HAL
open science

A Brain Controlled Wheelchair to Navigate in Familiar Environments

Brice Rebsamen

To cite this version:

Brice Rebsamen. A Brain Controlled Wheelchair to Navigate in Familiar Environments. Human-Computer Interaction [cs.HC]. national university of singapore, 2009. English. NNT : . tel-00459007

HAL Id: tel-00459007

https://theses.hal.science/tel-00459007

Submitted on 23 Feb 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A BRAIN CONTROLLED WHEELCHAIR TO NAVIGATE IN FAMILIAR ENVIRONMENTS

BRICE REBSAMEN

(M.Sc., ENSEIRB, France)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

ACKNOWLEDGMENTS

I would like to thank Prof. Etienne Burdet, Prof. Marcelo Ang and Prof. Teo Chee Leong, my NUS supervisors, for their many valuable suggestions and constant support during this research.

Prof. Christian Laugier, my French supervisor, for sending me to NUS for this project, for inviting me to various seminars and conferences, and for welcoming me in his team.

My friend and collaborator, Dr. Zeng Qiang, who contributed a lot to the development of a working wheelchair prototype, and who helped me so many times with my experiments.

Dr. Guan Cuntai, Dr. Zhang Haihong and Dr. Wang Chuanchu from the Neural Signal Processing Lab at I^2R, for helping me with the EEG hardware and letting me hack their BCI code.

Prof. J. Edward Colgate and Prof. Michael Peshkin, for kindly welcoming me in their lab (LIMS: Laboratory for Intelligent Mechanical Systems, Northwestern University, Evanston, IL, USA) to test the elastic path controller on the scooter COBOT. Dr. Eric Faulring gave me selfless help during my stay at the LIMS.

My friend and collaborator, Mr. Long Bo, who came with me to LIMS to help me with the experiments experiments on the scooter COBOT there.

And finally all the people who agreed to participate in my experiments.

Without the help of the people mentioned above, this work would never have come into existence.

TABLE OF CONTENTS

Page

Acknowledgments i

Summary viii

List of Tables x

List of Figures xi

Chapters::

1. Introduction 1

    1.1 Motivation 1

    1.2 Objectives and Scope 2

    1.3 Design Constraints 4

    1.4 Overview of Our Strategy 6

    1.5 Organization of the Thesis 7

    1.6 Contributions 8

2. Literature Review 9

2.1 Recording the Brain Activity 9

2.1.1 Invasive Methods 9

2.1.2 Blood Flow Based Methods 10

2.1.3 Electromagnetic Based Methods 11

2.1.4 Summary 12

2.2 EEG-based BCIs 13

2.2.1 Slow Cortical Potential (SCP) 13

2.2.2 P300 15

2.2.3 μ and β Rhythms 20

2.2.4 Steady-States Visually Evoked Potentials (SSVEP) 20

2.2.5 Mental State Recognition 23

2.2.6 EEG-BCIs for the Severely Disabled 23

2.3 Review of Other Brain Controlled Wheelchairs 24

2.3.1 Tanaka et al. 24

2.3.2 Minguez et al. 25

2.3.3 MAIA 26

2.3.4 Toyota/Riken 29

3. Hardware and Software Description 31

3.1 Hardware Description 31

3.2 Bar Code Based Global Positioning System 33

3.3 Software Description 34
3.3.1 Computing Platforms 34
3.3.2 Control Software Description 35

4. Motion Guidance 40

4.1 Path Following Controller 40
4.1.1 Kinematics 40
4.1.2 Path Controller 42
4.2 Evaluation of the Motion Guidance Controller 43
4.2.1 The Scooter COBOT 43
4.2.2 The Collaborative Wheelchair Assistant (CWA) 44
4.2.3 Motion Guidance Reduces Control Effort 44
4.3 Elastic Path Controller 46
4.4 Designing Paths 47

5. BCI For Destination Selection 54

5.1 Details of the P300 BCI 54
5.2 Experimental Procedure 58
5.3 Score Distributions 60
5.4 Performance Study 61
5.5 Calibration 66
5.6 Evaluation of the P300 BCW 68
5.7 Locking the Interface 69

6. Faster BCI for Stopping 70

6.1 Stopping with a P300-BCI 71

6.1.1 Threshold-Based Algorithm 72

6.1.2 Threshold-Less Algorithm 73

6.1.3 Conclusion 76

6.2 Stopping with a μ/β-BCI 76

6.3 Hybrid Interface 79

6.4 On-line Evaluation 81

6.5 Comparison of Off-line and On-line Results 81

6.6 Comparison of the μ/β and P300 BCIs for Stopping 85

7. Overall Strategy Evaluation 88

7.1 Experimental Setup 88

7.2 Evaluation Metrics 89

7.3 Results 90

7.4 Comparison 91

7.5 Discussion 94

8. Conclusion 95

8.1 Summary 95

8.2 Discussion 97

8.3 Recommended Directions 98

Bibliography 102

Appendices 109

Appendices:

A. Mathematics of the Path Following Controller 110

A.1 Coordinates Transformation 110

A.2 Time independent equations 113

B. Code for Setting Up the Real Time Architecture 115

C. List of Publications 120

SUMMARY

The Brain Controlled Wheelchair (BCW) is a simple robotic system designed for people, such as locked-in people, who are not able to use physical interfaces like joysticks or buttons. Our goal is to develop a system usable in hospitals and homes with minimal infrastructure modifications, which can help these people regain some mobility.

The main challenge is to provide continuous and precise 2D control of the wheelchair from a Brain Computer Interface, which is typically characterized by a a very low information transfer rate. Besides, as design constraints, we want our BCW to be safe, ergonomic and relatively low cost. The strategy we propose relies on 1) constraining the motion of the wheelchair along predefined guiding paths, and 2) a slow but accurate P300 EEG brain interface to select the destination in a menu.

This strategy reduces control to the selection of the appropriate destination, thus requires little concentration effort from the user. Besides, the trajectory is predictable, which contributes to reduce stress, and eliminates frustration that may be associated with trajectories generated by an artificial agent. Two fast BCIs are proposed to allow stopping the wheelchair while in motion. A hybrid BCI was developed to combine the slow P300 BCI used for destination selection with a faster modality to stop the wheelchair while in motion.

Experiments with healthy users were conducted to evaluate performances of the BCIs. We found that after a short calibration phase, the destination selection BCI allowed the choice of a destination within 15 seconds on average, with an error rate below 1%. The faster BCI used for stopping the wheelchair allowed a stop command to be issued within 5 seconds on average. Moreover, we investigated whether performance in the STOP interface would be affected during motion, and found no alteration relative to the static performance.

Finally, the overall strategy was evaluated and compared to other brain controlled wheelchair projects. Despite the overhead required to select the destination on the interface, our wheelchair is faster than others (36% faster than MAIA): thanks to the motion guidance strategy, the wheelchair always follows the shortest path and a greater speed is possible. Comparison was also performed using a cost function that takes into account traveling time and concentration effort; our strategy yields by far the smallest cost (the best other score is 72% larger).

This work resulted in a novel brain controlled wheelchair working prototype. It allows to navigate in a familiar indoor environment within a reasonable time. Emphasis was put on user's safety and comfort: the motion guidance strategy ensures smooth, safe and predictable navigation, while mental effort and fatigue are minimized by reducing control to destination selection.

LIST OF TABLES

Table **Page**

2.1 Comparison of brain recording technologies 13

5.1 Performances at lowest cost point. 63

6.1 Cost analysis for algorithms 1a-d. 73

6.2 Cost analysis for algorithms 2a and 2b. 76

6.3 RT statistics (in seconds) with the μ/β stop BCI. 79

6.4 Off-line and on-line performances for the P300 and μ/β stop BCIs. . 85

7.1 Metrics to evaluate the overall strategy. 91

7.2 Comparison of strategy costs. 93

LIST OF FIGURES

Figure **Page**

1.1 Photograph of the prototype BCW. 3

2.1 Equipment to record the brain activity. 14

2.2 The Thought Translation Device. 16

2.3 P300 speller by Farwell and Donchin. 17

2.4 The P300 signal in EEG data. 18

2.5 2D control of a cursor using a $\mu\beta$-BCI. 21

2.6 EEG spectrum in a SSVEP BCI. 22

2.7 Tanaka's brain controlled wheelchair. 25

2.8 Minguez's brain controlled wheelchair. 27

2.9 MAIA's brain controlled wheelchair. 28

2.10 Toyota/Riken's brain controlled wheelchair. 30

3.1 BCW prototype. 32

3.2 Functional diagram of the BCW. 36

3.3 Functional diagram of the BCW software system. 37

4.1 Frames and notations for the controller. 41

4.2 Wheelchair's kinematics. 42

4.3 The scooter COBOT 45

4.4 Example of a map with guiding paths in a home environment. 48

4.5 A path obtained from WTP. 49

4.6 Fitting a B-Spline to a path. 51

5.1 P300 BCI visual stimulator. 56

5.2 Segmentation of the EEG signal in labeled samples. 57

5.3 Detection of P300 signals in the EEG. 59

5.4 SVM scores distributions. 61

5.5 RT distribution for the P300 BCI. 64

5.6 Influence of threshold on performances of the P300 interface. 65

5.7 Influence of depth on performances of the P300 interface. 67

6.1 Evaluation of the threshold-based P300 stop algorithms. 74

6.2 Evaluation of threshold-less P300 stop algorithm. 75

6.3 Distribution of RT and FA for P300 stop BCI. 77

6.4 RT distribution for the μ/β-BCI. 78

6.5 Control diagram with the hybrid BCI. 80

6.6 P300 stop BCI on-line evaluation 82

6.7 Off-line vs on-line comparison for the μ/β stop BCI. 83

6.8 Off-line vs on-line comparison for the P300 stop BCI. 84

6.9 Comparison of μ/β and P300. 86

7.1 Overall strategy test. 90

7.2 MAIA results. 92

8.1 Emotiv EPOC brain sensor. 99

8.2 Beagleboard and Gumstix. 100

8.3 A wheelchair powering module by Ngee Ann Polytechnic, Singapore. . 101

A.1 Frames and notations for the controller. 110

A.2 Wheelchair's kinematics. 112

CHAPTER 1

Introduction

1.1 Motivation

Amyotrophic Lateral Sclerosis (ALS), brainstem stroke, brain or spinal cord injury, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases impair the neural pathways that control muscles or impair the muscles themselves. They affect nearly two million people in the United States alone [1, 2]. Those most severely affected may lose all voluntary muscle control and may be completely *locked-in* to their bodies.

Although there are no statistics available on the number of patients with locked-in syndrome, the locked-in population is growing due to advances in artificial respiration. One estimation based on National Institute of Health statistics on brain-stem strokes and survival information, puts the number at as many as 50,000 individuals in the United States alone.

In order to help physically challenged people control a computer, a communication device or a wheelchair, various input devices are available. This includes a simple stick held between the teeth, buttons and joysticks of various sizes that can be activated by various parts of the body, gaze tracking systems or head movement based systems

to enable control of a cursor on a screen. However, all those input devices are of no use to locked-in people.

The only alternative for locked-in people is to establish communication and control channels directly from the brain, bypassing the disfunctioning brain's normal output channels of peripheral nerves and muscles. In a Brain Computer Interface (BCI), signals from the brain are acquired and processed to extract specific features that reflect the user's intent. These features are then translated into commands to operate a device.

The Brain Controlled Wheelchair (see Figure 1.1) described in this thesis was designed to provide some motion capability to locked-in people.

1.2 Objectives and Scope

A common feature between all BCIs is that, since the recorded brain signal is very noisy and has a large variability, either the uncertainty on the command will be high, or the time between consecutive commands will be long, in the order of seconds. *Can such a poor signal be used to safely and efficiently control a wheelchair that requires a real-time specification of its position within the three dimensional space of planar motion?* This is the challenge we address in this thesis.

Numerous applications of BCIs are reported in the literature, mostly for communication or computer control. However, a brain controlled wheelchair implies more considerations:

- Safety: especially since it transports a particularly vulnerable person.
- Ergonomy: the wheelchair should provide intuitive and efficient navigation with a minimum of effort.

Figure 1.1: Photograph of the prototype Brain Controlled Wheelchair (BCW). The BCW is built on top of a standard powered wheelchair. An EEG cap is used to record the brain signal.

- Low cost: so that people who need it can afford it.

Our goal in this work is to propose a strategy to control a wheelchair from a BCI. This requires a robotic wheelchair able to assist the user with the navigation task, and a BCI together with a control scheme adapted to the task. All of these requirements should be achieved while respecting the above constraints.

While the signal processing part of the BCI was based on previous developments at the Institute for Infocom Research (I^2R) in Singapore, it was adapted here to the purpose of controlling a wheelchair safely and comfortably. Its specific properties were evaluated experimentally and analyzed, and the overall control integrated different modalities to yield an efficient solution for controlling the wheelchair.

1.3 Design Constraints

As mentioned above, to be successful, a robotic wheelchair has to fulfill the following constraints: safety, ergonomy and low cost.

Safety is a critical factor for any device operating near or with humans. A wheelchair, although usually operating at low speed, has to comply to many safety requirements before being allowed on the market, for it transports a particularly vulnerable person. Robotic wheelchairs [3–10] generally rely on multiple sensors which provide information about the surrounding area to the navigation module which then decides what is the best course of action to safely reach destination. The question is to which extent can a user trust a robot, which perception and inference capacities remain low to this date? Although avoiding collision with walls, furniture and other obstacles is a relatively easy task for modern robots, avoiding stairs, bumps and unstable grounds, zones with low ceilings, proximity to dangerous areas (a fireplace

for instance), etc., is a complex problem. Some of these situations might be very difficult to detect by general sensors, or many specialized sensors would be needed to detect each of them. The question is even more relevant for a brain controlled wheelchair since it is designed to transport a locked-in person who may not have the ability to press an emergency stop button.

By **ergonomy** we mean that the wheelchair should allow the subject to reach destination with as little effort as possible. The later point is particularly important for a brain controlled wheelchair since using a BCI requires concentration and may prove exhausting. The control burden must be as light as possible, yet allow certain freedom to the user, such as stopping at any time during motion or change course. Besides, as for any robot designed to transport people, the trajectory should be smooth and correspond to the user's understanding of a trajectory as much as possible. Since human interpretation of the environment often differs from the robot's interpretation, the decision taken by the system might seem awkward to a human observer [11]. Moreover, autonomous vehicles have been observed to refuse to move forward due to some obstacles, while a human driver would easily be able to move its way through [8]. This undesirable behaviors may prove irritating and with time lead to the user stopping using the system.

