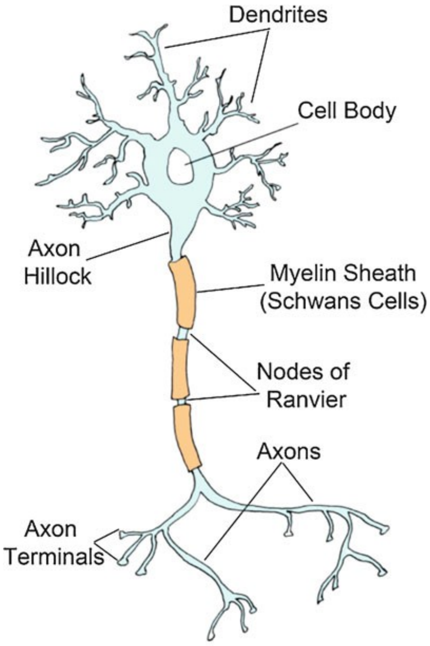


Figure 1 -*Typical Neuron* **[2]**

Three main types of electrical activity can be recorded from a single neuron, just by inserting a microelectrode into its membrane: resting membrane potential, (post-) synaptic potential and action potential.

The resting membrane potential is a constant voltage that can be measured across the membrane when the cell is at rest (no excitation applied) and whose values typically range from -40 to -90 mv, depending on the type of neuron being examined **[3].**

Post synaptic potential (Figure 2) is associated with transmission of information between neurons forming complex neural circuits found in the nervous system at synaptic contacts, and such potential is generated following activation at this level. It is a graded change in the membrane potential of the receiving neuron.

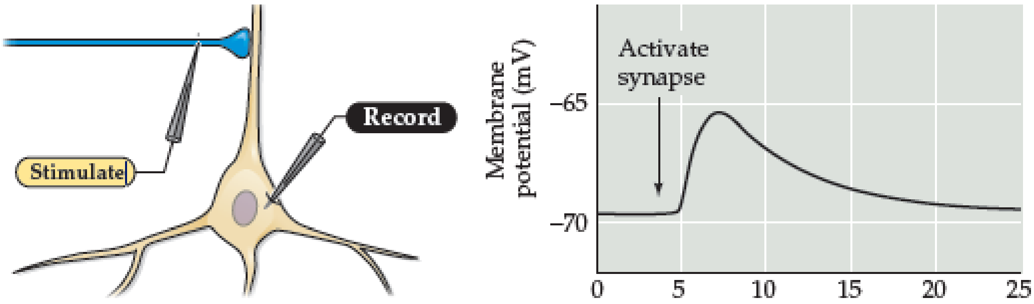


Figure 2 – *Post-synaptic potential **[3]**, modified from Ch.2 p. 34.*

The last type of signal is represented by action potentials. They travel along the axons, starting from the axon hillock (see Figure 1) up to the synapses, causing their activation. To generate such potential, enough current must flow across the membrane of the neuron. In normal circumstances, this current would be caused by different synaptic potentials that have propagated from the dendrites and have added up in the cell body.

Whenever summation of such potentials causes the depolarization (membrane potential becoming more and more positive) of the cell membrane up to the overcoming of a certain threshold potential, an action potential is spontaneously generated, characterized by a large overshoot (positive spike), followed by a repolarization phase (potential reverts to being negative).

Action potentials are also known as all-or-none events, meaning that when they occur, they always reach the same maximal amplitude voltage (circa +10 mV) and have the same duration (1 ms), tens or hundreds of times smaller than PSPs, longer in duration than action potentials, lasting up to tens or even hundreds of milliseconds (Figure 3).

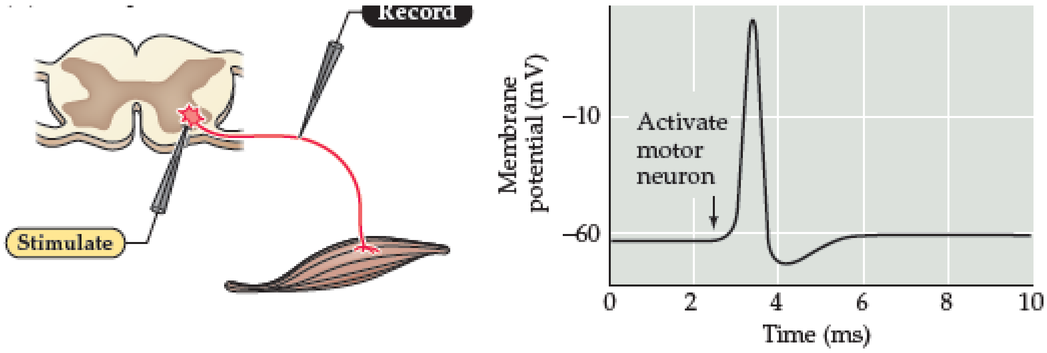


Figure 3 – *Action potential* ***[3]****, modified from Ch.2 p. 34.*

1.1.2 EEG physiology

After having illustrated the fundamental properties of the neurons, it's time to describe their contribution to the generation of EEG signals.

In fact, it's widely recognized that the EEG is generated by the summation of synchronized synaptic activities of cortical neurons, mainly post-synaptic potential, because, as it has been already mentioned, it lasts longer (up to hundreds of milliseconds) and has a greater potential field with respect to the action potential, increasing the probability of detection of such activity.

The population of neurons that are the most diffused in the cerebral cortex and also contribute the most to the composition of EEG waves are the so-called pyramidal cells, a type of neuron that takes its name from the triangular shape of its cell body, also characterized by a very long axon. The specific reason why this kind of cells play such an important part in EEG signal generation is that they meet the following conditions: 1) these cortical neurons are close to the scalp in distance; 2) the quantity of these cortical neurons should be large enough; 3) these cortical neurons must be synchronously activated with a certain geometric configuration.

The pyramidal neurons, which are in proximity to the scalp, are highly polarized with the major orientation perpendicular to the cortical surface **[4,5].**

The geometrical configuration and morphology of these cells and the principle of potential summation make sure that their clusters can be seen as single electrical dipoles, whose activity is in fact detected by EEG electrodes (Figure 4).

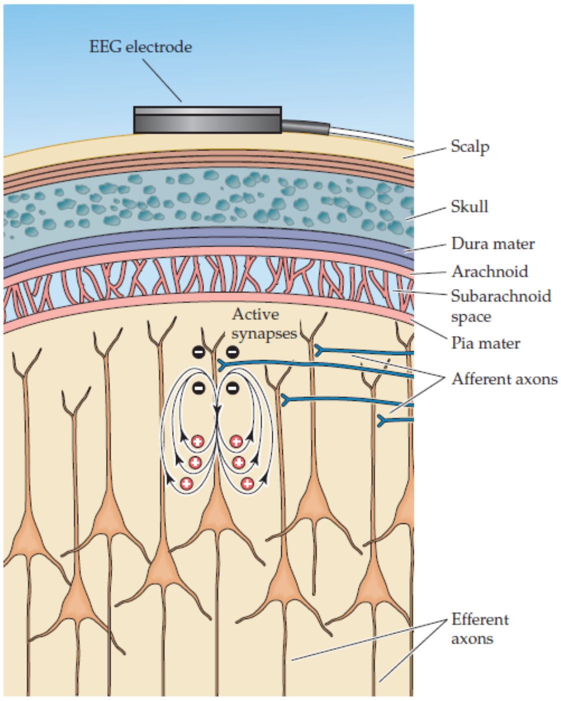


Figure 4 – *Pyramidal Neurons seen as dipoles* **[6]**

Each recording electrode placed on the scalp captures activity occurring within large parts of the underlying brain, perhaps 10–40 cm^2 of the cortical sheet, or cortical tissue volumes in the 10^3 to 10^4 mm^3 range. Thus, EEG represents the space-averaged source activity in tissue containing a number of neurons on the order of magnitude ranging from hundreds of millions to a billion **[7].**

1.2 Common BCI inputs: Event-Related Potentials

Having illustrated the fundamental concepts behind the EEG signal, in this section we will describe electrophysiological activities whose variations, when recorded by a surface electroencephalogram (EEG), are analyzed to extract a marker that can be used as an input for the BCI system. Such activities are called Event-Related-Potentials (ERPs), so called because they are specific patterns that are elicited in response to specific events or stimuli and, most importantly, are time-locked with respect to the latter. An ERP can be characterized, either partially or totally, by two types of components, exogenous or endogenous. Endogenous components are related to cognitive processes, and inner elaboration of information, while exogenous components are elicited as an automatic sensorial response to event detection **[8]**.

When a stimulus associated with a specific sensory, cognitive, or motor event occurs, it disturbs the spontaneous EEG activity. Although such neural responses are embedded within the EEG, given their low signal to noise ratio (SNR), as their amplitude is measured in terms of few microvolts against the tens of microvolts of background EEG **[9,10]**, they cannot be detected "as is" from the raw EEG signals, but signal processing techniques are to be applied in order to enhance SNR and extract such valuable signals.

One of the most widely used approaches to enhance SNR is the across-trial averaging in the time domain **[11,12]**. Other techniques include time-frequency analysis and single-trial analysis, but they will not be discussed in this work. Averaged ERPs are often composed of monophasic deflections which are characterized by their polarity, latency, amplitude, and scalp distribution **[5]**.

What follows is a brief description of the ERPs and other inputs that are most exploited in BCI applications.

1.2.1 P300

P300 is an endogenous potential that appears as a cognitive response whenever the user reacts to an unexpected (but known beforehand) sensory stimulus, capturing his/her attention. Its latency (time of appearance after stimulus presentation) is between 250 and 500 ms or greater, according to the complexity of the cognitive task performed by the individual, and this signal can be detected mainly in the central region of the scalp **[13]**.

In 1988, an experimental paradigm was devised, the "oddball paradigm", to elicit the P300 ERP. In this paradigm, a sequence of events, that can be classified into two categories, is presented to the subject at a very fast rate. In general, events of one category, that can be considered as "target stimuli, must have a lower frequency of occurrence with respect to the other. By engaging the subject in a cognitive task, for example by instructing him/her to progressively and mentally count the number of times the target stimulus appears, the P300 is generated **[5]**. (Figure 5).

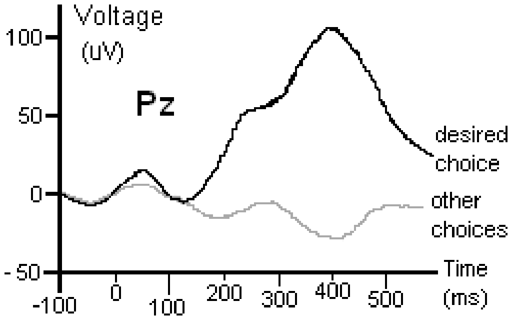


Figure 5 - *A P300 wave [14]*

1.2.2 Error-related Potential

The response that can be detected in the EEG, after an error is made or perceived by the subject, is called Error Related Potential. This term was coined to define the plurality of different signals, that have been reported in BCI literature throughout the years **[15]**. What follows is a brief description of the most significative forms of ErrP potentials that can be found in the literature. The first specimen of ErrP is documented by Falkestein et al, during choice reaction tasks. The authors describe a wave defined by two main components. The first component is the error-related negativity (ERN or Ne) **[16,17]** which is a negative potential peaking 50–200 ms after an erroneous response. Depending on the task, a second component an error-related positive potential, called error positivity

(Pe), may follow the ERN., with a latency of 200–500ms after the error (Figure 6). According to the literature, two possible meanings have been attributed to positive and negative components, respectively: the first being linked to conscious error perception, the second to error processing. **[18,19]**

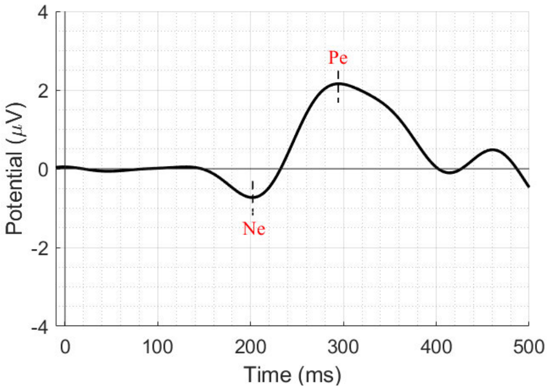


Figure 6 - *Example of Error-Related Potential* **[18]**

Experiments conducted by Ferrez et al. **[20]** produced a different and more complex type of ErrP, named "Interaction ErrP", measured in human-machine interaction context and/or, after a wrong BCI feedback (that is, an outcome not reflecting the user original intent) is presented to the user. This signal is generally characterized by four main components: a small positive peak around 200 ms, a negative peak around 250 ms, possibly related to the feedback-related negativity **[21]** a positive peak around 320 ms (possibly related to Pe) and a second negative peak around 450 ms after the BCI feedback (Figure 7) **[22]**.