Finally, the system should be available at a **low cost** so that people who need it can afford it. The BCI is already an expensive equipment; a powered wheelchair with the required amount of straps and cushions to support the user's body is also an expensive device. Therefore, the additional equipment, as well as the modifications required to mount the sensors, should not cost more than a fraction of the price of the wheelchair.

1.4 Overview of Our Strategy

Providing a robot that would respect all constraints mentioned above is the goal of many researchers in robotics. The biggest challenge is that the robot has to react appropriately to a wide variety of situations that occur while exposed to unconstrained environments. While improvements are made each year, solutions remain expensive, complex and unsatisfying.

However, the problem can be simplified by limiting the usage of the wheelchair to a few environments only: the user's home, office, care center, etc. By doing so, the environment can be learned in collaboration with a healthy human operator, thus eliminating the problem of detecting complex obstacles. Following that simplification, we decided to represent the environment by a network of paths connecting a finite but unlimited number of locations of interest for the user. These paths are human defined and stored in the system memory, and serve as guides for all subsequent motions. The trajectory is thus safe and natural, while no complex and expensive sensors are required.

To navigate with the wheelchair, the user simply selects the desired location while the wheelchair takes care of the whole trajectory by following the appropriate guiding path. The user can stop the wheelchair along the way, in case an unexpected obstacle appears on the path, or simply if he/she desires so. The control is thus limited to the initial selection of the destination and rarely issuing stop commands, hence minimizing the control effort.

This strategy therefore fulfils all constraints mentioned above. Safety is insured by the use of human designed paths, plus the supervision by the driver for unexpected situations. And since we are using only a few simple sensors the low cost constaint

is also respected. The ergonomy constraint is fulfilled by the use of human designed paths which provide smooth and natural trajectories, plus the destination selection control strategy that minimizes the control effort.

However, the usage of the wheelchair is limited to pre-defined trajectories and locations, and although new locations can be added at any time, the wheelchair does not offer the possibility to go everywhere the user would like to. Besides, it is assumed that the environment does not change, and especially that guide-paths are kept obstacle free, as the robot is not equipped with sensors that would allow it to detect obstacles. We think that this constraint is easily fulfilled since the wheelchair's motion is constrained to familiar environments: other person evolving in this environment will be aware of that constraint and voluntarily keep the guide-paths clear.

1.5 Organization of the Thesis

Chapter 2 reviews existing technology to record the brain activity and construct a BCI. We will then present other brain-controlled wheelchair projects.

The BCW hardware, the localization system we use, and the software architecture for real time control are described in Chapter 3.

Chapter 4 explains in detail the path following navigation system. After a brief mathematical description of the path following controller, we will detail the Elastic Path Controller, which was developed during this project to allow temporary escape of the guiding path upon user's instruction. We will present experiments that prove that motion guidance effectively simplifies motion control. Then we will see how to create and edit maps of guiding paths.

Chapter 5 describes the BCI we use in this project for destination selection and presents experimental results with able subjects using the interface.

Chapter 6 presents two fast BCIs to allow stopping the wheelchair in a decent time while in movement. A novel hybrid BCI, developed to combine the destination selection BCI and the fast BCI for stopping, is introduced. Off-line and on-line evaluation results are presented.

Chapter 7 evaluates the developed system and compares it to other brain controlled wheelchair projects.

1.6 Contributions

The major contributions of this thesis are:

- The control strategy itself, which provides a way for controlling a wheelchair from a low information transfer input device such as a BCI, safely and efficiently while requiring minimum effort from the user and a minimal amount of sensors.
- The development of a robotic wheelchair and its integration with a BCI, which demonstrated the first brain controlled wheelchair able to move in a building environment.
- The elastic path controller (EPC) which allows temporary escape from the guiding path, and used for on-line path editing.
- The evaluation of the existing P300 interface for item selection.
- The modification of the existing P300 BCI for stopping, and its evaluation.

This work resulted in many peer-reviewed publications. The list can be found in Appendix C.

CHAPTER 2

Literature Review

In this chapter we will see what are the different technologies available to record (Section 2.1) the brain activity. Then, in Section 2.2 we will review some EEG-based BCIs. Finally - in Section 2.3 - we will review other brain-controller wheelchair projects.

2.1 Recording the Brain Activity

The first step toward a BCI is recording the activity of the living brain. This can be done invasively by surgically implanting electrodes in the brain, or non-invasively. In this section we will review various brain imaging technologies.

2.1.1 Invasive Methods

Biologists can measure the potential at different parts of a single neuron in a culture. Recording neuron activity in a living brain is possible using surgically implanted micro-electrodes arrays, although it is no longer a single neuron recording but the activity of groups of neurons.

Monkeys with brain implants have been reported [12–14] to brain-control the displacement of a cursor on a screen or to control the motion of a robotic arm. Surgical implantation of electrodes is still consider too risky to be performed on

humans. However, some teams have had successful results with them: Kennedy [15] and Donoghue [16] reported successful brain-control of a mouse pointer on a computer screen with patients who had been implanted an electrode in the outer layer of the neocortex.

2.1.2 Blood Flow Based Methods

The typical blood flow based methods include Functional Magnetic Resonance Imaging (fMRI) and Near-Infrared Imaging.

Functional Magnetic Resonance Imaging (fMRI)

Functional Magnetic Resonance Imaging (fMRI) [17] is a relatively recent imaging technique that aims to determine the neuro-biological correlate of behavior by identifying the brain regions that become "active" during the performance of specific tasks in vivo.

The technique is based upon the different magnetic susceptibilities of the iron in oxygenated and deoxygenated hemoglobin. Oxygenated blood is diamagnetic and possesses a small magnetic susceptibility, while deoxygenation of hemoglobin produces deoxyhemoglobin, which is a significantly more paramagnetic species of iron. Blood Oxygenation Level Dependent (BOLD) measurements measure local variation in the relaxation time caused by variations in the local concentration of deoxygenated blood.

It has become the diagnostic method of choice for investigating how a normal, diseased or injured brain is working. The spatial resolution can be sub-millimeter with temporal resolutions on the order of seconds. The ability to measure solitary neural events is not yet possible but improvements in sensitivity have been made steadily over the past 10 years. Figure 2.1-b shows a typical fMRI machine.

Functional Near-Infrared Imaging (fNIR)

Functional Near-Infrared Imaging (fNIR) is a relatively novel technology based upon the notion that the optical properties of tissue (including absorption and scattering) change when the tissue is active. Two types of signals can be recorded: fast scattering signals, presumably due to neuronal activity [18] and slow absorption signals, related to changes in the concentration of oxy- and deoxy-hemoglobin [19]. However, fNIR lacks the spatial resolution of fMRI and cannot accurately measure deep brain activity.

The fast fNIR signal is measured as an "event-related optical signal" (EROS). The spatial localization of fast and slow fNIR measurements both correspond to the BOLD fMRI signal [20]. The latency in the slow (hemodynamic) signal roughly corresponds to that for the BOLD fMRI response [21].

The major limitation of optical methods (both fast and slow signals) is their penetration (max: approximately 3 cm from head to surface), which makes it impossible to measure brain structures such as the hippocampus or the thalamus, especially if they are surrounded by light-reflecting white matter. However, the vast majority of the cortical surface is accessible to the measurements. The technology is relatively simple and portable, and may serve a sort of portable, very rough equivalent of fMRI, which may supplement or substitute for some EEG measures.

Figure 2.1-a shows the setup typically used for NIR imaging.

2.1.3 Electromagnetic Based Methods

The currents generated by an individual neuron are too tiny to be recorded non-invasively, however excitatory neurons in the cortex all have their axon parallel one to

another and grouped in redundant populations called macro-columns [22] which act as macroscopic sources of electromagnetic waves that can be recorded non-invasively.

Magnetoencephalography (MEG)

Magnetoencephalography (MEG) [23–25] is an imaging technique used to measure the magnetic fields produced by electrical activity in the brain. Because of the low strength of these signals and the high level of interference in the atmosphere, MEG has traditionally been performed inside rooms designed to shield against all electrical signals and magnetic field fluctuations. Figure 2.1-c shows a typical MEG equipment.

Electroencephalography (EEG)

Electroencephalography (EEG) is the recording of electrical activity along the scalp produced by the firing of neurons within the brain [26, 27]. The recording is obtained by placing electrodes on the scalp with a conductive gel or paste. The number of electrodes depends on the application, from a few to 128, and they can be mounted on a cap for convenience of use (see Figure 2.1-d). The electric signal recorded is of the order of few microvolt, hence must be amplified and filtered before acquisition by a computer. The electronic hardware used to amplify, filter and digitize the EEG signal is of the size and weight of a book; it is easily transportable and relatively affordable. Spatial resolution is on the order of centimeters while the time of response to a stimulus is on the order of 100s of milliseconds.

2.1.4 Summary

Table 2.1 shows a comparison of the six methods presented above. Only NIR imaging and EEG can be used for a BCI: MEG and fMRI equipment is too expensive

Table 2.1: Comparison of brain recording technologies

Technology	Spatial Resolution (millimeters)	Temporal Resolution (seconds)	Size	Other Constraints
invasive	0.1	0.001	small	surgical risk
fMRI	1	5	large	high magnetic field
fNIR	1	5	small	-
EROS	1	0.05	small	-
MEG	10	0.01	large	shielding required
EEG	10	0.01	small	-

and cumbersome, and invasive methods are not safe enough yet. However, as NIR is a relatively new method, it is not as popular as EEG in BCI studies.

2.2 EEG-based BCIs

A Brain Computer Interface (BCI) is any system which can derive meaningful information directly from the user's brain activity in real time [28]. The most important applications of the technology are mainly meant for the paralyzed people who are suffering from severe neuromuscular disorders. Most BCIs use information obtained from the user's encephalogram (EEG), though BCIs based on other brain imaging methods are possible. This section briefly describes several EEG-based BCIs. The P300 BCI is described in detail in next section.

2.2.1 Slow Cortical Potential (SCP)

The Slow Cortical Potential (SCP) signal is the modulation of the global EEG potential (very low frequency). It is recorded by a single electrode at the top of the head. Because SCPs indicate the overall preparatory excitation level of a cortical network, they are universally present in the human brain, and therefore make them a

Figure 2.1: Equipment to record the brain activity: a)NIRS, b)FMRI, c)MEG, d)EEG.

good signal for BCIs. Healthy subjects as well as severely paralyzed patients can learn to self-control their SCPs when they are provided with visual or auditory feedback of their brain potentials and when potential changes in the desired direction are positively reinforced.

Birbaumer's team [29] in Tübingen University developed a brain computer interface device called the Thought Translation Device (TTD), in which the vertical position of a feedback cursor reflects the amplitude of an SCP shift. After a patient has achieved reliable control over his or her SCP shifts, the responses can be used to select items presented on a computer screen. A spelling program included in the TTD allows patients to select single letters by sequential selection of blocks of letters presented in a dichotomic structure with five levels (Figure 2.2): the left to right movement of the cursor is constant; the vertical movement is controlled by the user's brain activity. To improve speed of communication, this program has been supplemented by a dictionary offering word completion after only a few letters have been selected.

2.2.2 P300

The P300 evoked potential is a well studied and stable brain signal [30, 31] belonging to the Event Related Potential (ERP) group. It is a natural and involuntary response of the brain to rare or infrequent stimuli, which can provide a BCI through an oddball paradigm. In this paradigm a random sequence of stimuli is presented, only one of which is of interest to the subject. Around 300 milliseconds after the target is presented, a positive potential peak is recorded in the EEG signal. Upon

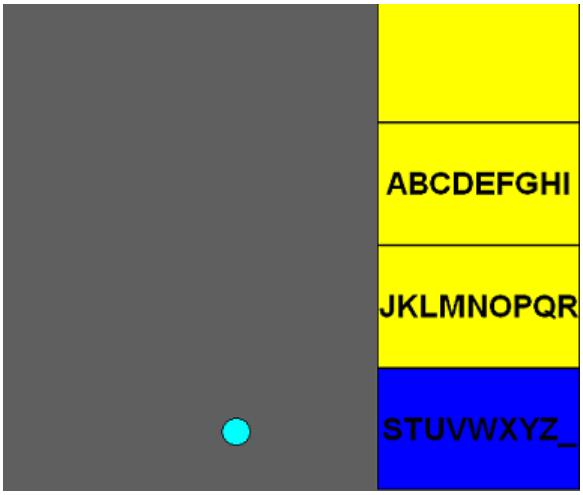


Figure 2.2: The Thought Translation Device [29]. The left to right movement of the cursor is constant; the vertical movement is controlled by the user's brain activity. A letter is selected by sequential selection of blocks of letters presented in a dichotomic structure.

detection of this P300 signal (P for positive, 300 for the 300ms delay), the target can be determined as the stimulus that occurred 300 ms earlier.

In 1988, Farwell and Donchin [32] developed the first P300 based BCI to select letters from a virtual keyboard (see Figure 2.3). Items are presented on a 6 by 6 matrix; rows and columns are flashed in a random sequence, eliciting a P300 signal 300 ms after the key the user wants to select has been flashed.

P300 Signal Detection

Given the importance of the P300 signal in this thesis, we will present here a short review of methods to detect it. The main difficulty with the P300 signal is that the signal to noise ratio is very low. The top panel of Figure 2.4 shows the raw EEG signal from ten electrodes. The vertical lines mark the times of stimuli, the red/thick

Figure 2.3: In the P300 speller by Farwell and Donchin [32], items are presented on a 6 by 6 matrix. Rows and columns are flashed in a random sequence, eliciting a P300 signal 300 ms after the key the user wants to select has been flashed.

line corresponding to a target stimulus. The P300 signal cannot be seen with the naked eye in the EEG.

Traditionally, ERPs are synchronously averaged to enhance the evoked signal and suppress the background brain activity [33]. This way uncorrelated noise is canceled out and the P300 signal appears more clearly as can be seen on bottom panel of Figure 2.4. Once the signal to noise ratio has been enhanced, the P300 signal can be detected. For instance, Farwell and Donchin [32] used step-wise discriminant analysis (SWDA) followed by peak picking and evaluation of the covariance. Alternatively, the discrete wavelet transform can also be added to the SWDA to localize efficiently the ERP components in both time and frequency [34].

Independant component analysis (ICA) was first applied to ERP analysis by Makeig et al. [35]. Infomax ICA [36] was used by Xu et al. [37] to detect the ERPs for

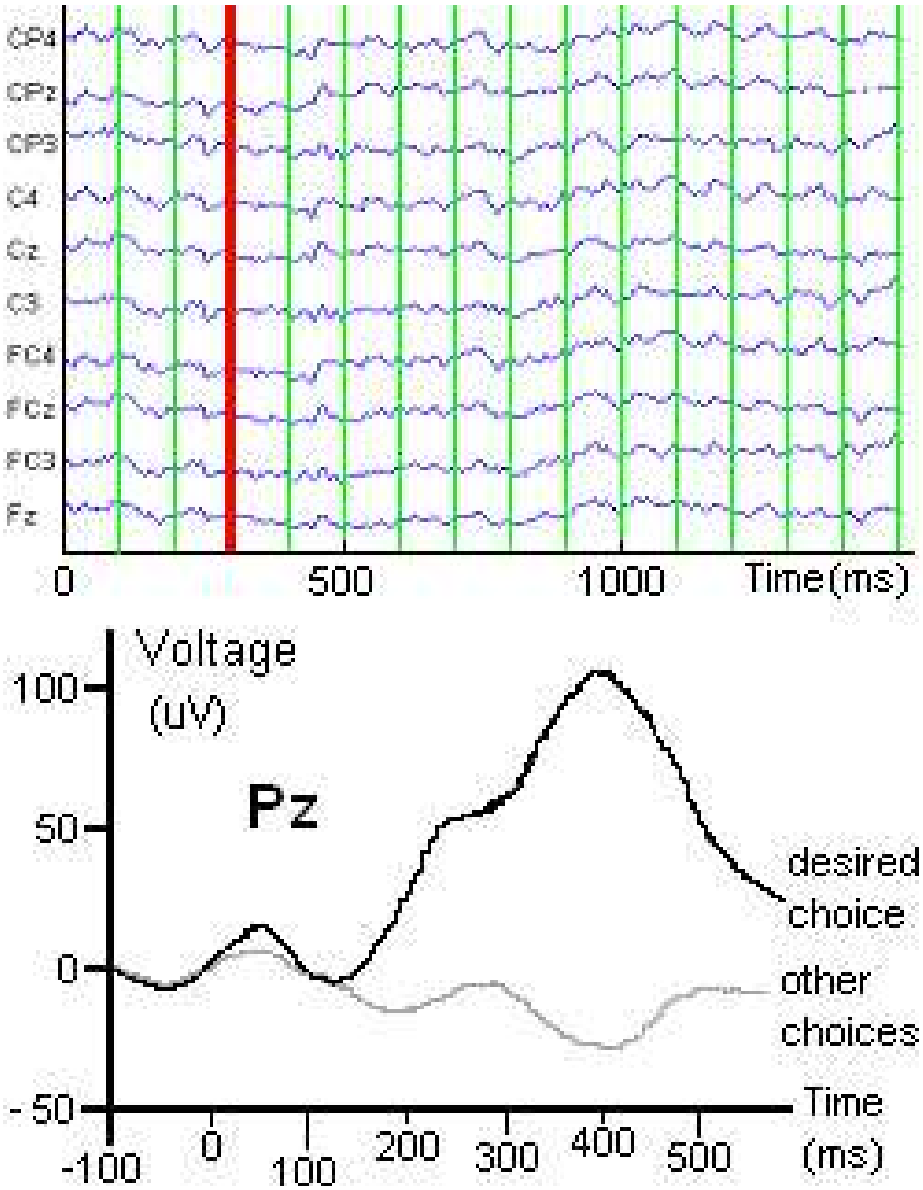


Figure 2.4: The P300 signal in an EEG. Top panel: raw EEG signal from ten electrodes; the vertical lines mark the times of stimuli, the red/thick line corresponding to a target stimulus. Note that the P300 signal is not visible as the signal to noise ratio is very low. After averaging however, uncorrelated noise is canceled out and the P300 appears clearly (bottom panel).

the P300-based speller. In their approach, those idependent components with relatively larger amplitudes in the latency range of P300 were kept, while the others were set to zero. Also, they exploited a priori knowledge about the spatial information of the ERPs and decided whether a component should be retained or wiped out.

ICA has also been used for the detection of P300 signals by Serby et al. [38]. Their work involved the application of a matched filter together with averaging and using a threshold technique for detecting the existence of the P300 signals. The IC corresponding to the P300 source is selected and segmented to form overlapping segments from 100 to 600 ms. Each segment is passed through a matched filter to give one feature that represents the maximum correlation between the segment and the average P300 template.

The detection of ERPs from only a single-trial EEG is very favourable since online processing of the signals can be performed. Unlike the averaging (multiple-trial) [39] scheme, in this approach the shape of the ERPs is first approximated and then used to recover the actual signals. A decomposition technique that relies on the statistical nature of neural activity is one that efficiently separates the EEGs into their constituent components, including ERPs. A neural activity may be delayed when passing through a number of synaptic nodes, each introducing a delay. Thus, the firing instants of many synchronized neurons may be assumed to be governed by Gaussian probability distributions [33].

Adaptive filtering is also a popular approach. With the assumption that the ERP signals are dynamically slowly varying processes, the future realization are predictable from the past realizations. These changes can be studied using a state-space model.

Kalman filtering and generic observation models have been used to denoise the ERP signals [40].

2.2.3 μ and β Rhythms

Oscillatory activity in the brain is generated by feedback loops in complex neural networks. For example, synchronization of neuron assemblies gives rise to lower frequency of oscillations [41]. In EEG, μ (8-12Hz) and β (18-26Hz) rhythms measured on sensorimotor cortices are of particular interest.

The μ and β regulation is considered as an "operant conditioning" approach, i.e. it is of voluntary nature. Therefore the subject is free to think about anything or nothing until he or she decides to achieve control/communication through the interface.

Kuhlman showed in 1978 that people can learn to regulate the EEG power in the μ and β bands [42]. Recently, Wolpaw & McFarland designed a $\mu\beta$-BCI for 2D cursor control [43–45]. Using this BCI, four disabled subjects were allowed to move a cursor from the center of the screen to one of eight targets on the borders. Figure 2.5 shows the cursor's trajectories and times to target.

2.2.4 Steady-States Visually Evoked Potentials (SSVEP)

Steady-states visually evoked potentials (SSVEP) correspond to the response of the visual cortex to stimulation of the retina by a blinking light source. Figure 2.6 shows the amplitude spectrum of SSVEP in response to 7 Hz stimulation [46]. Three peaks at 7 Hz, 14 Hz, and 21 Hz can be found clearly. Panel (a) shows the single trial amplitude spectrum, while panel (b) shows the mean amplitude spectrum averaged over 40 trials. Vertical lines give standard deviation.

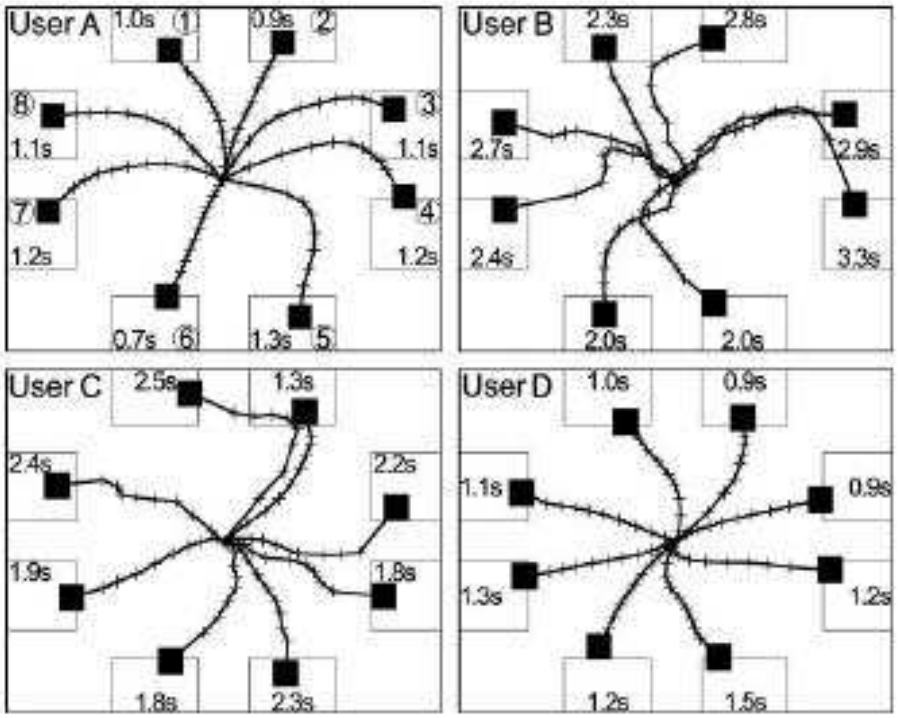


Figure 2.5: 2D control of a cursor using a $\mu\beta$-BCI by 4 disabled people (from Wolpaw et al. [45]). The subjects were instructed to move the cursor to one of eight targets: the figures show cursor's trajectories and times to target.

In a typical SSVEP-based BCI setup, an array of LEDs (or buttons on a computer screen), blinking at different frequencies and associated with commands, are disposed in the visual field of the subject. To select a command the user simply has to focus his attention to the desired button. As with the P300 signal, SSVEP is a natural response of the brain, which therefore does not require any training. Typical response time is in the order of a few seconds [47].

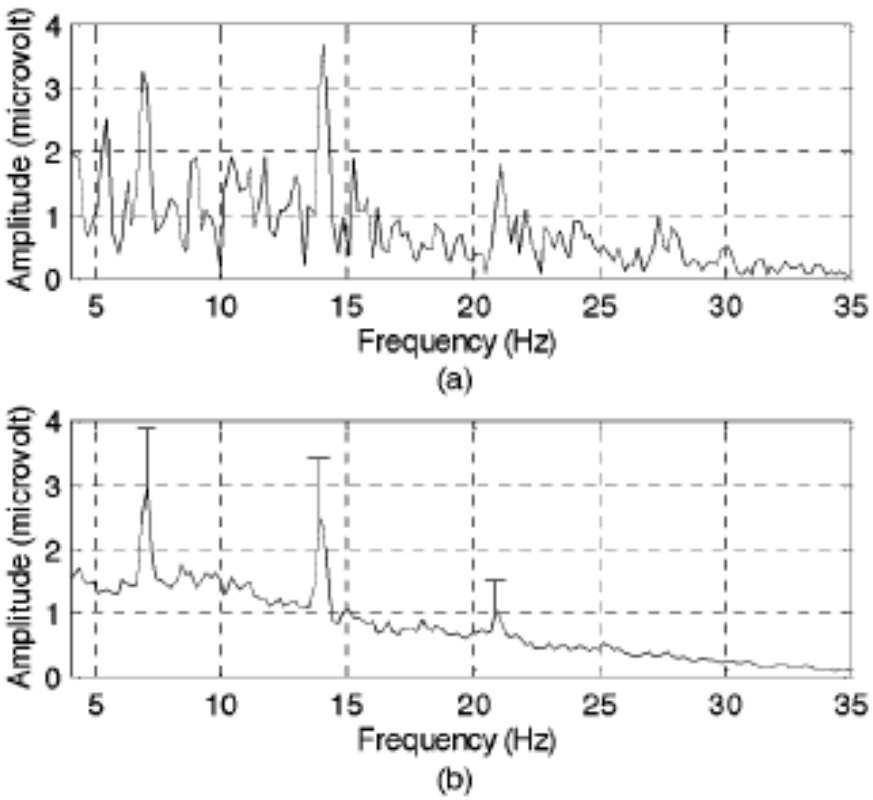


Figure 2.6: EEG spectrum corresponding to a 7 Hz stimulation in a SSVEP BCI (from [46]). Three peaks at 7 Hz, 14 Hz, and 21 Hz can be found clearly. Panel (a) shows the single trial amplitude spectrum, and panel (b) shows the mean amplitude spectrum averaged over 40 trials. Vertical lines give standard deviation.

2.2.5 Mental State Recognition

Millán & Mourino [48] designed the Adaptive Brain Interface (ABI) based on asynchronous recognition of three mental states. After a short evaluation, every user selects the three mental tasks that he/she finds easier out of the following choices: "relax", imagination of "left" and "right" hand (or arm) movements, "cube rotation", "substraction", or "word association". More specifically, the tasks consist of getting relaxed, imagining repetitive self-paced movements of the limb, visualizing a spinning cube, performing successive elementary subtractions by a fixed number (e.g., 64 0 3 = 61, 61 0 3 = 58, etc.), and concatenating related words. A neural network is trained to recognize the EEG pattern associated with each task.

ABI also recognizes an "idle" state when the user is not involved in any particular mental task, by using a statistical rejection criteria. In an asynchronous protocol, idle states appear during the operation of a brain-actuated device, while the subject does not want the BCI to carry out any action. Although the neural classifier is not explicitly trained to recognize those idle states, the BCI can process them adequately by giving no response.

ABI achieves error rates below 5% for three mental tasks, while correct recognition is 70% (or higher), producing an output every half second. In the remaining cases (around 20%-25%), the classifier does not respond, since it considers the EEG samples as uncertain ("idle" state).

2.2.6 EEG-BCIs for the Severely Disabled

Brain-Computer Interfaces are generally developed as a rehabilitation tool for locked-in people. Yet research is often conducted with healthy subjects, mostly for

practical reasons. In this section we will review the few papers that cover tests with severely disabled people.

One of the earliest study is by Birbaumer in year 2000, which showed that five patients suffering from end-stage ALS could use the TTD (introduced in Section 2.2.1). In 2003 six other patients confirmed those results [29].

Motor imagery based BCIs were also shown to work with severely disabled patients. Pfurtscheller and Neuper showed in 2001 that a C4/C5 tetraplegic patient could control the opening and closing of a hand orthosis [49]. In 2003, a patient with Severe Cerebral Palsy (SCP) could spell letters at a rate of one letter per minute [50]. And in 2005, four people severely disabled by ALS learned to operate such a BCI [51].

Recently, Sellers and colleagues evaluated a P300 BCI with ALS patients. In [52, 53] six ALS patients were trained and tested. They obtained similar classification results as non-ALS patients. Moreover, the study shows that those performances can sustain over several months without degradation.

2.3 Review of Other Brain Controlled Wheelchairs

In this section we will review four brain controlled wheelchairs developed by other groups.