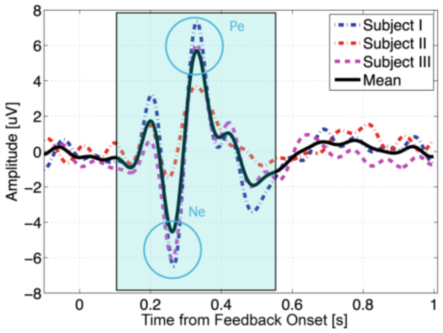


Figure 7 – *Typical Interaction ErrP Potential; Pe and Ne highlighted in figure* **[1]**.

In a more recent study, Milekovic et al. **[23]**, rename interaction ErrP as "execution error" (if the interface delivers erroneous feedback) and defines an "outcome error" (if a goal of an action is not achieved, i.e., the user is making an error). These two ErrPs have been detected also by Spuler et al. (Figure 8).

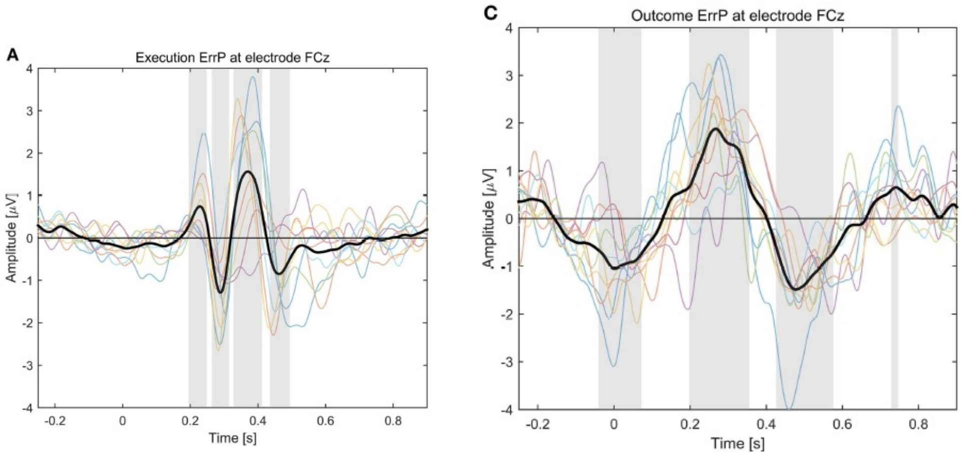


Figure 8 – *Left: Execution ErrP; Right: Outocome ErrP* **[19]**

Three common components are defined for both execution error (ExErrP) and outcome error (OutErrP) **[19]:**

First Negative Deflection (FRN/ERN), at 287 ms for ExErrP, at 2 ms for OutErrP;

Positive Peak (Pe), at 367ms for ExErrP, at 264ms (OutErrP);

Second Negative Deflection (N400), at same instant for both ExErrP and OutErrP.

As it can be noted from the examples reported here, the Error-Related Potential can assume different shapes. However, such differences among results, found in related literature, can be explained since, according to Iturrate et al. **[24]**, latencies and morphologies of ErrP components slightly change and differ depending on the experimental paradigm.

Chapter 2

BCI general concepts and applications

2.1 BCI Overview

BCI (Brain Computer interface) is a device that allows capturing of brain activity, either in an invasive (intra-cortical potentials recording with arrays of implanted micro-electrodes, not the focus of this paper) or more frequently non-invasive way (scalp recording via an electrode cap of the Electroencephalogram, or EEG) and translation of such signals in an appropriate interpretable command for an external device (namely, a robot or a computer) to carry out a specific action. In the most common non-invasive approach, BCIs consist of a cap with a variable number of mounted electrodes, that is connected to a processing unit (or computer) that parses incoming EEG data from the electrodes, and then transduces them into a suitable control input for other connected peripherals. Specifically, the BCI listens for specific signals called ERP (Event Related Potentials): generated by synchronous activation of groups of cortical neurons following presentation of electrophysiological stimuli either of visual (e.g., flashing lights) auditory, or somatosensory type.

BCIs find applications and have been tested in a variety of fields and applications (Figure 9):

- restoring lost neural output (for example, after spinal cord injuries, BCI stimulation of limb muscles, which would have otherwise gone into atrophy) or maintaining functionality;
- enhancement of natural CNS output: a BCI might assist the user while driving, by detecting drowsiness and/or attention lapses and then provides an output (e.g., a sound) that alerts the person; as an assistive tool for emergency automatic braking system **[1,25]**. or reminding directions **[26]** in order to improve overall car safety and to prevent traffic accidents;
- supplement natural CNS output, to facilitate specific task competition and increasing productivity: a person might use a BCI to control a third (i.e., robotic)

arm and hand or, if she/he is controlling the position of a computer cursor with a hand-operated joystick, might use it to select items that the cursor reaches;

- Finally, a BCI output might conceivably improve natural CNS output, by inducing activity-dependent CNS plasticity, in a rehabilitative context. For example, such BCI application reinforces normal motion of the body by decoding brain signals from a damaged brain area, relaying what physiological output would be to an orthotic device or to muscles, assuring their physiological recruitment and allowing the person to regain more normal limb control.

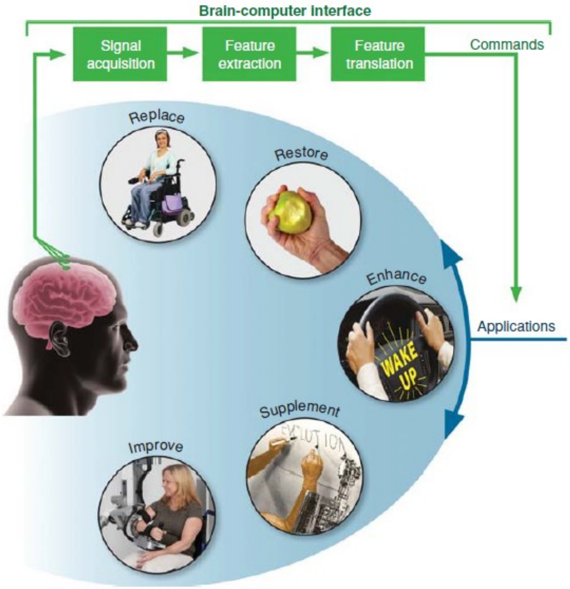


Figure 9 -*The basic design and operation of a brain-computer interface (BCI) system* **[7]**

Undoubtedly, however, the most exemplificative use that can be found for BCI is that of replacement of natural output that is progressively degenerating (as in case of amyotrophic lateral sclerosis **[27]**), or has been completely lost as a result of injury or disease: conventional motor pathways long time compromised, affecting phonation, word articulation, speech production, or resulting in full quadriplegia, tough retaining some form of movement such as blinking and eye movement (e.g., locked in syndrome) **[25,28].**

2.2 Significant biomedical applications for the BCI: spellers and Brain Computer Wheelchairs

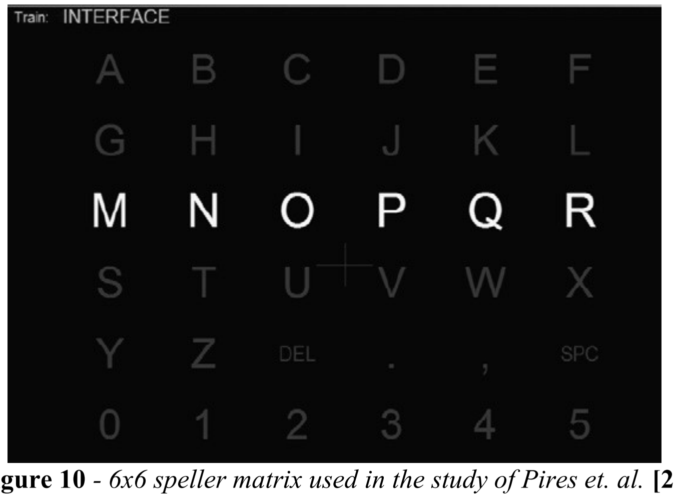
2.2.1 BCI spellers

Over the years in fact, several solutions involving BCIs have been presented/developed, as a way of restoring communication skills or some degrees of autonomy of movement.

For speech-deprived people, systems like BCI spellers have been devised, which allow the user to type in messages displayed on a computer screen or read via text-to speech, without effectively pressing any key on the keyboard but selecting one letter at a time just by looking at it from a grid displayed on screen.

For completeness, a brief explanation of BCI speller operating principle is to follow. The user wears an EEG cap connected to the computer and sits in front of the computer screen on which a square grid of letters is displayed (original paradigm provides a 6-by-6 matrix, but dimensions may vary), its rows and columns flashing intermittingly according to the "oddball paradigm".

The user chooses the letter he/she desires to communicate. By concentrating on and counting the number of times the desired letter flashes, the P300 ERP waveform appears on the EEG. Therefore, the BCI, after proper training, can automatically detect this waveform and provide the appropriate feedback and communicate with the rest of the system, which ultimately result in selection and presentation of the desired letter back to the user (Figure 10) **[8]**.



Train: INTERFACE

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

Figure 10 - *6x6 speller matrix used in the study of Pires et. al.* **[27]**

2.2.2 Brain-Controlled Wheelchairs

BCI are mostly employed as an assistive tool for severely impaired people who suffered brainstem or spinal lesions, which result in loss of motor skills, thus limiting their self-sufficiency and ability to perform daily activities.

For what concerns quadriplegic people, in particular those employing wheelchairs as a means of transport, interesting solutions, that researchers have been investigated for mostly 30 years, revolve around the idea of the so-called Brain Controlled Wheelchair (BCW), which is an assistive robot consisting of a smart, sensor-embedded wheelchair, capable of semi-autonomous navigation, that can interact with the user via BCI. This device allows severely impaired (i.e., quadriplegic, locked in) people to regain autonomy concerning movement and relocation, particularly in indoor environment. General operating principle is the same seen before: BCI system detects ERPs in the EEG and translate such information into desired movements of the wheelchair. Having defined this common background, different solutions were adopted throughout the years, in terms of type of ERP driving the system, level of compliance required from the user, and approaches to solve the navigation problem of the smart wheelchair, that involves self-localization and mapping of surrounding environment and trajectory planning to reach the desired indoor location. Here some examples of BCW settings are reported.

- *Tanaka et. al.*

For their approach to navigation problem, Tanaka et al. in **[29]** proceeded in the discretization of the environment, which is subdivided in squares of 1m (Figure 11) and the user has to choose where to move next. They use an EEG BCI based on motor imagery: by imagining left or right limb movements, thus activating the corresponding motor cortex, the user selects next motion of the wheelchair.