2.3.1 Tanaka et al.

Tanaka et al. in [54] (2005) come with a discrete approach to the navigation problem: the environment is discretized in squares of 1m (see Figure 2.7) and the user is prompted 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 where to move next.

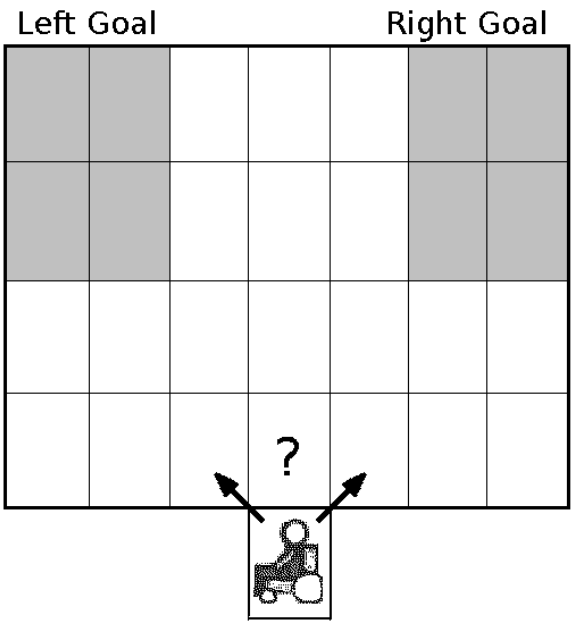


Figure 2.7: Tanaka's brain controlled wheelchair. By imagining left or right limb movements, the user decides the next move of the wheelchair (from [54]).

Although simple and safe, the system requires series of decisions to complete even a simple movement and may thus exhaust the subject. Therefore this strategy clearly breaks the ergonomy constraint, although it respects the safety and low cost constraints.

2.3.2 Minguez et al.

A similar principle was used in the sophisticated wheelchair system recently (2009) developed by Minguez et al. [55], 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 (see Figure 2.8), and these short term goals are reached automatically. As with Tanaka, the system requires a large number of steps to reach a destination, which might exhaust the

subject. For instance, Minguez reports that it took 11 minutes and 9 decision steps to realize a 40 meters long path with this system.

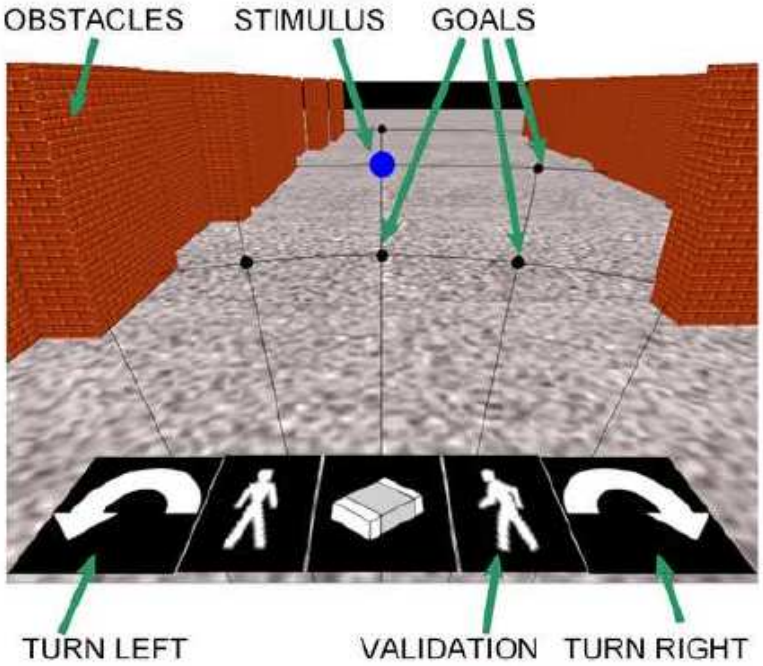
2.3.3 MAIA

In the MAIA (Mental Augmentation through Determination of Intented Action) project [56] (2007) the asynchronous IDIAP BCI [57,58] has been integrated with the intelligent wheelchair Sharioto of the KU Leuven [5].

The BCI continuously analyzes the user's EEG to detect a pattern associated with one of three learned mental states. After a short evaluation, the user selects the three mental tasks that he/she finds easier out of the following choices: "relax", imagination of "left" and "right" hand (or arm) movements, "cube rotation", "substraction", or "word association". More specifically, the tasks consist of getting relaxed, imagining repetitive self-paced movements of the limb, visualizing a spinning cube, performing successive elementary subtractions by a fixed number (e.g., 64-3=61, 61-3=58, etc.), and concatenating related words. This BCI achieves error rates below 5% for three mental tasks, while correct recognition is 70%. When the pattern in the EEG is not found to correspond to any learned pattern the BCI does not respond, thus artificially creating an idle state. The three mental states are associated with three commands for the wheelchair: forward, left and right. Every $500ms$ the BCI sends a probability distribution over the three commands to the wheelchair's shared control system, which translates it to a joystick-like command, i.e. a translational and rotational velocity.

Instead of executing the user's command immediately, this distribution is merged with the autonomous decision of the robot, based on sensors readings and activated

(a)



(b)

Figure 2.8: Minguez's brain controlled wheelchair (from [55]). (a) A snapshot of a participant navigating along a corridor. (b) Information represented in the visual display, which is an environment abstraction displayed from the users point of view. The user can select which goal to reach using a P300 BCI.

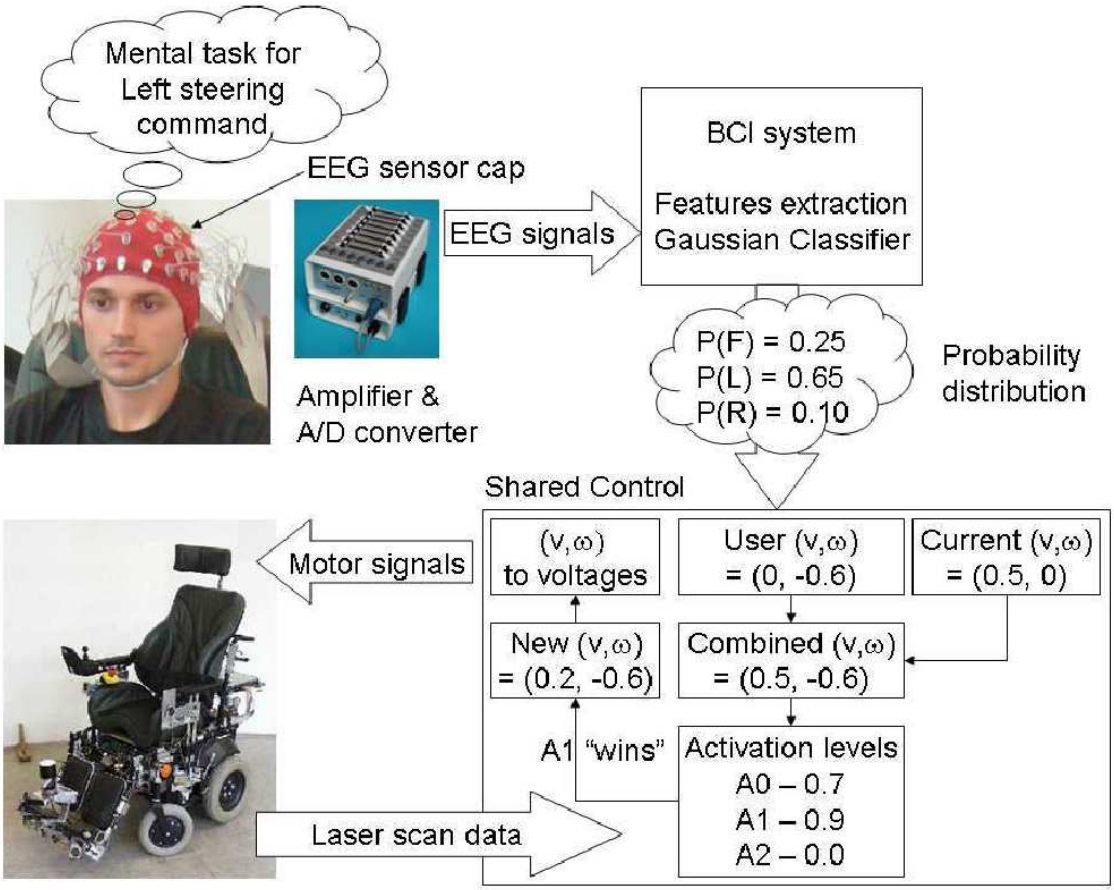


Figure 2.9: With the MAIA's brain controlled wheelchair the user continuously interacts with autonomous behaviors of the wheelchair (from [56]).

behaviors. Three autonomous behaviors are built-in: a collision prevention mechanism that keeps a safety distance with surrounding obstacles, an obstacle avoidance module that computes the best trajectory to avoid obstacles, and a goal tracking behavior that globally keeps the wheelchair in the direction of the goal, thus avoiding getting lost. All assisting behaviors have an appropriateness level computed according to the environmental information. The shared control system then activates the assisting behavior with the highest appropriateness level.

Figure 2.9 illustrates that strategy: the user's mental task corresponds to moving the wheelchair left. In this case, obstacle avoidance is the winning behavior and adjusts the velocity to prevent collision.

This approach proved very successful with non-disabled subjects. However, it requires the user to be constantly alert, which is likely to cause stress and fatigue. The ergonomy constraint is therefore only half fulfilled: the strategy provides an efficient way of navigating toward the goal, however it imposes a heavy burden on the subject. The safety constraint is fully respected: the sensor-based navigator provides a layer to avoid most obvious dangers (obstacles) and the operator provides for all others. However the low cost constraint is clearly broken due to the large amount of sensors used and modifications required to mount them on the wheelchair.

2.3.4 Toyota/Riken

With the brain controlled wheelchair from the Toyota-Riken lab [59] (2008) – shown on Figure 2.10 – the user continuously controls the velocity of the wheelchair by imagining left hand, right hand or both feet movements.

Although impressive demonstration videos were published on the internet, only one article in a peer-reviewed publication could be found, hence no much information can be reported about this project. Besides, it seems that these results were obtained with only one (able) subject, suggesting that their success is based on the exceptional performance of a single gifted subject rather than on a particular technique for EEG signal analysis that could transfer to more subjects. This assumption is reinforced by the mismatch with results in [45] obtained with a similar method (see Figure 2.2.3).

Figure 2.10: With the Toyota/Riken's brain controlled wheelchair, the user continuously controls the velocity of the wheelchair by imagining left hand, right hand or both feet movements (from [59]).

CHAPTER 3

Hardware and Software Description

The control strategy described in this thesis is generic and does not depend on specific hardware. The only requirement is a wheelchair with a localization system and some way of controlling it from a computer. Although more complex implementations are possible, ours was kept simple on purpose: as stated in introduction (Section 1.3), we believe that cost is a decisive factor for the success of a robotic wheelchair.

3.1 Hardware Description

The BCW/CWA prototype is based on a standard powered wheelchair (Yamaha JW-I, see Figure 3.1). Two glide wheels with rotary encoders attached were mounted on the wheelchair's frame below the seat. A mechanical structure with a pivoting arm and a spring was designed to ensure constant contact between the floor and the glide wheels even if the floor is uneven. The glide wheels are made from in-line roller skating wheels, since they offer a good grip and a sharp edge (hence a low sliding ratio during rotation), are made of a robust material, and are available at low cost almost everywhere.

For more flexibility in the development of the first prototype, the control is done from a standard laptop rather from an embedded system with a micro-controller. The laptop is equipped with a PCMCIA digital acquisition card (NI-6024E). It measures

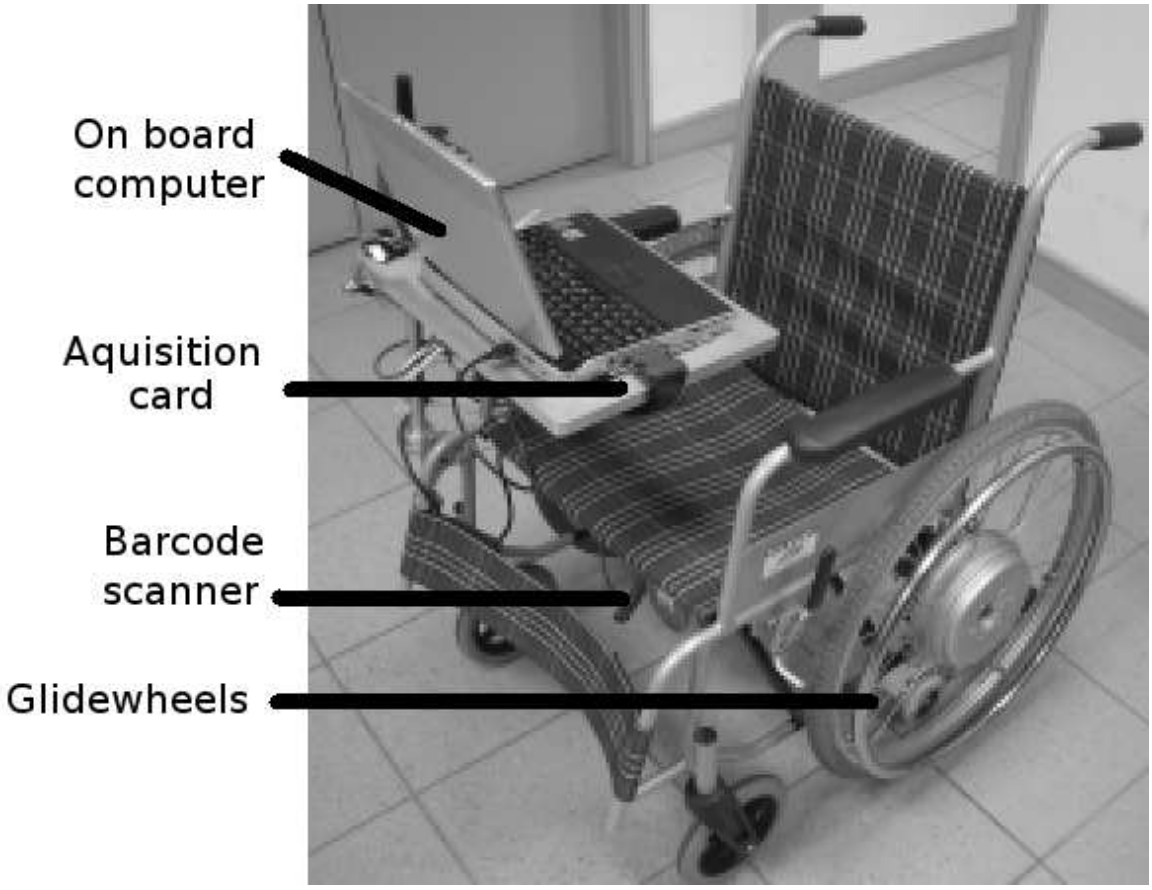


Figure 3.1: BCW prototype with the different components. The EEG system is not represented.

the joystick voltage on both axis and records encoder counts for the two glide wheels. At the output, it simulates the joystick: two voltages (one for each axis) in the same range as the joystick.