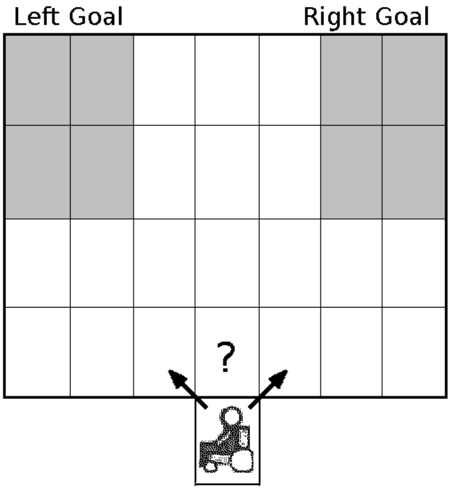


Figure 11 - *Experimental paradigm by Tanaka et al.* **[29]**

- *Minguez et. al.*

A similar principle was used in the wheelchair system developed by Minguez et al. [**30**], where a virtual reconstruction of the surrounding environment (as inferred from laser range scanner data) is displayed with a set of points in the free space that can be selected using a P300 EEG BCI (Figure 12), and these short-term goals are reached automatically. At the bottom of the screen a bar with additional commands is added, for either left/right 90-degrees turning of the wheelchair (arrows at the extremes), stopping or validation of given commands (traffic-lights-like figures) and abort selection (button in the middle). Since this system requires 11 minutes and 9 decision steps to travel 40 meters, as with Tanaka, it could be too tiring for the subject to use it.

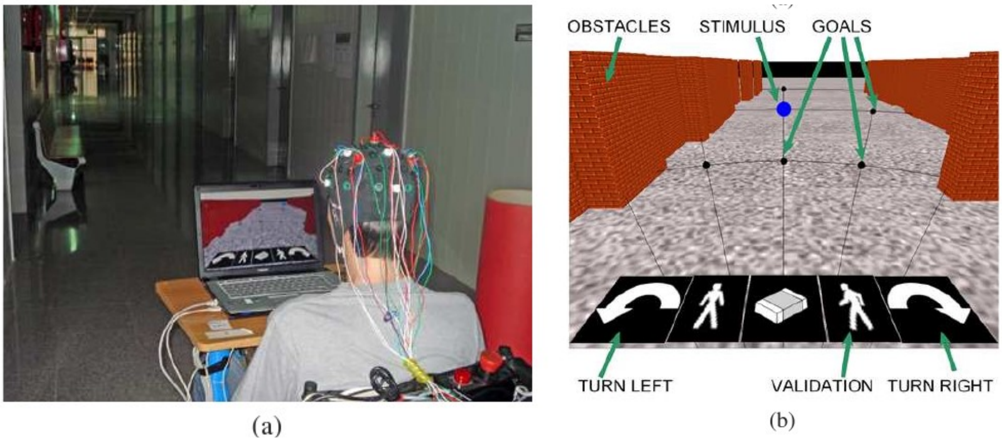


Figure 12 - *Minguez's brain controlled wheelchair* **[30]**. *(a) A snapshot of a participant navigating along a corridor. (b), environment abstraction displayed from the users point of view, with additional commands on the lower part.*

- *Rebsamen*

Rebsamen et. al. **[28]** proposed an alternative strategy for indoor navigation which exploited P300 BCI and oddball paradigm to select desired location amongst nine predetermined alternatives disposed on a 3x3 grid (Figure 13). The wheelchair will then move to such location following software pre-defined paths and can be stopped, during motion, either via another P300 algorithm employing only one item or by using a motor imagery approach.

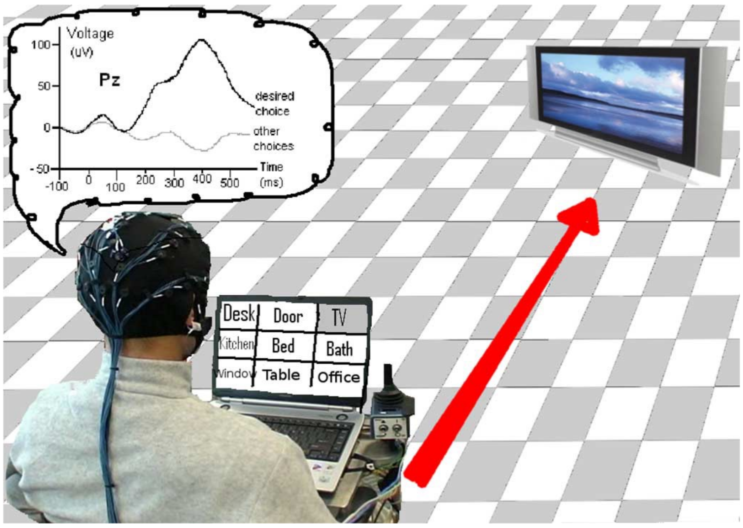


Figure 13 - *Overview of the BCW by Rebsamen et al.* **[28]:** *the user selects the destination*

P300 and motor imagery-based BCIs are very efficient for completing tasks concerning smart wheelchair navigation: target destination selection and start/stopping of the wheelchair. However, motor imagery approaches have two drawbacks:

-they require a lot of concentration from the user;

-they require extensive training in the BCI system before being put into use.

For what concerns the last presented P300 approach, the authors reported two limitations: the impossibility of using the BCW in an environment with no predefined paths, and the need to update such paths following modifications of the environment.

Unfortunately, a major problem, that might cause potential safety hazards to the user, persists in all the presented solutions. In fact, they do not have an efficient failsafe system for trajectory planning mistakes or wrong sensor readings thus failing to detect small obstacles which enter a collision course with the moving wheelchair, nor for unexpected dangerous circumstances, such as wheelchair approaching potholes or downward stairs.

For this reason, in recent years, the research activity has focused on the development of real-time feedback solutions from the human operator to the robotic system: the inclusion of the human observation in the control feedback loop of the wheelchair, exploiting the Error Related Potential **[1]** (Figure 14). A founding example of this approach will be illustrated in the next chapter.

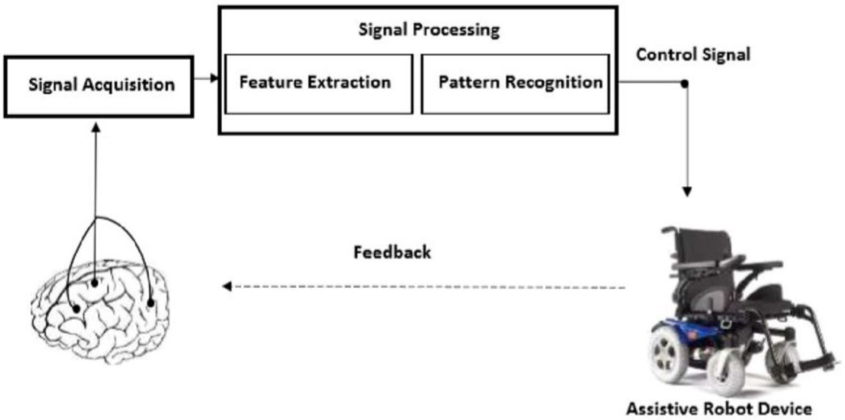


Figure 14 - *The feedback loop discussed in* **[1]**

Chapter 3

ErrP BCW Approach

3.1 Wheelchair Setup Description

In this chapter, a Brain Computer Wheelchair approach, based on ErrP detection will be illustrated, since it is the starting point from which this thesis work has stemmed.

About this kind of application, a solution has been designed and provided by Ferracuti and colleagues **[1]**.

In their work, they set up a smart wheelchair (Quickie Salsa R2), by adding proprioceptive and external sensors whose inputs, after being analyzed, are re-routed to a computer. The actuators of the wheelchair received appropriate velocity commands, once set a destination, following trajectories (both local and global) calculated according to a path planning algorithm. The wheelchair is equipped with the OMNI interface device, an internal control module. This controller could receive input from different devices of SIDs (Standard Input Devices) and convert them to specific output commands compatible with the R-net control system. In addition, the smart wheelchair equipment includes: a Microstrain 3DM-GX3-25 inertial measurement unit, an Arduino MEGA 2560 microcontroller, two Sicod F3-1200-824-BZ-K-CV-01 encoders, a Webcam Logitech C270 and a Hokuyo URG-04LX laser scanner (Figure 15).

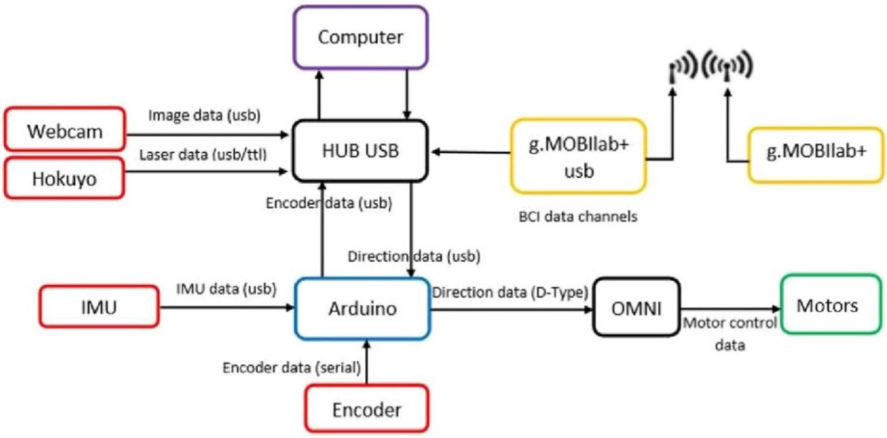


Figure 15 - *The scheme of the system setup with all sensors and devices* **[1]**

The encoders, the inertial measurement unit and the OMNI are connected to the microcontroller, while the microcontroller itself and the other sensors are connected via USB to a computer running ROS (Robot Operating System). Signals from the Sicod and Microstrain devices are converted by the Arduino and sent to the ROS localization module. The information provided by the Hokuyo laser scanner is used by the mapping module and by the path planning module for obstacle avoidance.

Sensors inputs and output commands are managed by a dedicated wheelchair software built in ROS environment, whose fundamental characteristics will be described in the following section, as it is propaedeutic for the rest of the dissertation.

3.1.1 ROS fundamental concepts

ROS is a free BSD-licensed software framework, that has become quite popular in the field control system design for robots thanks to its strong modularity, open-source nature and compatibility with other programming environments, interfaces and languages (e.g., Python, C++, Linux) **[31]**.

Essentially, ROS environment is based on packages. A ROS package is the most basic unit of the ROS software and contains the ROS runtime processes (nodes), libraries, configuration files, and so on, organized together as a single unit.

A node is the functional unit of ROS, representing specific runtime processes, receiving (*subscribing*) or sending (*publishing*) outcomes of their respective calculations (*messages*) through specific communication channels (*topics*).

Each package (and node) could model every aspect of the controlled system: sensor inputs (encoders, lasers etc.), the implementation of functions or algorithms, hardware and electronic components. Interconnections and proper communication among ROS packages and nodes constitute the whole wheelchair system (or wheelchair package), reported in Figure 16.

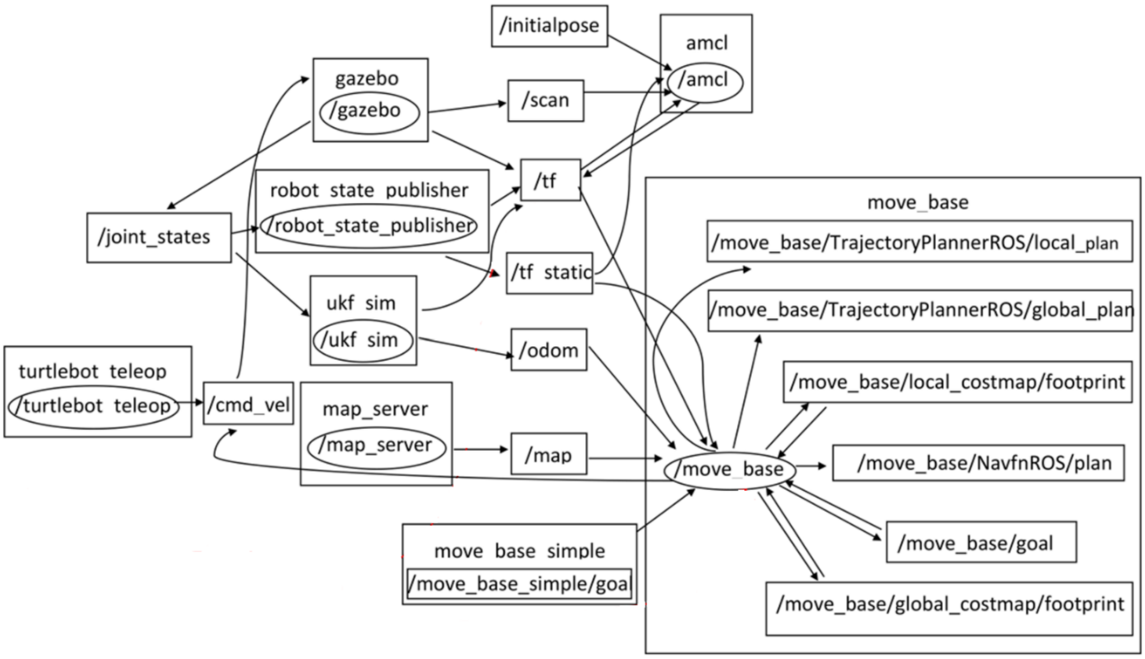


Figure 16 – *The wheelchair package (modified from* **[1]***)*

One important feature that ROS implementation added to the smart wheelchair system, is the capability of automatic navigation of wheelchair, having set a target destination. ROS has a specific set, or "upper level" package, called the navigation stack, that is specifically used to handle the navigation problem of the wheelchair. Such problem can be divided into three aspects: self-location of the wheelchair in the acquisition, destination setting, and trajectory planning. A pre-programmed version is available for download at **[32]**, along with related documentation.

A 2D navigation stack takes in information from odometry, sensor streams and a goal pose, then outputs safe velocity commands to a mobile base. A block scheme of the stack is displayed below (Figure 17).

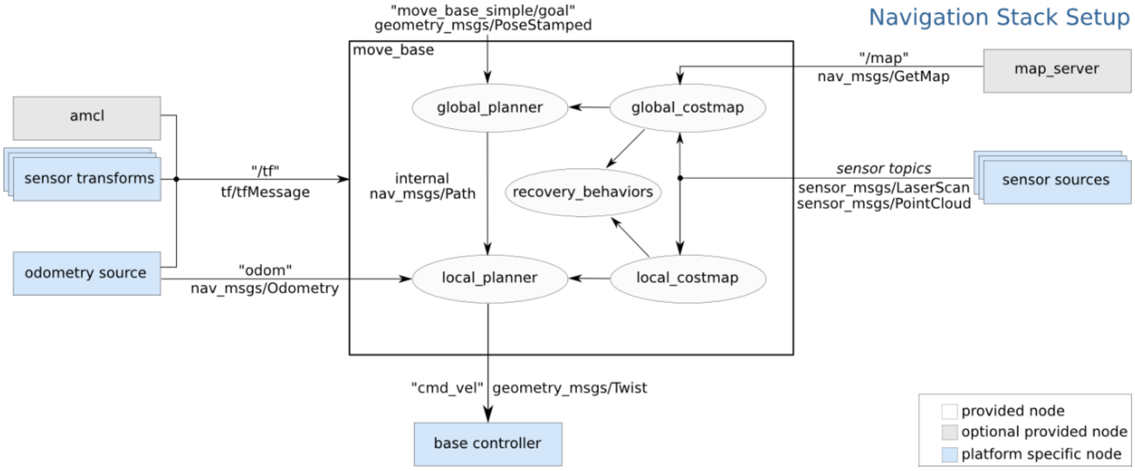


Figure 17 - *The navigation stack implemented architecture*

The navigation stack assumes that the robot (or in our case, the wheelchair) is configured in a particular manner in order to run. The diagram above shows an overview of this configuration. The white components are required components and the grey components are optional components, both are already implemented, while the blue components must be created for each robot platform. **[32]**.

3.2 ErrP-BCI processing module integration

Starting from the scheme in Figure 16, Ferracuti and al. **[1]**, wished to add human feedback to the control loop of the wheelchair, particularly in the context of obstacle avoidance. In particular, a new software package (the Point Cloud Node) has been developed which creates a link between the BCI and the navigation task.

The implemented package is able to:

- subscribe and listen continuously to the robot position;
- transform the robot pose from the robot frame to the map frame;
- subscribe/listen for the trigger generated by the BCI
- create the point cloud geometry of a fictitious obstacle and position it on the inner wheelchair map of the robot workspace, precedingly drawn via laser scanning.

The adjourned wheelchair package is presented in Figure 18.

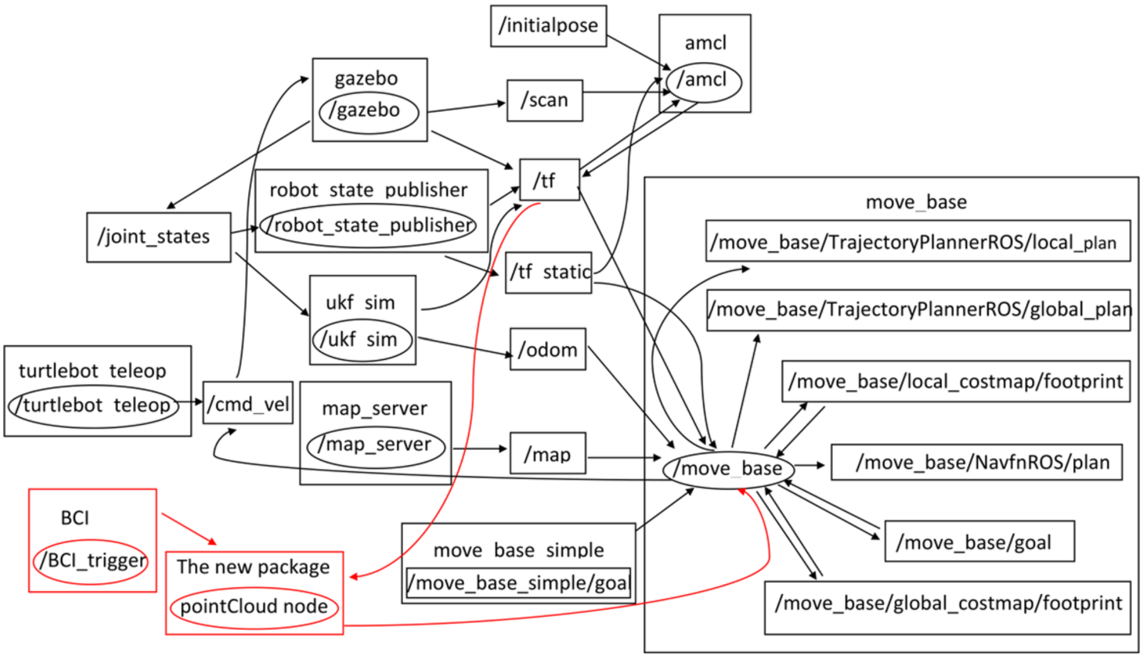


Figure 18 - *The wheelchair package, integrated with the BCI_trigger and pointCloud node* [**1**]
Note: nodes and topics used for testing the wheelchair simulated in Gazebo are the same as those developed for the real system.

A feedback policy is then added which defines the procedure the smart wheelchair system has to follow when an obstacle is detected:

- Speed reduction of the wheelchair, to allow the navigation system to apply due corrections on predefined path to account for the new detected obstacle, without endangering the human operator.
- Virtual obstacle creation on the inner environment map (built beforehand with previous laser scanning) stored in the navigation stack; cylinder of the same dimension of the wheelchair is placed along its trajectory (i.e., in the line of sight of the user) at a predefined distance.
- Path planning iteration: as soon as the virtual obstacle is introduced into the map, the path planning step is repeated, with modification of the local trajectory in order to avoid the obstacle and guarantee safety as long as the area of the real obstacle is inscribed in corresponding virtual one.

A flow chart of the feedback policy is displayed in Figure 19.

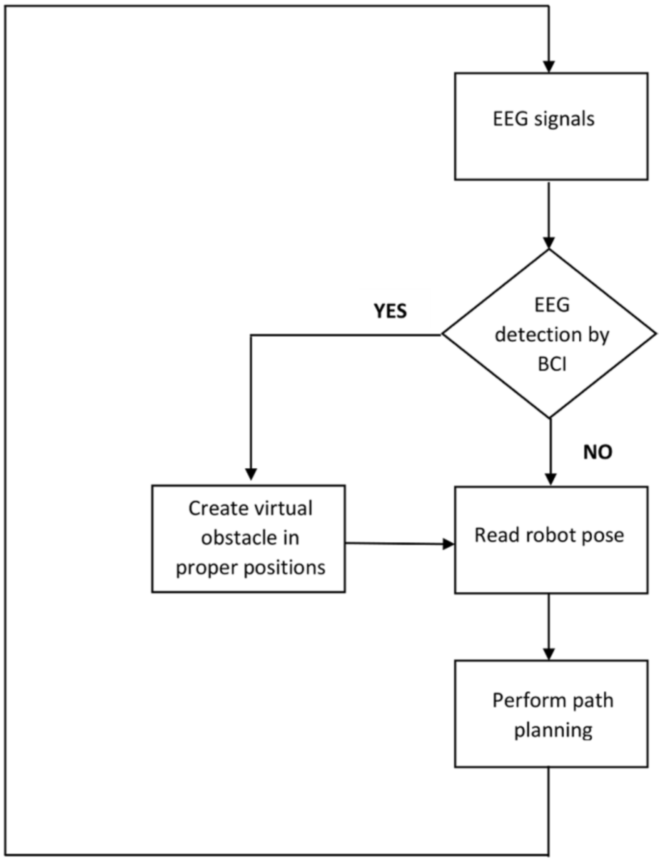


Figure 19 - *Flowchart of the feedback policy, presented in* **[1]**

3.3 Testing new architecture

In order to test this new architecture, Ferracuti and colleagues **[1]** firstly trained the BCI with an offline dataset available from Chavarriaga et al, which exploited ErrP (Error-Related Potential) detection.

The dataset described in Chavarriaga and Millan **[22]** was used by the authors to evaluate the preliminary tests of the proposed framework for assisted vehicles.

Six participants were enrolled and received the training before the acquisition in two recording sessions separated weeks apart. Six subjects performed two recording sessions (session 1 and session 2) separated by several weeks. Both sessions consisted of 10 blocks of 3 min each: each block was composed of approximately 50 trials. In each trial, the user, only assessed whether an autonomous agent performed the task properly. In particular, the task consisted of a cursor (green square) reaching a random target on a computer screen (red square). At the beginning of each trial, the user was asked to focus on the center of the screen, while during the trial was asked to follow the movement of the cursor, knowing the goal of the task. Thus, ErrPs were elicited by monitoring the behaviour of the agent. If the agent's cursor did not reach the position as the random target was considered an error, while trials on which the agent's cursor reached the target position were considered correct. After correct trials, the target position randomly changed positions. Each session is composed of about 500 trials and the agent error probability was set to either 0.20 or 0.40. All the trials were windowed (0–500 ms). In the paper, the Non-Target trials (NT) refer to a successful reaching of the final target, whereas, the Target trials (T) refer to those trials where the cursor does not reach the target position, namely the ErrP signal is evoked.