Ultrasonic sensors (Devantech SRF-08) are mounted on the front part of the wheelchair to provide a safety mechanism against collision. If the measured distance is smaller than 50cm the wheelchair stops.

3.2 Bar Code Based Global Positioning System

An important issue with our motion guidance strategy is localization. The position of the wheelchair must be known with a precision in the order of a few centimeters in order to be able to pass through narrow ways such as doors. Since odometry alone is subject to an incremental error, a global position information is needed from time to time to reset the odometry error.

There are two ways of obtaining a global position information. The first is to localize the robot with respect to distinctive features of the environment. This is known as the Simultaneous Localization And Mapping (SLAM) [60,61] problem, and solving it has been one of the major objectives of robotic research over the past decade. To work properly, accurate measurement of the environment is required, thus a laser range finder is often used as the main sensor for SLAM. While widely used on modern autonomous robots, it remains a complex and expensive localization system.

The other way of obtaining a global position is to use beacons which position is known accurately. By measuring the position of a couple of beacons with respect to itself the robot can determine its position by triangulation. Beacons can be passive or active; active beacons usually emit radio beams. For indoor navigation, some problems occur due to (i) reflection of radio beams on walls, (ii) occlusion of beacons. Therefore it is difficult to find localization systems that meet our accuracy requirements.

The solution we propose is to place some bar code beacons on the floor and to read them using a simple bar code reader mounted below the chair. Each bar code encodes a unique code that is matched to a position in the computer's memory. When the vehicle passes over a bar code, global position is provided to the system. An extended

Kalman Filter (EKF) is used to combine the position information from odometry and the bar codes. Bar codes are placed at key positions, such as before zones that require precise navigation (i.e. doors or narrow corridors).

The problem with this system is that the position information is only available when the wheelchair passes over a bar code, which means that if the wheelchair misses a bar code then it will have no way of localizing itself. However the wheelchair follows guiding paths, therefore, by placing the bar codes along the paths, we make sure that the wheelchair will pass over it. The only constraint is to maintain the inter bar codes distance small enough so that the odometry drift remains smaller than the size of the bar code.

This system has proven to be simple to set up and sufficiently accurate for our purposes. The maximum positioning error is always less than 10cm when bar codes are placed about every 10m [62]. The position error is less than 2cm just after reading a bar code.

3.3 Software Description

3.3.1 Computing Platforms

For our first prototype, we decided to control the wheelchair from a laptop since it allows easier integration of new features, is cheaper than an embedded platform such as the PC104, and offers a visual display. The low level control is realized by a PD velocity feedback loop. Velocity is obtained by differentiating the reading from the rotary encoders. A hard real time system is required for the velocity measurement and PD control. We used the *RTAI* (Real Time Application Interface) operating system, as it is a reliable, open source, and based on the familiar Linux environment.

In a nutshell, RTAI modifies the Linux kernel to make it preemptible and allow real time tasks to run with strict timing constraints. We use RTAI in conjunction with *Ubuntu* 6.06, which offers good community support due to its popularity. Communication with the DAQ card is done using *Comedi* which is an open source project that develops drivers, tools, and libraries for data acquisition. RTAI offers a real time Comedi module which ensures real time access to DAQ cards supported by Comedi. All code was written in C.

For the BCI we collaborated with the Neuro Signal Processing (NSP) laboratory at I^2R, Singapore. They developed a C++/C# engines for stimulation and signal processing. EEG acquisition is done with the application provided by the manufacturer of the EEG acquisition device. The BCI runs under Microsoft Windows only (XP in our system).

Therefore two laptops are required to operate the BCW: one to control the wheelchair, the other one to run the BCI. The two of them communicate via an Ethernet link.

The connection between the different elements is shown on Figure 3.2.

3.3.2 Control Software Description

In this section we will describe the architecture of the system running on the control laptop. This system has to handle a real time module for localization and velocity control, non real time modules for the path following controller and interaction with the BCI system. Related code can be found in Appendix B. Figure 3.3 shows the interaction between the different components.

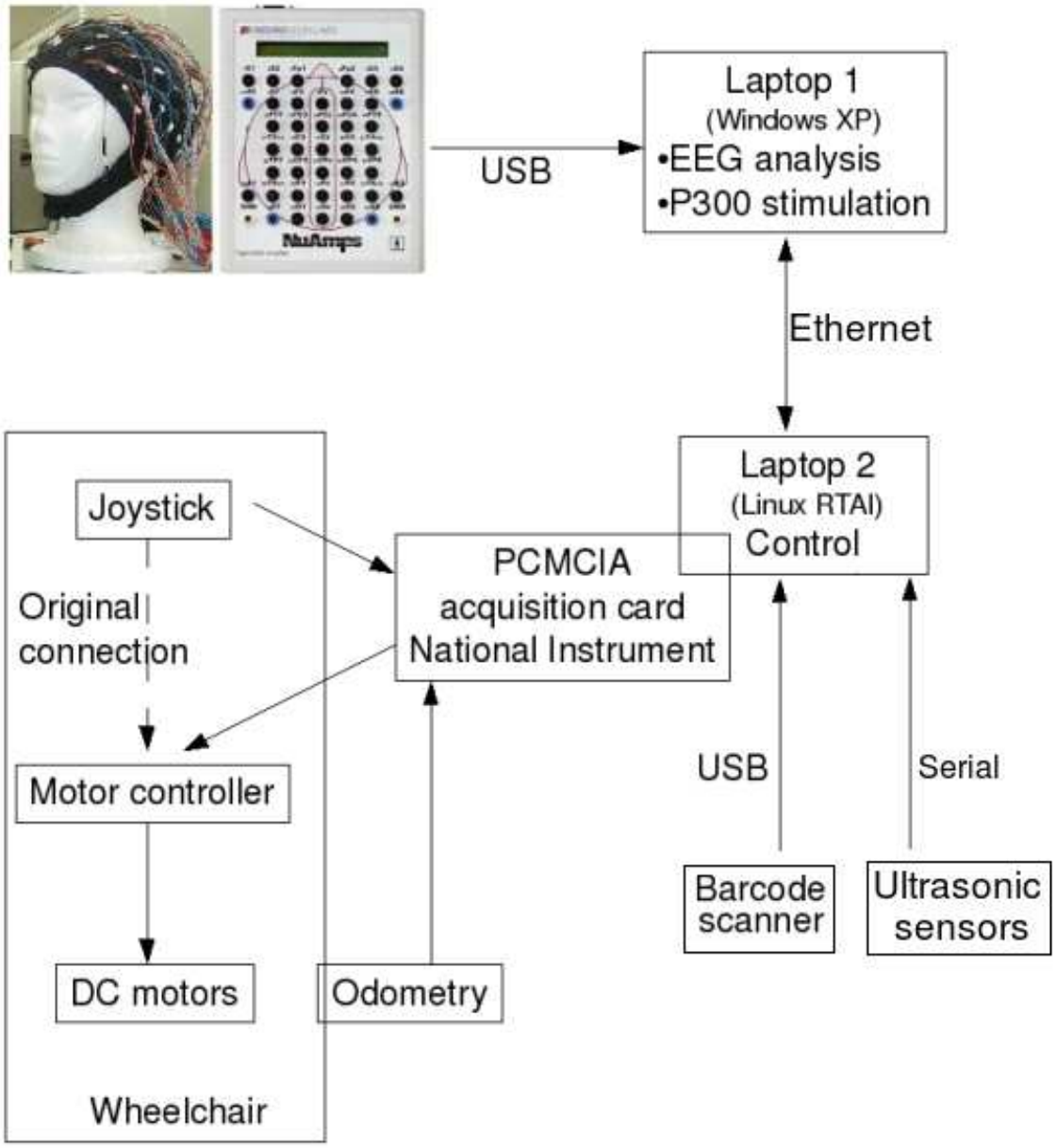


Figure 3.2: Functional diagram of the BCW. Two laptops are used: one to handle the BCI, the other to control the wheelchair. The control command simulates the output of the joystick.

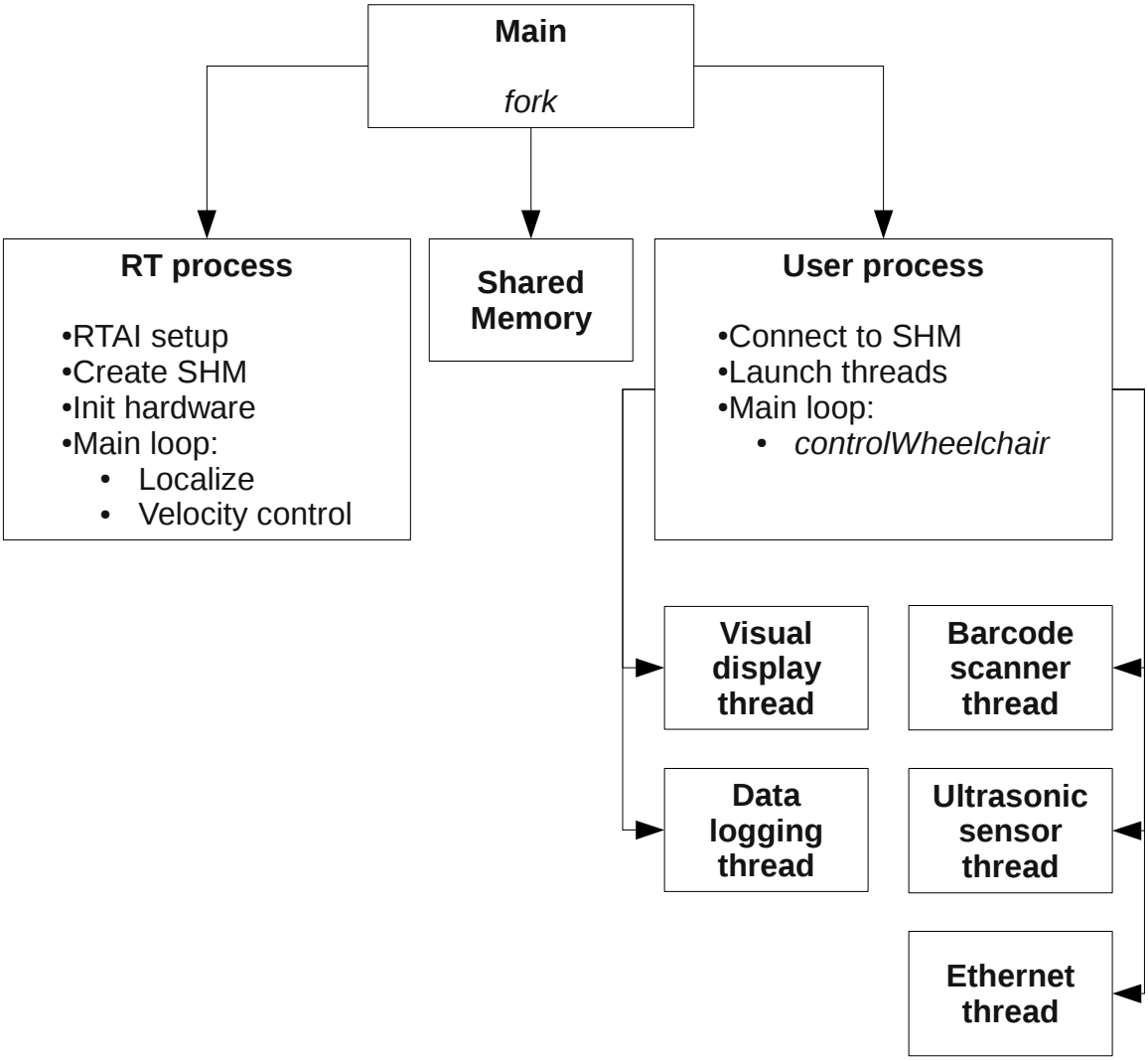


Figure 3.3: Functional diagram of the BCW software system. The main process spawns a real time process and a user process, which in turns launches five threads. A shared memory is used for communication between the two processes.

The main function simply forks off a child process which will be responsible of the real time tasks. The parent process handles all other tasks. Communication between the two processes is realized through a RTAI shared memory. The `hardware_loop` function is called periodically. It does the following:

1. read the counters associated with the glide wheels rotary encoders, from which it computes the current position and velocity.

2. perform PD velocity control.

The computed position and velocity, as well as the desired velocity, are stored in the shared memory. The real time process writes the position and velocity, the user process writes the desired velocity. Thanks to this unilateral writing, the data is naturally protected against corruption and no mutex or other protections mechanism is required. The `hardware_loop` function is called periodically by RTAI, hence with a small jitter. Therefore, the velocity is derived from the counter reading with good precision. This is also important for the D term of the PD controller which required good timing.

The user process main function starts by connecting to the shared memory. Then it launches a couple of utility threads to:

- display a visual feedback (OpenGL) of the wheelchair and the paths,

- data logging (for debugging purpose),

- monitor the ultrasonic proximeter,

- monitor the bar code scanner,

- handle the Ethernet communication with the BCI system.

The user process then enters a periodic loop (100ms) on the `controlWheelchair` function. This function realizes a state machine: at each iteration it will execute the control corresponding to the current state (for instance follow a guiding path), and check for the relevant events (i.e. a message from the BCI, or end of path reached).

Data between the different threads is shared as global variables. Thread synchronization is organised as follow:

- The visual display thread and data logging thread only read the global data. Minor inconsistency can occur if the thread in charge of writing is interrupted halfway, for instance after having updated the x position but before updating the y position. This type of inconsistency is totally harmless has it only results in an error of a few millimeters on the display or in the log file.

- The two sensor monitoring threads only write data. The Ethernet thread continuously monitors the connection for an incoming message and saves it. The data is time-stamped and protected by mutexes. At each iteration of the `controlWheelchair` function, it will check for the time stamp and take appropriate action if new data is available.