EEG data was fed to the BCI trigger node for pre-processing (temporal filtering, spatial filtering and epoching) and classification of said epochs into Target (with ErrP) and Non-Target (no ErrP). Cumulative performance scores are reported in Table 1.

	Accuracy	Sensitivity	Specificity	F1-score
Training	0.846	0.853	0.823	0.897
Testing	0.781	0.814	0.650	0.857

Table 1 - Overall performance indicators.

Classification outputs of Target and Non-Target epochs would generate a Boolean feedback message, sent to the pointCloud node, of value 0 (non-Target) or 1 (Target): if the latter occurs, virtual obstacle generation will be prompted (Figure 20).

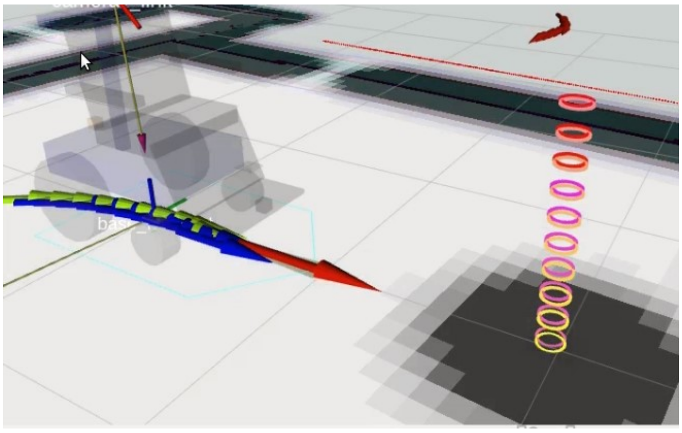


Figure 20 - *Wheelchair simulation, obstacle generation in Gazebo Credits: Karameldeen Omer*

The authors concluded describing future directions of their work **[1]:**

- new experiment performed with people watching videos realized using the wheelchair simulated in Gazebo: a first-person-view simulation of agent in motion avoiding virtual obstacles, in order to create an experience as close to the real one as possible and validate EEG classification performance at the same time;
- redefine feedback policy when BCI trigger is received and the path planning is iterated, since there is currently no way to precisely determine the effective distance between the obstacle and the wheelchair;
- performing trials with human operators wearing the BCI system integrated with a real wheelchair, in order to test and validate EEG classification performances and to find ways to reduce time of obstacle recognition and response;
- test of the whole system with the complete feedback loop, varying operators and obstacles.

Chapter 4

Proposed Processing Pipeline

Classification results obtained in the approach defined in the previous chapter are promising, but they are obtained using pre-acquired EEG data for offline BCI training generated with a protocol which is not representative of an experience with a real BCW modelled after the simulated one previously described.

The aim of this work is to present an EEG acquisition and processing protocol, defined in a setting as close as possible to the real-time experience, that ultimately would yield ERP signals suitable as feedback for obstacle/danger avoidance**.**

Given the low signal-to-noise ratio (SNR) due to artifact presence, the decoding of user intentions from brain patterns therefore requires the application of signal processing to attenuate noise from EEG data. In the context of classification, another important issue is the reduction of features dimensionality to attenuate overfitting of training data and to increase the computational efficiency of algorithms for real time operation **[27].**

In Figure 21, a block scheme of the proposed processing pipeline is portrayed. What follows is a detailed description of each block. This pipeline was conceived keeping in mind future real time application.

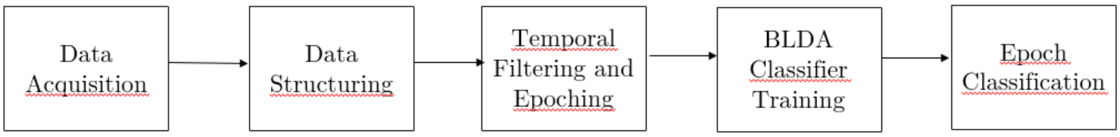


Figure 21 – *Processing Pipeline of EEG data analysis*

Apart from Data Acquisition, each step was carried out by a series of functions implemented in MATLAB source code, because it provides easy manipulation of data in the form of matrices, easy visualization of data, and it can interface with other programs (in this case, ROS environment).

Thus, the raw datasets are fed to MATLAB code for proper signal processing, training and testing of BCI classifier.

4.1 Data Acquisition

4.1.1 Equipment

Data is acquired using the following equipment (Figure 22):

- *g.GAMMAcap*, equipped with active electrodes (eight electrodes), specifically for EEG recordings;
- *g.MOBIlab+ADC*, that collects data from a g.MOBIlab+ device, a tool for recording multimodal biosignal data on the laptop or PC, namely an amplifier which transmits the data wirelessly via Bluetooth 2.0.

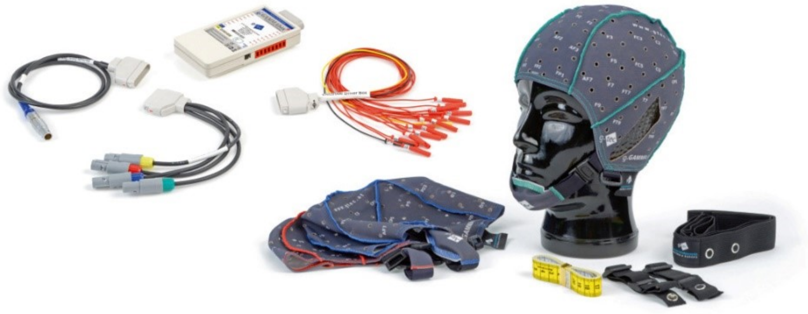


Figure 22 – *Fac-simile of used equipment* **[33]**

Electrodes on the cap are placed according to the 10-20 system and a total of eight channels were recorded: AF8, Fz, Cz, Pz, P3, P4, PO7, PO8 (Figure 23). Electrode P10 was used as the reference electrode, whereas the ground electrode was placed at left ear (electrode A1) **[34]**. All signals were digitized with 16-bit resolution and sampled at 256 Hz, amplified with sensitivity of 500 µV and pre-filtered with a 0.1-30 Hz bandpass filter plus 50 Hz notch filter, in order to cut out the majority of high frequency artifacts (i.e., muscle activity, line noise, and high frequency EEG waves), and low-frequency signal drift and DC offset. The software used for recording and acquisition management is BCI2000 which also allowed session definition, real-time EEG data visualization, pre-filtering and storage, trigger/stimulus events recording.

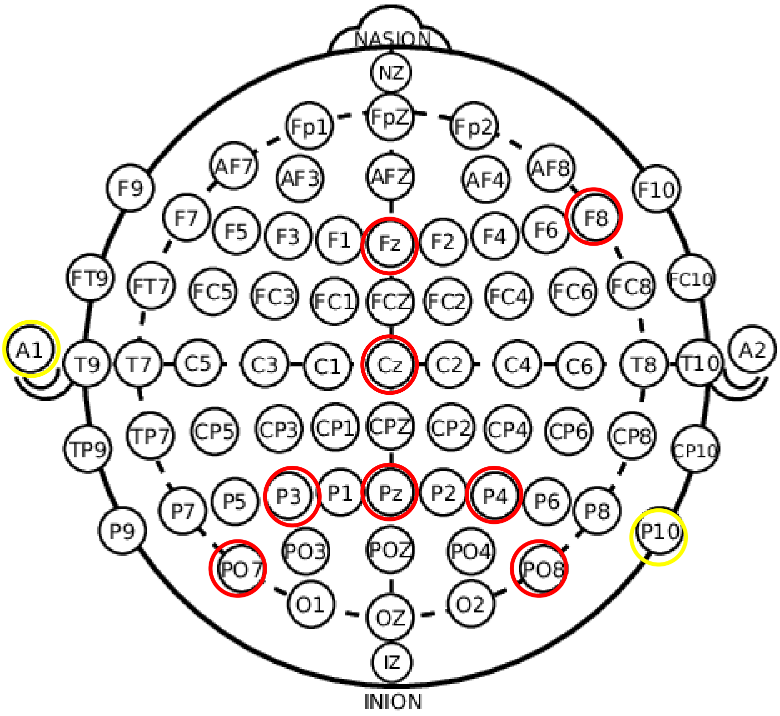
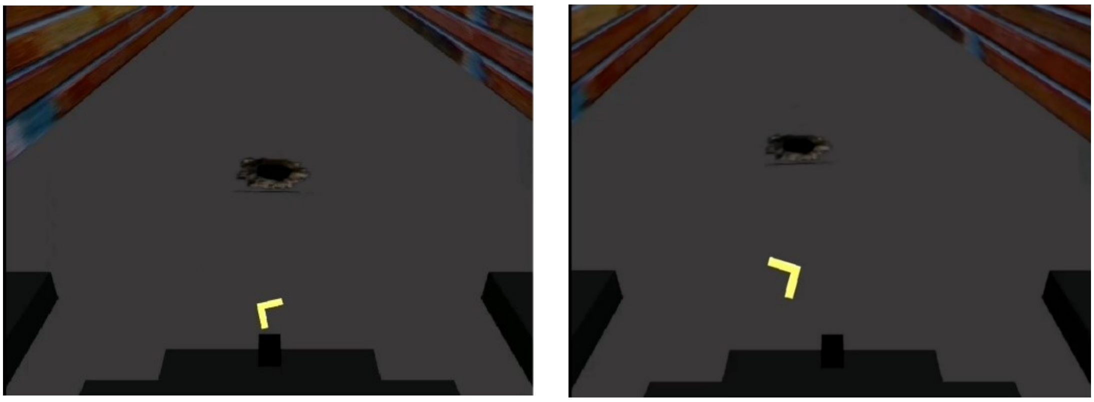


Figure 23 - *The extended International 10-20 system position of scalp electrodes. Electrodes are placed at 5%, 10% and 20% spacings relative to standard skull measurements (e.g., nasion-inion). Abbreviations: A = Auxiliary (Ear lobe, shown, or mastoid, in our experiments), C = central, P = parietal, F = frontal, Fp = frontal polar, O = occipital. Modified from* **[35]**

4.1.2 Experimental Paradigm

Ten Subjects have been involved in this experiment; one dataset recorded for each of them at different times. They were seated in front of a monitor on which was presented a first-person-view video simulation (seven minutes total) of the wheelchair system while it is moving, recreated using Gazebo. This video is made up of 50 repetitions (trials) of three variants of the same task: the wheelchair that is approaching the hole. The wheelchair at the beginning of trial is facing an obstacle (namely, a hole); a directional cue then appears, arrow pointing left or right, as prediction of next movement direction (Figure 24, a) and b)).



a) b)

Figure 24 – *Above a): left turn trial; below b): right turn trial. Credits: Karameldeen Omer*

At this point, simulated wheelchair may respect the directional cue, turning left (or right), thus avoiding the hole (correct trial, “Turn”), or it may go straightforward, ignoring directional cues and passing above the hole (non-correct trial, “Pass”).

The proportion of the trial types composing the video, mirrored the one used in a standard oddball paradigm **[7]**: 80% of the total are the “correct” trials (with the wheelchair avoiding the hole), the remaining 20% represent the “non correct” ones (the wrong behaviour of the wheelchair, passing the hole).

To augment the data available for classifier training, 3 (or 4) blocks of 50 trials, interleaved with rest pauses, were presented to the subject in one single sitting/session, for a total of 150 (or 200) trials. Each single trial medially lasted 5 seconds followed by a rest period of 3 seconds.

Subject was instructed to press the “Enter” key on a keyboard whenever he/she has an intuition about the future direction that would be taken by wheelchair, either to cross or to avoid the hole. The latency information about such user inputs is thus recorded, via BCI2000 software, in a channel apart from brain signal. The reason for this is to have a time sequence of trigger events, around which EEG signal could be epoched, to ensure subsequent time-locked EEG analysis and to better detect the presence of a time-locked ERP. Trials were presented shuffled, in order to remove any kind of bias due to prediction/anticipation of possible pattern in variants’ presentation i.e., the three possible wheelchair trajectories.

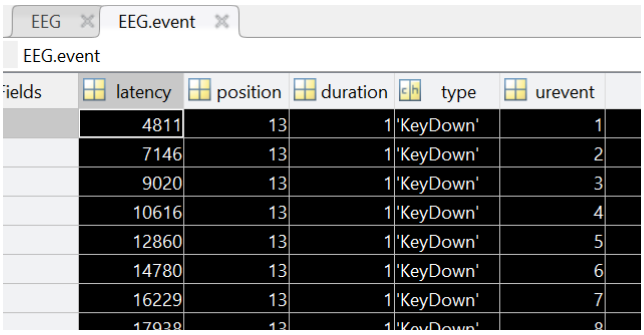
4.2 Data-structuring

The functions for actual pre-processing and classification require that, for each dataset, data are loaded into MATLAB environment using EEGLAB toolbox **[36]** and stored as struct variable ‘EEG’ in which event labels are renamed from “KeyDown” (indicating key press) to “Turn” or “Pass”.

To respect subsequent functions’ dependencies, data are re-organized in a structure (saved as a struct type variable, called Data), with the following fields:

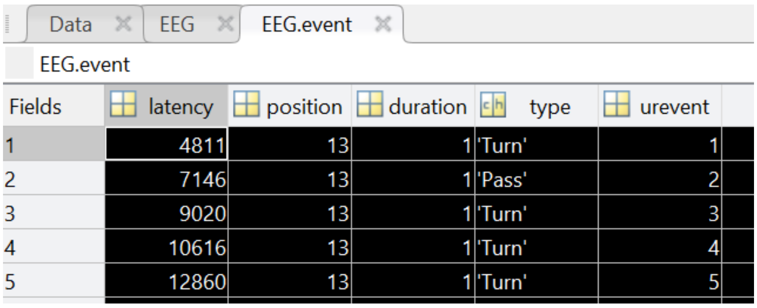
- `Data.srate`: contains the value of the sampling frequency (in this case ‘256’);
- `Data.pnts`: contains the value of number of samples of the EEG signal recorded from the channel;
- `Data.data`: contains the 8-by-N matrix of all EEG data, where 8 is the number of channels and N is the number of samples;
- `Data.events`: a sub-structure containing all information related events, characterized by the following fields:
  - ➢ `events.latency`: column listing all time instants (although in sample value) at which the key is pressed during the trials.
  - ➢ `events.type`: column listing the previous type of events (either ‘Turn’ or ‘Pass’) that are re-labelled, re-coded for later purposes in a binary fashion (‘1’==’Turn’, ‘0’==‘Pass’)

Results of structuring procedure can be seen in Figure 25 a), b), c).



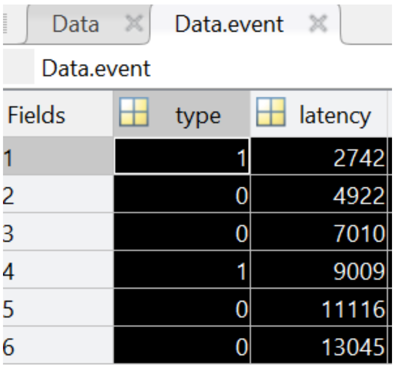
Fields	latency	position	duration	type	urevent
	4811	13	1	'KeyDown'	1
	7146	13	1	'KeyDown'	2
	9020	13	1	'KeyDown'	3
	10616	13	1	'KeyDown'	4
	12860	13	1	'KeyDown'	5
	14780	13	1	'KeyDown'	6
	16229	13	1	'KeyDown'	7
	17938	13	1	'KeyDown'	8

a)



Fields	latency	position	duration	type	urevent
1	4811	13	1	'Turn'	1
2	7146	13	1	'Pass'	2
3	9020	13	1	'Turn'	3
4	10616	13	1	'Turn'	4
5	12860	13	1	'Turn'	5

b)



Data.event

Fields	type	latency
1	1	2742
2	0	4922
3	0	7010
4	1	9009
5	0	11116
6	0	13045

c)

Figure 25 – *Structuring steps*

4.3 Temporal Filter and Epoching

EEG data undergoes temporal filtering. For denoising, at first a 4th order bidirectional Butterworth low-pass filter is applied, with cut-off frequency of 5 Hz.

Then, using the MATLAB function decimate, decimation is applied, by a factor of 4, as a form of down sampling and anti-aliasing, so that the sampling frequency of the signal is reduced to 64 Hz. This function in turn relies on the application of Infinite-Impulse-Response (IIR) Chebyshev filter of order 8. So, to reduce artifacts introduced by this filter, decimation was done in 2 steps, each by a factor of 2.

Afterwards, a final 4th order bidirectional high pass filter at 1 Hz is applied to the signal. The ultimate choice of 1-5 Hz cut-off frequencies is based on the fact that the main components of the sought ERP waveforms are in the delta wave frequency range **[19].**

If the signal is digitized at a rate that is higher than the required Nyquist rate to capture the relevant activity, it may be useful to decimate the sampled signal to the minimum effective sampling rate for more efficient processing and storage. However, just as with sampling in analog-to-digital conversion, it is important to avoid aliasing, by low pass filtering the signal before decimation, with a cut-off frequency equal to one-half of the decimated sampling frequency **[7].**

Epoching of signal is performed, with windows of [-100 600] ms, centered around the events (t=0 ms). Epochs are then stacked into a single tridimensional matrix (*Nx8xNumber of Epochs*).

Given the fact that only one dataset is available for each subject, five-fold stratified partition (repeated 20 times) in training and validation sets is applied, and this will be used for subsequent classifier learning and testing.

4.4 BLDA Classifier

The Bayesian Linear Discriminant Analyzer (BLDA) **[37,38]**, derived from the Fisher Linear Discriminant Analyzer (FLDA), discriminates feature vectors (stacked together in a matrix *X*) belonging to two different classes *y* {-1,1}, by mapping feature points along a direction (defined by weight vector *w*, which needs to be estimated) that allows maximization of inter class differences while minimizing intra class variability at the same time.

Differently from FLDA, weight regularization is done automatically in a Bayesian framework (defining probability distributions representing a priori information and likelihood about the weights), without the need of time-consuming and computationally demanding cross-validation. Such framework is structured in at least two levels of inference **[37]**: first level deals with *w* estimation, second level accounts for estimation of hyperparameters *β* and *α*, respectively describing likelihood and a-priori distribution about *w*.

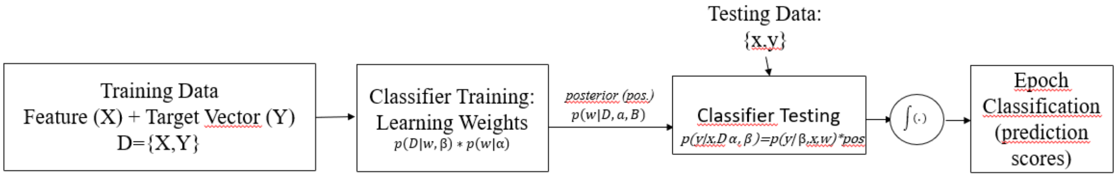


Figure 26 – *General scheme of BLDA*

The BLDA works starting from the assumption that each class label value y_j is the weighted sum of features in the corresponding feature vector x_j, plus additive independent noise sample n_j taken from a white Gaussian noise distribution (see Equation 1).

$$y_j = w^t x_j(i) + n_j, \;\; j = 1 \ldots N, \;\; w \in \Re^W . \qquad (1)$$

Here, *N* is the number of class labels and corresponding to feature vectors and *W* is the number of features. Dependency on *(i)* values can be ignored, since it pertains to a third level of Bayesian inference, not treated in this thesis. As can be seen from BLDA scheme, we denote *X* as the matrix obtained by horizontal stacking of all feature vectors, *Y* as the row vector obtained from concatenation of corresponding class labels; therefore we can define the training dataset D as D={*X,Y*}.

Next, the joint-likelihood (2) and a priori distribution (3) for the weights *w* are calculated, given the assumption of Gaussian noise, and they assume the following shape, respectively:

$$p(D|\beta, w) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp(-\frac{\beta}{2}\|w^t X - Y\|_2^2). \quad (2);$$

$$p(w|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{W}{2}} \exp(-\frac{\alpha}{2}\|w\|_2^2). \quad (3);$$

where β and α represent the respective inverse variances. Setting the prior distribution also as a Gaussian has two main advantages. First, this formulation is equivalent to put a constraint on the weights' norm (similar to ridge regression, or weight decay) **[38]**, thus preventing them from getting too large and cause overfitting. Second, the posterior distribution (*pos*. in Figure 26) is also a Gaussian, since derives from prior and likelihood distribution, and can be computed using Bayes rule:

$$p(w|\beta, \alpha, D) = \frac{p(D|\beta, w)p(w|\alpha)}{\int p(D|\beta, w)p(w|\alpha)\, dw}. \quad (4)$$

To test classifier with new feature vector *x* and predict its class *y*, it's only sufficient to calculate a new likelihood function, with the same form as in (1) and, by combining it with the posterior distribution of the weights, we obtain the predictive distribution for the new data:

$$p(y|\beta, \alpha, x, D) = \int p(y|\beta, x, w)p(w|\beta, \alpha, D)\, dw$$

$$p(y|\beta, x, w) = \left(\frac{\beta}{2\pi}\right)^{\frac{1}{2}} \exp(-\frac{\beta}{2}(w^t x - y)^2). \quad (5)$$

Then, to calculate actual class probability (for ex., if *y*=1), using previous equation, it can be written:

$$p(y \geq 0|\beta, \alpha, x, D) = \left(\frac{\sigma}{2\pi}\right)^{\frac{1}{2}} \int_0^{\infty} \exp(-\frac{\sigma}{2}(y-\hat{y})^2)\, dy \qquad (6);$$

with the parameters of Gaussian in (6). being:

$$\begin{aligned} \hat{y} &= A^{-1}XYx, \;\; \sigma = \frac{\beta}{1+\beta x^t A^{-1} x} \\ A &= (\beta XX^t + \alpha I). \end{aligned} \qquad (7).$$

Until now parameters α and β are taken for granted, but they are really inferred in the second Bayesian level of inference (not discussed here, for more information, see **[37,38,39]**).

Chapter 5

Classification Results and Discussion

This chapter is dedicated firstly to the definition of metrics, and indicators used to evaluate classifier performance, then a table containing such matrics' values is reported, followed by a brief discussion on the matter.

5.1 Definitions of performance indicators

In a binary classification problem, such as the BLDA classifier described in the previous chapter, each feature vector/epoch needed to be classified into two groups ("Turn" or "Pass"). For each of these groups, a probability distribution curve can be defined. Since they overlap the feature axes and an item to be classified can only belong to one curve, it's necessary to set a threshold value, with respect to which classification can be performed ("positive" class above, "negative" below). Thus, four possible classifications can be attributed to each feature vector, according to corresponding true class label:

- ➢ if predicted class label and true class label are both "positive", classification result is defined as *true positive* (TP);
- ➢ if predicted class label is "positive", but true class label is "negative", classification result is defined as *false positive* (FP);
- ➢ if both predicted and true class labels are "negative", we have a *true negative* (TN);
- ➢ if predicted class label is negative, but true class label is positive, we have a *false negative* (FN).