Thanks to the simple design and limited interaction between different threads, no dead lock can occur within the system.

CHAPTER 4

Motion Guidance

According to our strategy, we propose to represent the environment by a geometric graph of virtual paths connecting locations of interests. A path following controller is then used to navigate along those guides. The controller we use is based on the work by Claude Samson and Alain Micaelli [63–65], which mathematical development is summarized in Section 4.1. In Section 4.2 we give experimental proof that motion guidance facilitates motion control. Equations of the path following controller were modified to add an elastic deformation capability (Section 4.3), which is useful for path editing (Section 4.4).

4.1 Path Following Controller

We will here briefly summarize the mathematical development of the original path following controller by Samson & Micaelli since some of the control equations are necessary to understand our own elastic path controller. Mathematical details can be found in [63–65] and in appendix A.

4.1.1 Kinematics

We will first look at the kinematic model of a moving point. Consider a moving point $\mathcal{M}$ and the associated Frenet frame ($\mathcal{T}$) defined on the curve ($\mathcal{C}$) as indicated in

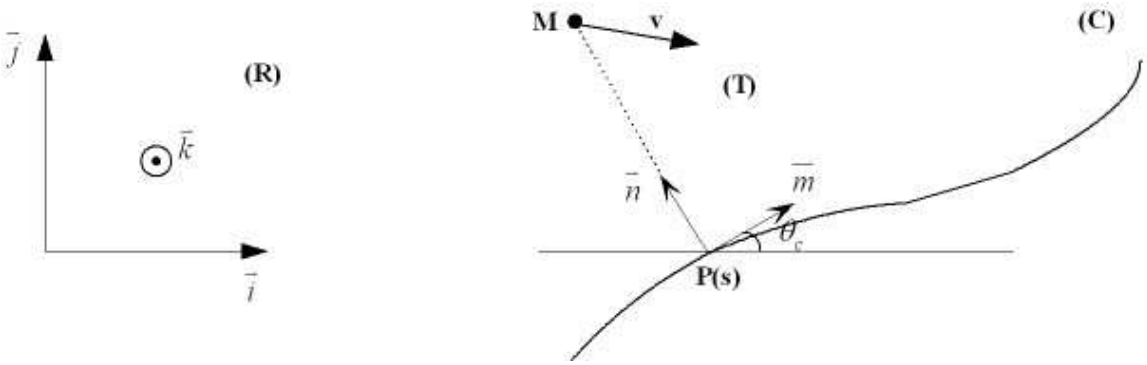


Figure 4.1: Frames and notations for the controller.

Figure 4.1. The point $\mathcal{P}$ is the orthogonal projection of the point $\mathcal{M}$ onto the curve $\mathcal{C}$. Basic mathematics leads to:

$$\begin{cases} \dot{s} = (\cos\theta_c \quad \sin\theta_c) \cdot \binom{\dot{X}}{\dot{Y}} / [1 - c_c(s)y] \\ \dot{y} = (-\sin\theta_c \quad \cos\theta_c) \cdot \binom{\dot{X}}{\dot{Y}} \end{cases} \tag{4.1}$$

where:

s	is the curvilinear abscissae of the origin P of the Frenet frame (T)
y	is the distance between the mobile M and the path
θ_c	is the angle between the reference frame (R) and the Frenet frame (T)
$\dot{X}$, $\dot{Y}$	are the components of the velocity of the point M in the reference frame (R)
c_c	is the curvature of the path: $c_c = d\theta_c/ds$

Our wheelchair platform has two actuated wheels on a common axis and the reference point $\mathcal{M}$ at mid-distance of these two wheels (see Figure 4.2), so the kinematic equations of this unicycle-type vehicle are as follows:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = v \cdot \begin{pmatrix} \cos\theta_m \\ \sin\theta_m \end{pmatrix} \tag{4.2}$$

From the above two functions, we have the following expression of unicycle expressed in coordinates $\{s, y\}$:

$$\begin{cases} \dot{s} = v\cos(\theta_m - \theta_c)/(1 - c_c y) \\ \dot{y} = v\sin(\theta_m - \theta_c) \end{cases} \tag{4.3}$$

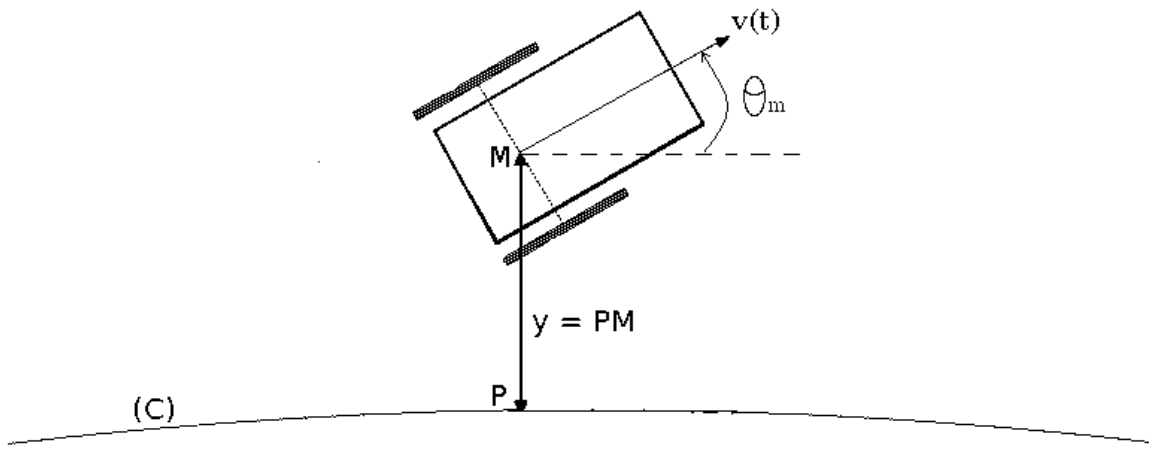


Figure 4.2: Wheelchair's kinematics.

For simplicity, we make $\theta = \theta_m - \theta_c$, so we have the kinematics function of the wheelchair expressed as (using $c_c = d\theta_c/ds = \dot{\theta}_c/\dot{s}$):

$$\begin{cases} \dot{s} = v\cos\theta/(1 - c_c y) \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \dot{\theta_m} - c_c\dot{s} \end{cases} \tag{4.4}$$

4.1.2 Path Controller

The control variable chosen for this system is the angular velocity $w = \dot{\theta_m}$. To derive the control variable w, we modify the kinematics model of unicycle-type vehicle in terms of the distance traveled by the vehicle along the desired path instead of the time index t. After trivial calculation, we get the expression below (Please refer to appendix A for details):

$$\begin{cases} s' = sign(v\frac{\cos\theta}{1-c_c y}) \\ y' = \tan\theta(1 - c_c y)sign(v\frac{\cos\theta}{1-c_c y}) \\ \theta' = \frac{w|1-c_c y|}{|v\cos\theta|} - c_c sign(v\frac{\cos\theta}{1-c_c y}) \end{cases} \tag{4.5}$$

The control objective is to stabilize the output y to zero. To fall upon a stable closed-loop system, we choose the following PD control law:

$$y'' + k_{vy}y' + k_{py}y = 0; \quad k_{py} > 0, k_{vy} > 0 \tag{4.6}$$

y'' is obtained after lengthy but straightforward calculation as:

$$y'' = \frac{w}{v\cos^3\theta}(1 - c_c y)^2 - c_c(1 - c_c y)\frac{1 + \sin^2\theta}{\cos^2\theta} - g_c y \tan\theta \tag{4.7}$$

where $g_c = \frac{dc_c}{ds}$ is the variation of the curvature.

From Equation (4.4), (4.5), (4.6) and (4.7), the resulting control is:

$$\begin{cases} s' = sign(\frac{v\cos\theta}{1-c_c y}) \\ y' = \tan\theta \ \ (1 - c_c y) \ \ sign\left(\frac{v\cos\theta}{1-c_c y}\right) \\ \theta' = w\left|\frac{1-c_c y}{v\cos\theta}\right| - c_c \ sign\left(\frac{v\cos\theta}{1-c_c y}\right) \end{cases} \tag{4.8}$$

Equation (4.8) is the final equation implemented in the control code. Samson and Micaelli in [63–65] demonstrated that this controller is stable provided that a set of simple initial conditions is satisfied.

4.2 Evaluation of the Motion Guidance Controller

The path following controller was implemented and evaluated on two robots, the Scooter COBOT and the CWA.

4.2.1 The Scooter COBOT

The scooter COBOT [66,67] (COllaborative roBOT) is a haptic device developed at the Laboratory of Intelligent Mechanical Systems (LIMS), Northwestern University (Illinois, USA). It is a triangular vehicle with a steerable wheel at each corner (Figure 4.3). The orientation of each wheel is controlled by the computer, however

the rotation speed of the wheels is free: the scooter is set into motion by pushing it, hence it is a passive device. A 6-DOF force sensor is mounted below the handle to detect user's input. Velocity and position are measured by odometry using three glide wheels with encoders.

The motion guidance strategy evolved in the following context: the COBOT was designed to be an assistive device for workers at General Motors whose task is to assemble doors in the car frame. This repetitive task requires good precision while manipulating heavy objects. The objective was to reduce manipulation effort while at the same time keeping a human in the loop for safety and control. Since the trajectory from the doors stock to the car construction line was always the same, it made sense to trace a guiding path once and subsequently repeat it without having to control the trajectory anymore.

4.2.2 The Collaborative Wheelchair Assistant (CWA)

The CWA [62, 68] is closely related to the BCW: they use the same wheelchair prototype and are both based on the motion guidance strategy. However, the CWA is targeted at users who can use continuous input devices, such as a joystick, although with difficulties using it accurately due to poor motor control. With the CWA, the wheelchair's motion is constrained on guiding paths as with the BCW, however the user has full control over the speed.

4.2.3 Motion Guidance Reduces Control Effort

Boy [69, 70] performed experiments to prove that using motion guidance reduces the control effort and therefore facilitates the navigation task. Experiments were carried with the scooter COBOT: young and healthy subjects were asked to push it

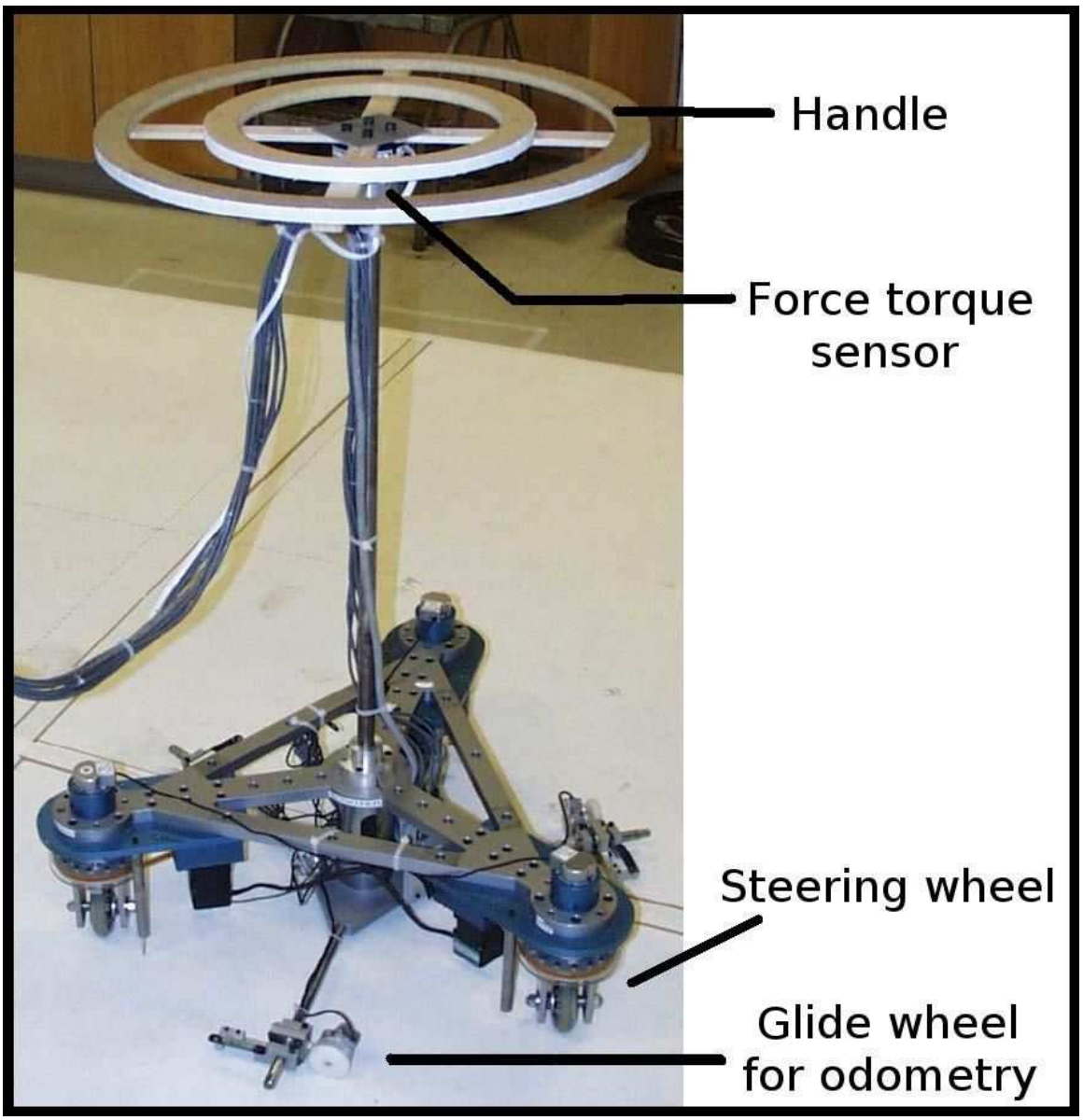


Figure 4.3: The scooter COBOT [66, 67] on which the Elastic Path Controller was tested.

through an environment that forced them to continuously steer it, and at the end insert a pin on the tip of the COBOT inside a small hole. In Free Mode (FM), the path following controller is deactivated and the COBOT behaves as a platform mounted on caster wheels. In Guided Mode (GM) it is constrained along a guiding path. The results show that the speed in GM is typically higher than in FM and that the high-frequency content of the torque is significantly larger in FM than in GM.

Similar experiments were conducted on the CWA [71]. Healthy subjects were asked to navigate the wheelchair on a complex trajectory. The published results show that in GM the joystick control is reduced to selecting forward or backward motion, and that almost no steering is required. Besides, travel time in GM is shorter than in FM.

These experiments prove that motion guidance efficiently simplifies navigation by reducing the control effort to the control of the forward velocity.

4.3 Elastic Path Controller

The idea of the Elastic Path Controller (EPC) is to deform the actual path by applying an input perpendicular to the guiding path. One can think of the actual path as a rubber string. The shape of rubber band will be changed when the user give a force perpendicular to it. When the user releases the force, the rubber band will recover to its original shape. This behavior is implemented by modifying the control law of Equation (4.6) as follows:

$$y'' + (1-\alpha)(\underbrace{k_{py}y + k_{vy}y'}_{restoring\ force}) + \alpha F_{\perp} = 0; \quad k_{py} > 0, k_{vy} > 0 \tag{4.9}$$

where $F_{\perp}$ is the deforming input (i.e. the lateral component of the joystick) and α is the elastic factor ($0 <= \alpha <= 1$).

When $\alpha = 0$, Equation (4.9) is the same as Equation (4.8), i.e. the elastic mode is disabled. When $\alpha = 1$, the *restoring force* term $k_{py}y + k_{vy}y'$ disappears from the control, i.e. the wheelchair behaves as in Free Mode. For intermediate values of α ($0 < \alpha < 1$), the larger the value, the less the user will feel the attraction from the path, i.e. the controller is more elastic.

The resulting control is:

$$\begin{cases} \dot{s} = v\cos\theta/(1 - c_c y) \\ \dot{y} = v\sin\theta \\ \dot{\theta} = v\frac{\cos\theta}{1-c_c y}\Big\{y\frac{\cos\theta}{1-c_c y}(g_c\sin\theta - (1-\alpha)k_{py}\cos\theta) + \sin\theta\Big[c_c\sin\theta - \cdots \\ -(1-\alpha)k_{vy}\cos\theta sign(\frac{v\cos\theta}{1-c_c y})\Big] - \alpha F_{\perp}\frac{\cos^2\theta}{1-c_c y}\Big\} \end{cases} \tag{4.10}$$

Equation (4.10) is the final equation implemented in the elastic controller's code. A more detailed description of this elastic controller can be found in [68, 72–76].

4.4 Designing Paths

The motion guidance strategy relies on a collection of guiding paths – a map – for navigating between locations of interest in the environment. Figure 4.4 shows an example of such a map for a home environment.

Guiding paths can be built up automatically if a reliable plan of the building is available. Alternatively and preferably, guiding paths can be created by Walk Through Programming (WTP): the on-board computer records the trajectory while a helper is pushing the wheelchair between two locations along the desired trajectory. The trajectory is recorded as a sequence of points. Points are placed 5cm apart which corresponds to the localization accuracy of the system as shown in [62].

Figure 4.5 shows an example of a path obtained by WTP. A hook can be seen at the end of the path. Such a hook frequently appears on paths designed by WTP

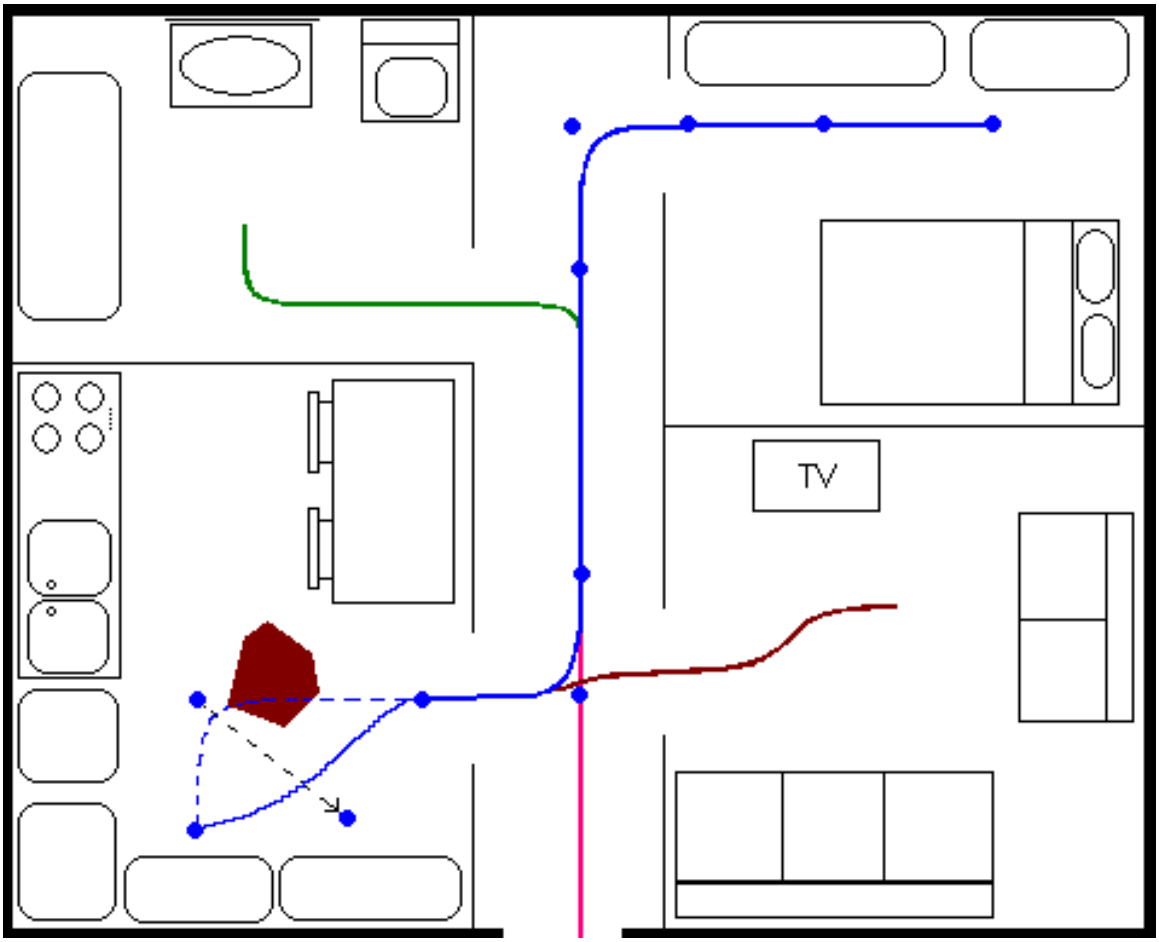


Figure 4.4: Example of a map with guiding paths in a home environment.

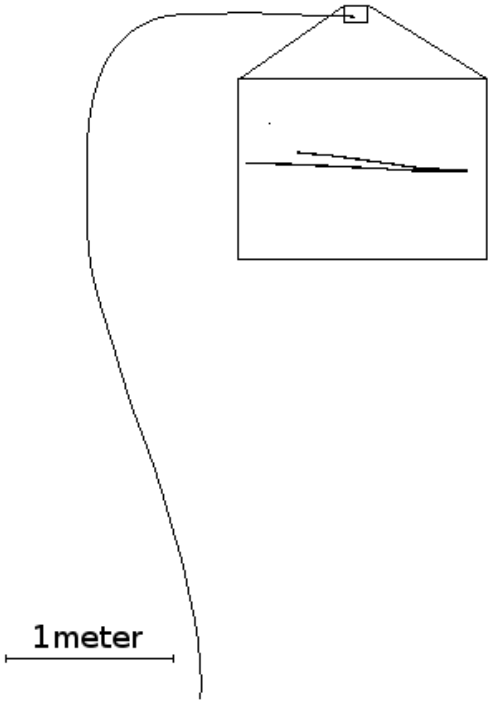


Figure 4.5: A 8m long path obtained from WTP. A hook can be seen at the end of the path.

because the wheelchair usually moves backward a little bit after the user has stopped driving or pushing it. It has to be removed from the path otherwise the path following controller cannot reach the end of the path. This can be done manually by removing the last few points from the path file. Alternatively, a small utility program can be used.

Once a path has been traced, a cubic B-spline is least-squared fit to the recorded trajectory and serves as guiding path for subsequent movements. A B-Spline is a piecewise polynomial function which is locally simple, yet smooth and globally flexible. Each segment is defined by few control points (4 for a cubic B-Spline) that act as attraction points to the curve. Modeling the trajectory by a B-Spline smoothes the cure (low-pass filtering). Moreover, it reduces the number of points required to describe the path from hundreds to a couple, and since control points have an intuitive geometrical meaning as attraction points, it facilitates path editing.

The number of control points used is chosen by the user while splining. The more control points, the closer the spline fits the original path. The fewer control points, the smoother the spline, and the easier to edit. To that extent, the splining programs starts with one control point per 1.5 meter of curve and displays the resulting spline together with the original path. The user can then change the number of control points and immediately see the result. The user can also add, remove or move control points with the mouse. Figure 4.6 shows the same path as in Figure 4.5 splined with 5, 6 and 10 control points respectively.

Three tools are available to adapt the paths to permanent modifications in the environment such as changes in the furniture locations:

1. Using WTP the user can retrace the whole path.

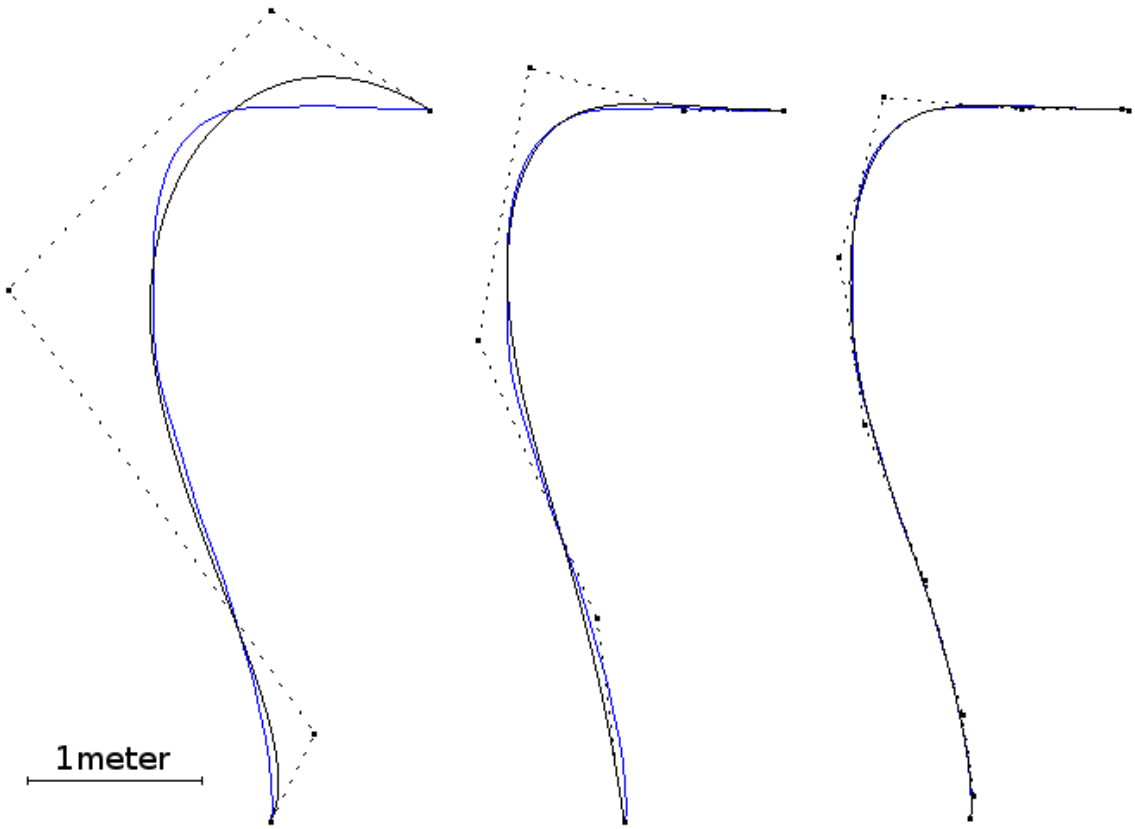


Figure 4.6: Fitting a B-Spline to a path using 5, 6 and 10 control points respectively. The more control points, the closer the spline fits the original path; the fewer control points, the smoother the spline, and the easier to edit.

2. Using the elastic mode the user can deform a small portion of the path.

3. Using a dedicated software the user can modify the paths by adding, removing or dragging control points on the graphical user interface (GUI).

One major advantage of these tools is their great simplicity: they require no environment model and no complex operator procedures.

Boy [70] investigated how humans use the WTP and the GUI tool to design a good guiding path for the scooter COBOT. Subjects were asked to trace a guiding path with (i) WTP, (ii) the GUI, and (iii) WTP together with the GUI for minor improvements. They were then asked to grade each of these paths. It was found that paths designed by WTP then improved with the GUI were the best paths.

In [77, 78] we evaluated on the CWA how humans use the GUI and the EPC to modify an existing guiding path. It was found that the best strategy is to first modify the path with the EPC in order to set the general shape, then improve it with the GUI.

Overall, these experiments gave us an insight at the strengths and weaknesses of each of these three tools:

1. WTP is best used to trace a new path (or retrace an existing path that requires too many modifications). However, the resulting path may require some minor improvements: straight lines may not be perfectly straight, and some curves may be smoothened.

2. The major problem of the GUI is that the surrounding environment is not displayed along with the path. However, after tracing a path by WTP, one can easily and quickly improve it in the GUI.

3. The EPC is best used to modify a small section of a path to adapt it to some modifications in the environment. However, due to the difficulty of using the EPC, the resulting path is usually a bit jerky, and requires some improvements, which are easily and quickly done with the GUI.

When building a network of paths it is possible to reduce the amount of WTP tracing by concatenating chunks of paths using our GUI, which is especially useful for large environments. Also, paths can be organized hierarchically in maps and sub-maps, and maps can be connected together. For instance at a lift several maps, one for each level, are connected together. This hierarchical organization allows to reduce the number of options displayed at one time on the destination selection interface, which facilitates selection.

CHAPTER 5

BCI For Destination Selection

According to our strategy, locations of interest to the user are connected by virtual guiding paths. To navigate with the BCW, the user simply selects with the BCI, among all possible location, which one he or she wants to go to. In Section 2.2 we reviewed a variety of existing BCIs and all of them could be adapted to serve our purpose. For instance, possible locations could be displayed on screen and the μ/β BCI (Section 2.2.3) could be used to control a pointer over the desired item.