By considering the total number of elements for each of these four categories, four classifier performance indicators have been calculated:

- *Sensitivity*, defined as the capability of the classifier to correctly classify *true positives*, it ranges between 0 and 1, and can be obtained using the following formula:

$$Sensitivity = \frac{TP}{TP + FN}$$

- *Specificity*, defined as the capability of the classifier to correctly classify *true negatives*, it ranges between 0 and 1, and can be obtained using the following formula:

$$Specificity = \frac{TN}{TN + FP}$$

- *Accuracy*, defined as the capability of the classifier to predict classes correctly, either *positive* or *negative*, it ranges between 0 and 1, and can be obtained using the following formula:

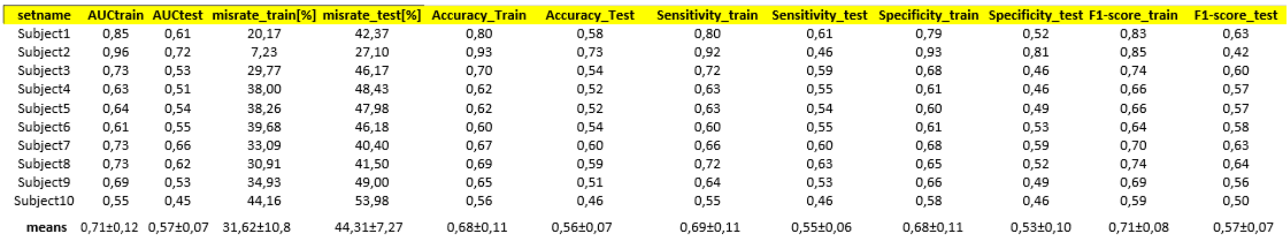
$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

- *F1-score*, which is another way to define the accuracy of a classifier:

$$F1\text{-}score = \frac{2 * TP}{2 * TP + FN + FP}$$

Another indicator used in this paper is the Area Under the Curve (AUC) of the Receiver-Operating Characteristic (ROC) curve. The ROC curve is constructed from prediction scores obtained by performing several classifications, each time moving the score threshold along the feature axis. Good AUC values range from 0.5 to 1, the latter being the maximum and meaning perfect classification of elements belonging to both positive and negative classes.

Finally, the last indicator used is the misclassification rate, that is the ratio between the number of misclassifications done by the classifier over the total number of items. Classifier performance was evaluated for each subject, by calculating the aforementioned parameters, both for training and testing sets. Given the scarcity of data available for each subject, averaged performance results were calculated among all the partition folds, to reduce bias and polarization that may affect our evaluation of the classifier (Table 2).



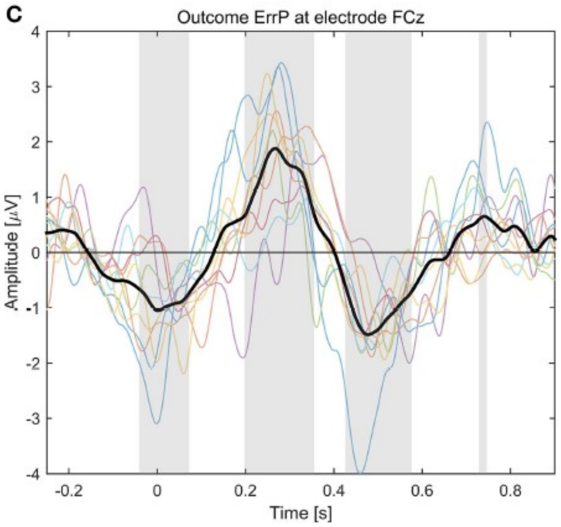
setname	AUCtrain	AUCtest	misrate_train[%]	misrate_test[%]	Accuracy_Train	Accuracy_Test	Sensitivity_train	Sensitivity_test	Specificity_train	Specificity_test	F1-score_train	F1-score_test
Subject1	0,85	0,61	20,17	42,37	0,80	0,58	0,80	0,61	0,79	0,52	0,83	0,63
Subject2	0,96	0,72	7,23	27,10	0,93	0,73	0,92	0,46	0,93	0,81	0,85	0,42
Subject3	0,73	0,53	29,77	46,17	0,70	0,54	0,72	0,59	0,68	0,46	0,74	0,60
Subject4	0,63	0,51	38,00	48,43	0,62	0,52	0,63	0,55	0,61	0,46	0,66	0,57
Subject5	0,64	0,54	38,26	47,98	0,62	0,52	0,63	0,54	0,60	0,49	0,66	0,57
Subject6	0,61	0,55	39,68	46,18	0,60	0,54	0,60	0,55	0,61	0,53	0,64	0,58
Subject7	0,73	0,66	33,09	40,40	0,67	0,60	0,66	0,60	0,68	0,59	0,70	0,63
Subject8	0,73	0,62	30,91	41,50	0,69	0,59	0,72	0,63	0,65	0,52	0,74	0,64
Subject9	0,69	0,53	34,93	49,00	0,65	0,51	0,64	0,53	0,66	0,49	0,69	0,56
Subject10	0,55	0,45	44,16	53,98	0,56	0,46	0,55	0,46	0,58	0,46	0,59	0,50
means	0,71±0,12	0,57±0,07	31,62±10,8	44,31±7,27	0,68±0,11	0,56±0,07	0,69±0,11	0,55±0,06	0,68±0,11	0,53±0,10	0,71±0,08	0,57±0,07

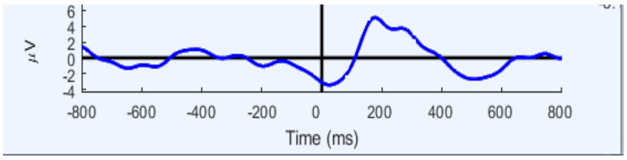
Table 2 - *Table of performance metrics, calculated for all subjects.*

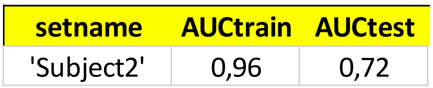
5.2 Discussion of Results

As can be seen from the bottom row of Table 2, particularly the AUC, acceptable results were obtained in training phase (AUC=0,71), while questionable in testing phase (AUC=0,57), but seemingly in line with results in literature: for example, Chavarriaga and colleagues **[22]**, in their study of ErrP single trial detection, achieved an average recognition rate of 0,64, whereas Spuler et al. **[19]**, in their classification studies for Outcome and Execution Error, obtained AUCs of 0,692 and 0,66, respectively.

One issue that impacted indirectly on classification results was the impossibility to detect common ERPs waveforms specific for each condition (‘Turn’ or ‘Pass’), across all subjects. In fact, by unrelated and preliminary examination (temporal filtering, spatial average referencing and epoching performed in EEGLAB environment), followed by visual inspection and comparison of Grand Average ERPs across all subjects, massive heterogeneity was found, and those ERPs where incompatible with those found in literature, in terms of morphology and latency. The only exception were the results obtained from ‘Subject2’ dataset, in terms of Average ErrPs and AUC values.







setname	AUCtrain	AUCtest
'Subject2'	0,96	0,72

Figure 27 - *Left: Outcome ErrP;Top right: grand Average ERP from ‘Subject2’ set at Fz electrode, «Pass» condition, obtained with EEGLAB; Bottom right: AUC results for ‘Subject2’ set*

As shown from Figures 27 (left) and 27 (top right) the shape and latencies of the morphological components of the grand Average ERP, coincide with the Outcome ErrP obtained by Spuler et al. **[19]**. Therefore, even if not perfect, they are the indication that we are on the right path. As further proof, the corresponding AUC values (Figure 27, bottom right) are more than acceptable, achieving an almost perfect score in training.

Many reasons may have affected the overall classification results:

- *Hardware limitations*: A very important factor surely is the hardware limitation. Only eight active electrodes were available for the configuration, quite sub-standard if compared to similar EEG-BCI studies, in which the minimum number of electrodes utilized is 64 **[22].**
- *No bad data removal before processing.* The datasets recorded for this study are all plagued by noise and movement artifacts (some less, some more extensively), as can be seen from Figures 28 a) and b), which represent only an example of this issue.

Figure 28 b) was obtained in EEGLAB environment after bandpass filtering and epoching, using the same parameters of the proposed processing pipeline, while, concerning Figure 28 a), only epoching was applied. Considering epoch 90, a huge negative drift in the signal propagates across all channels; moreover, spikes appear in epochs 87 and 88. These drifts are most likely due to stretching and movements of electrodes all over the cap, (particularly after blinking), or by increased electrode-skin impedance due to conductive gel drying up. As can be seen, these artifacts persist even after temporal filtering and, regardless of event type, these noisy epochs enter training or validation sets during classification, worsening overall ERP discrimination performance.

- *Non-uniform ERP latencies, both intra-subject (among epochs of same event) and inter subject.* As already stated, visual inspection of Grand Average ERPs for each subject showed some degrees of heterogeneity. Even postulating common polarity of ERP components, elicited during same typology of trials, their lack of respective synchrony (i.e., sometimes the ERP appears a little before the trigger, sometimes way after) may lead, to irrelevant oscillations or, worst case scenario, to reciprocal cancellation while performing the average.
- *Same windowing for each subject ([-100 600] ms, optimal window changes subject by subject* **[22]***).* In the proposed pipeline, same epoch window was applied for all subjects; however, this might not have been the most fruitful choice, because, as also stated in the previous point, ERP latencies differ from

subject to subject (for example some user got a hold on the wheelchair's next movement, way before pressing the key).

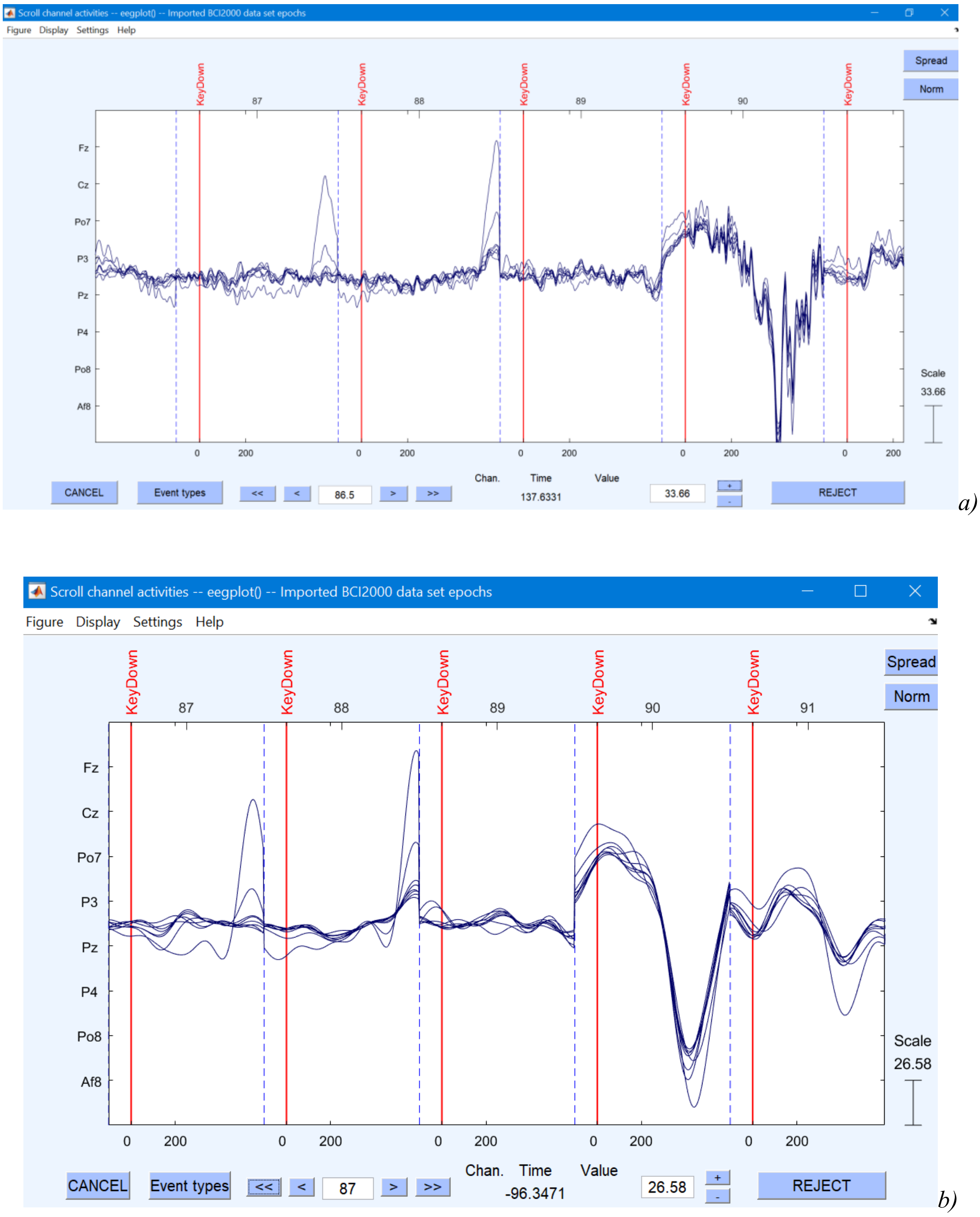


Figure 28 - *Examples of signal drift: pre a) and after temporal filtering. Note the drift in epoch 90, and artifacts peaks in epoch 87 and 88.*

Conclusions

The application field of Brain-Computer Interfaces has sparked the interest of many researchers throughout the years, particularly about the Brain Computer Wheelchair, a sensor-embedded wheelchair set to be controlled by the user via his brain signals and used in the context of indoor navigation, as a mean of restoring some degree of autonomy of movement to quadriplegic people.

Several BCW implementations proved to be successful in solving the navigation problem, but all lacking a failsafe system for wrong sensor readings, or avoidance of any kind of danger otherwise not detected. A recent implementation involved the use of an ErrP-detecting BCW that would send a feedback signal to the wheelchair control system, whenever a danger is detected by the user and thus an ErrP is generated. The trigger would prompt a trajectory modification to avoid the user-detected obstacle. The only drawback of this approach is that it was tested in a simulated environment, and using data not acquired with an experimental paradigm that does mirror the proposed BCW setting.

To fill this gap, in this thesis, a processing pipeline was presented, from data acquisition to temporal filtering and epoching, up to ERP epoch classification. By using video simulation of wheelchair system in motion, EEG data was recorded from ten subjects and was processed to extract suitable ERP for feedback.

Classification results were heavily affected by electrode movement artefacts and lack of specific ERP waveform regardless of eliciting trial type and, although they are seemingly in line with those found in literature, there is some room for future improvements. For example, criteria for bad data rejection could be implemented, prior to processing. Firstly, by rejecting activity coming from electrode Af8, since it is positioned above the eye, whose movements and blinking would cause important noise artefacts, that also affect simultaneously the other channels.

Independent component analysis (ICA) **[40]** could be an effective tool for such task, but after some tests (not reported in this thesis), the author has concluded that its application is too much time-consuming, also considering future real time applications.

In this perspective, blocks of incoming data can be scanned using a sliding window approach, and then any EEG data segment that overcomes a certain amplitude threshold or other criteria (e.g., entropy, signal variance, etc.), can be excluded a priori from the processing. The same reasoning can be applied to the classification step only: instead of fixing the epoch window, sliding, overlapping windows can be used to detect ERPs presence and the classifier can be modified accordingly. Moreover, instead of relying only on time-domain, frequency-domain (e.g., spectral power) or other non-linear features (e.g., entropy) can be exploited, as they can provide better ERP discrimination results **[19]**.

The author hopes that this work could be used as a stepping-stone for new investigations concerning BCWs and improvements in obstacle avoidance strategies, followed by real-time application studies and, in the long term, improve quality of life of disabled people.

Bibliography

[1] Ferracuti, F., Freddi, A., Iarlori, S., Longhi, S., Monteriù, A., & Porcaro, C., "Augmenting robot intelligence via EEG signals to avoid trajectory planning mistakes of a smart wheelchair"., *Journal of Ambient Intelligence and Humanized Computing*, *1*, 1–13, 2021.

[2] He B. "Neural Engineering" ,3rd ed. Springer, 2020.

[3] Purves D., et al. "Neuroscience. 6th Edition", Sinauer Associates, New York, 2018;

[4] Kandel ER, Schwartz JH, Jessell TM, Siegelbaum S, Hudspeth AJ. "Principles of neural science"., 5th ed. New York/London: McGraw-Hill; 2013.

[5] Hu, L., & Zhang, Z. "EEG signal processing and feature extraction". Springer, 2019

[6] D. Purves, G.J. Augustine, D. Fitzpatrick, W.C. Hall, A.-S. LaMantia, J.O. McNamara, and S.M. Williams, "Neuroscience", 3° ed., Sinauer Associates, Inc., 2004, Ch. 27 p. 669,

[7] J. Wolpaw, E. W. Wolpaw, "BCI principle and practice"*,* Oxford University *Press*, 2012.

[**8**] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: Assessing the speed of a P300-based brain- computer interface," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 174–179, 2000.

[**9**] Hu L, Mouraux A, Hu Y, Iannetti GD., "A novel approach for enhancing the signal-to-noise ratio and detecting automatically event-related potentials (ERPs) in single trials", *NeuroImage*,50:99–111, 2010

[**10**] Rugg MD, Coles MGH. "Electrophysiology of mind: event-related brain potentials and cognition. Oxford: Oxford University Press; 1995.

[11] Dawson GD. "A summation technique for detecting small signals in a large irregular background". *J Physiol.*;115(1):2p–3p., 1951

[12] Dawson GD. A summation technique for the detection of small evoked potentials. *Electroencephalogr Clin Neurophysiol.*, 6(1):65–84, 1954.

[13] Clerc, Maureen & Bougrain, Laurent & Lotte, Fabien. "Brain-Computer Interfaces 1: Foundations and Methods",Ch.4, p. 80, 2016.

[14] B. Rebsamen, "A Brain Controlled Wheelchair to Navigate in Familiar Environments", PhD thesis, National University of Singapore, Ch2 p. 20, 2008.

[15] Chavarriaga, R., Sobolewski, A., Millán, J. D. R., Zander, T. O., Kübler, A., & Wagner, J.. "Errare machinale est: the use of error-related potentials in brain-machine interfaces*",* 2014.

[16] Falkenstein,M.,Hohnsbein,J.,Hoormann,J.,andBlanke,L, "Effects ofCross modal divided attention on late erp components.ii. error processing in choice reaction tasks". *Electroencephalogr.Clin.Neurophysiol.* 78,447–455, 1991.

[17] Gehring,W.J.,Goss,B.,Coles,M.G.,Meyer,D.E., and Donchin, E. A neural system for error detection and compensation. *Psychol.Sci.* 4,385–390, 1993.

[18] Kumar, A., Gao, L., Pirogova, E., & Fang, Q. A Review of Error-Related Potential-Based Brain-Computer Interfaces for Motor Impaired People. *IEEE Access*, *7*, 142451–142466, 2019.

[19] Spüler, M., & Niethammer, C., "Error-related potentials during continuous feedback: Using EEG to detect errors of different type and severity". *Frontiers in Human Neuroscience*, *9*(MAR), 1–10., 2015.

[20] Ferrez, P. W., & del R. Millán, J. "Error-related EEG potentials generated during simulated brain-computer interaction". *IEEE Transactions on Biomedical Engineering*, *55*(3), 923–929, 2008.

[21] Holroyd, C.B. and Coles, M.G., "The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity", *Psychol.Rev*. 109:679, 2002.

[22] Chavarriaga, R., & Millán, J. D. R., "Learning from EEG error-related potentials in noninvasive brain-computer interfaces". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4), 381–388, 2010.

[23] Milekovic, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., and Mehring, C. "Error-related electrocorticographic activity in humans during continuous movements"., *J.NeuralEng. 9:026007,* 2012.

[24] Iturrate, I., Montesano, L., & Minguez, J. "Task-dependent signal variations in EEG error-related potentials for brain-computer interfaces". *Journal of Neural Engineering*, 10(2), 2013.

[25] Ferracuti, F., Casadei, V., Marcantoni, I., Iarlori, S., Burattini, L., Monteriù, A., & Porcaro, C. "A functional source separation algorithm to enhance error-related potentials monitoring in noninvasive brain-computer interface". *Computer Methods and Programs in Biomedicine*, *191*, 105419, 2020.

[26] Chavarriaga, R., Khaliliardali, Z., Gheorghe, L., Iturrate, I., & Millán, J. D. R. "EEG-based decoding of error-related brain activity in a real-world driving task.", *Journal of Neural Engineering*, 12(6), 66028, 2015.

[27] Pires, G., Nunes, U., & Castelo-Branco, M., "Statistical spatial filtering for a P300-based BCI: Tests in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis. Journal of Neuroscience Methods, 195(2), 270–281, 2011.

[28] Rebsamen, B., Guan, C., Zhang, H., Wang, C., Teo, C., Ang, M. H., & Burdet, E. "A brain controlled wheelchair to navigate in familiar environments", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(6), 590–598, 2010.

[29] Tanaka, K., Matsunaga, K., & Wang, H. O., "Electroencephalogram-based control of an electric wheelchair" *IEEE Transactions on Robotics*, 21(4), 762–766, 2005.

[30] Iturrate, I., Antelis, J. M., Kübler, A., & Minguez, J., "A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation. IEEE Transactions on Robotics, 25(3), 614–627, . 2009.

[31] O.Karameldeen, A.Monteriù, F.Ferracuti, "Correcting Mobile Robot Path-Planning Mistakes in Real Time Using EEG Signals", thesis for Master Degree in Biomedical Engineering, a.a. 2017/2018

[32] Navigation Stack. URL: *navigation/Tutorials/RobotSetup - ROS Wiki*

[33] Image avaliable at: https://www.gtec.at/product/ggammasys/#:~:text=g.GAMMAcap%20is%20designed%20for%20maximum%20comfort%20with%20minimal,additional%20intermediate%20positions%20are%20precut%20for%20easy%20placement.

[34] L. Ciabattoni, F. Ferracuti, A. Freddi, S. Iarlori, S. Longhi, and A. Monteriu, "ErrP Signals Detection for Safe Navigation of a Smart Wheelchair," *2019 IEEE 23rd Int. Symp. Consum. Technol. ISCT 2019*, pp. 269–272, 2019.

[35] URL: *https://commons.wikimedia.org/wiki/File:International_ 10-20_system_for_EEG-MCN.svg*

[36] Delorme A & Makeig S , "EEGLAB: an open-source toolbox for analysis of single-trial EEG dynamics", *Journal of Neuroscience Methods* 134:9-21, 2004.

[37] Hoffmann, U., Garcia, G., Vesin, J. M., & Ebrahimi, T., “Application of the evidence framework to brain-computer interfaces. Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, 26 I(February), 446–449, 2004.

[38] Hoffmann, U., Vesin, J. M., Ebrahimi, T., & Diserens, K., “An efficient P300-based brain-computer interface for disabled subjects”, *Journal of Neuroscience Methods*, 167(1), 115–125, 2008.

[39] MacKay DJ, “Bayesian interpolation.”, *Neural Comput*, 4:415–447, 1991.

[40] URL: d. Indep. Comp. Analysis - EEGLAB Wiki, consulted on 11th of February 2022.