The destination selection problem is similar to the spelling problem: it consists in selecting an item of interest in a possibly long list. The best BCIs for spelling are those based on the P300 signal. P300 spellers have proved very successful since introduced in 1988 by Farwell and Donchin [32], and were shown to work well with severely disabled subjects [52, 53]. Besides they require almost no training. Hence, for this work we naturally chose to use a P300 BCI for destination selection.

5.1 Details of the P300 BCI

To elicit P300 signals, we are using a visual oddball paradigm. Items to be selected are displayed in a 3x3 matrix on a screen and flashed one by one in a random order (see Figure 5.1), at a rate of approximately one every $100ms$. To select one item (the

target), the user focuses his or her attention on it; a simple way for focusing is to count the number of times the target is flashed.

Around $300ms$ after the target is presented, a positive potential peak appears in the EEG signal (Figure 2.4). Upon detection of this P300 signal, the target can be determined as the stimulus that occurred 300 ms earlier. One does not need to gaze at the target on the screen, but only to concentrate on it: the P300 is a measure of surprise, and not a direct visual signal.

The P300-based BCI has the advantage of requiring no training from the user and only a few minutes to calibrate the parameters of the detection algorithm. This is noteworthy since some BCI techniques require a very long training phase, up to several months in the case of slow cortical potential devices [79].

For the BCW we are using the asynchronous P300 BCI described in [80]. The signals from 15 electrodes recorded on the top of the head are first amplified and filtered, then cleaned from artifacts (ECG and signal resulting from eye blinks). These are then segmented to associate each button with a sample corresponding to data between $150ms$ and $500ms$ after this button has been flashed (see Figure 5.2). Downsampling and principal component analysis are applied to reduce data dimension.

A Support Vector Machine (SVM) is trained to discriminate samples that contains a P300 signal from the others. A collection of labeled samples for training can be obtained by telling the subject which item to look at. Once trained, the SVM classifies samples as containing / not containing a P300 signal. The SVM output also contains the sample's distance margin (the score) which expresses the likelihood that the sample contains a P300 or not. Finally scores are averaged over the last few epochs (all the buttons flashed). When one or several scores are higher than a

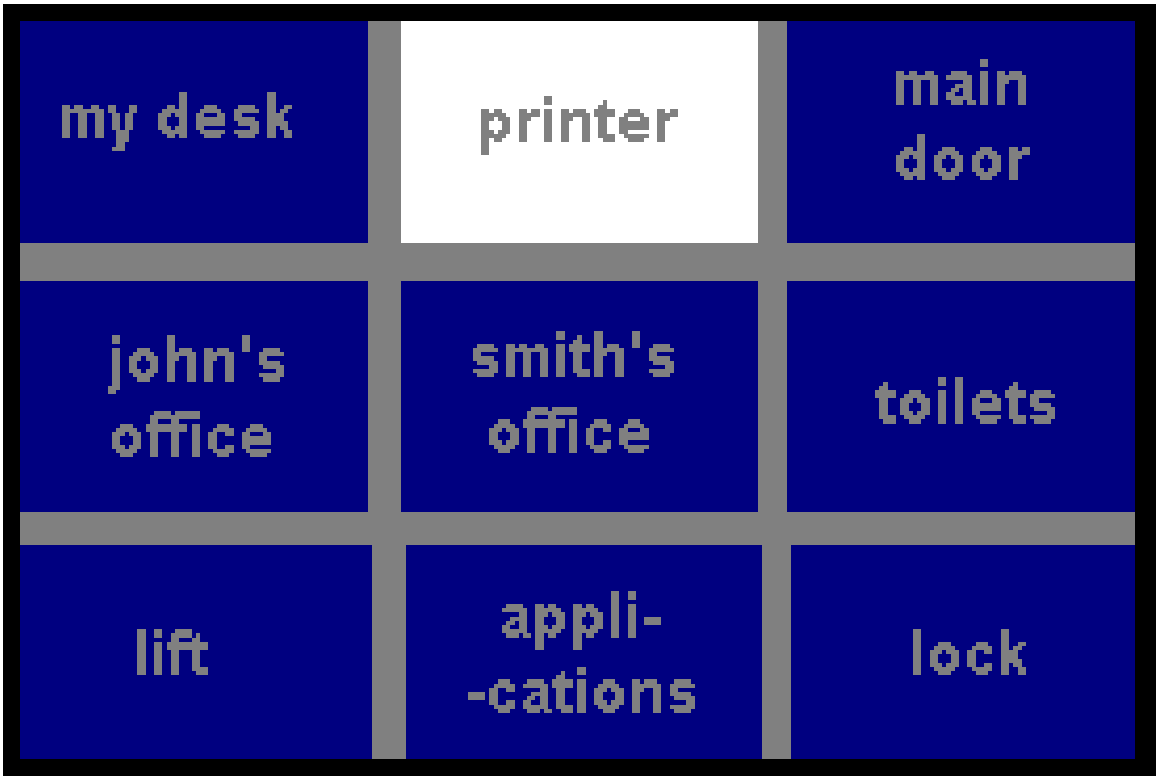


Figure 5.1: Items are flashed one by one in a random order. To select one item the user focuses his or her attention on it. Around $300ms$ after the target is presented, a positive potential peak appears in the EEG signal.

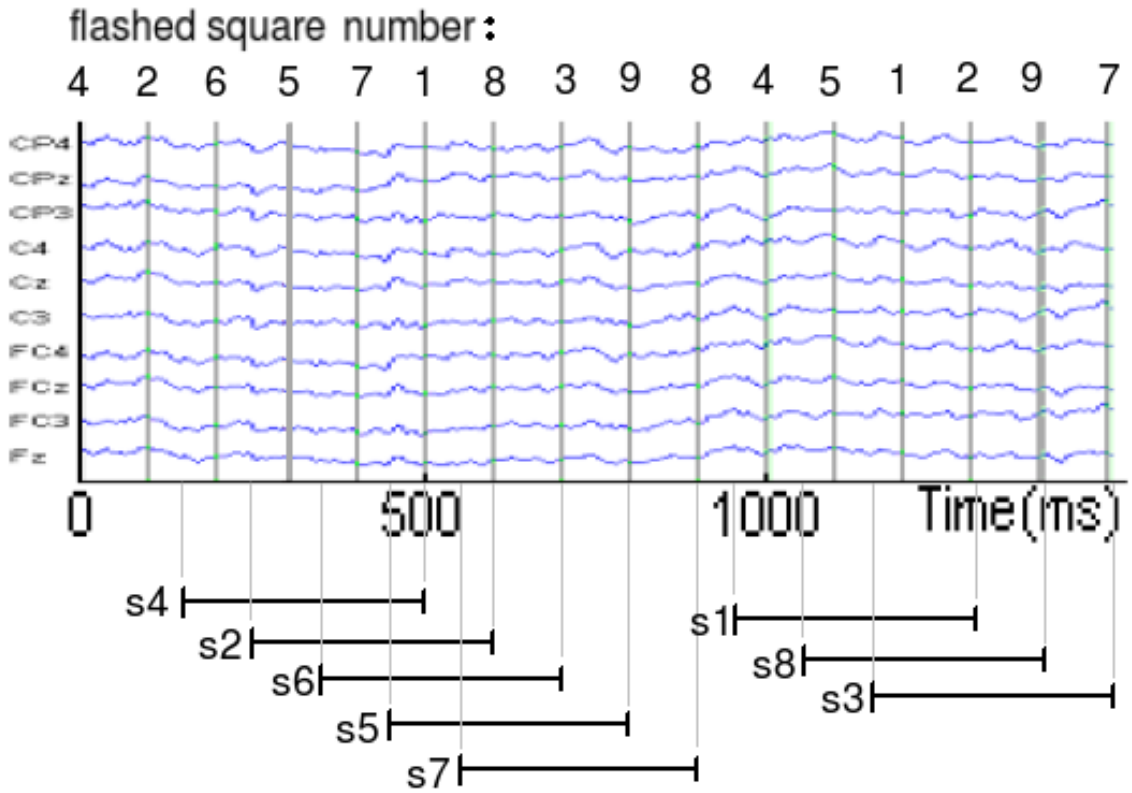


Figure 5.2: Segmentation of the EEG signal in labeled samples. Each button is associated with a sample corresponding to data between 150ms and 500ms after this button has been flashed.

decision threshold, the button with the maximum score is designated as the target. This is summarized on Figure 5.3.

5.2 Experimental Procedure

We devised a few machine-guided tasks so as to conduct analysis of our P300 interface when the subject does not move. Note that the interface used for presenting the stimuli was a 3x3 matrix as illustrated in Fig. 5.1. Buttons were flashed individually in a random order. Stimulation duration was $100ms$ with an interstimulus interval of $10ms$. Thus, one round of button flashing took $990ms$. We recorded data corresponding to the control and noncontrol condition.

The control task was divided in 4 sections. In sections 1 to 3, the subject attended to one button for eight rounds, paused $2s$, and moved on to the next button until he/she had gone through all the nine buttons. Hence, each of the three sections contained 72 rounds of EEG, corresponding to 72 epochs of target P300 and 576 epochs of nontarget P300 data. The three sections were used to train the support vector machine to discriminate between target and nontarget P300 data. The setting of the fourth section was the same as that of the first three, except that the subject had to attend to the targets for fifty rounds. Hence the fourth section contained 450 rounds of EEG, corresponding to 450 epochs of target P300 and 3600 epochs of non-target P300 data. This data was used to evaluate the trained support vector machine.

The noncontrol task was divided in 3 sections, during which the subject stayed in noncontrol state, paying no attention to any button nor the computer display. Each section contained 50 rounds of EEG for each of the nine buttons, corresponding to

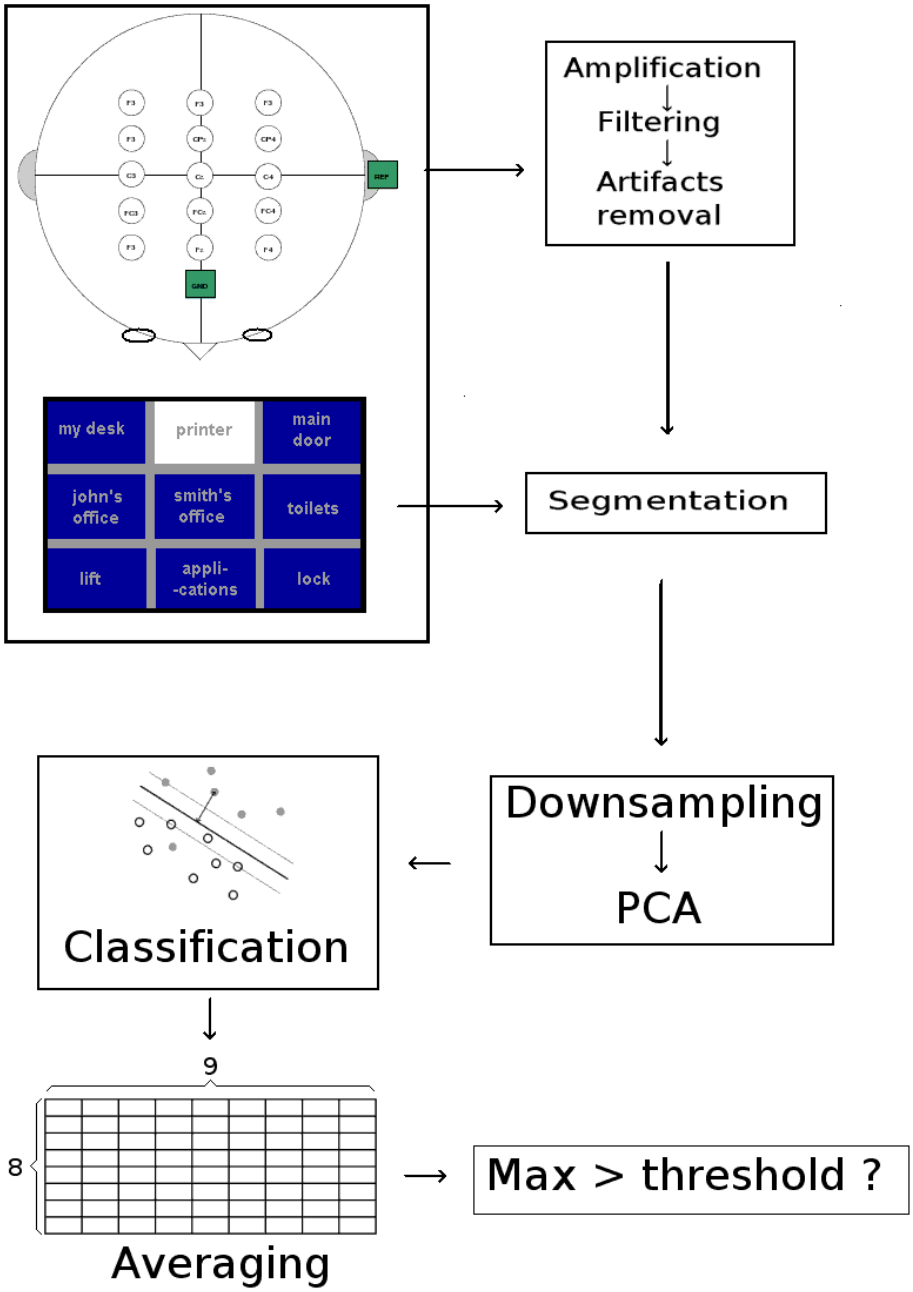


Figure 5.3: Detection of P300 signals in the EEG. EEG samples are fed to a SVM which returns a score that expresses the likelihood that the sample contains a P300 or not. Scores are averaged over the last eight epochs to avoid exceptions affecting the selection. When one or several scores are higher than a decision threshold, the button with the maximum score is designated as the target.

450 epochs of noncontrol EEG data. In the first section, the subject was reading a newspaper. In the second section, the subject was relaxed with closed eyes. In the third section, the subject was given a question sheet including a few arithmetic tasks, and needed to finish the tasks quickly. No significant difference was found in the behaviour with these three conditions, which were thus combined in the subsequent analysis.

Data collection was performed in a single session. Control and noncontrol tasks were interleaved, with a break of two minutes in between two tasks. According to the timing scheme mentioned before, the 8 rounds tasks took approximately $1'$ and the 50 rounds tasks $8'$. So the data collection on each subject ran for approximately 30 minutes, excluding the EEG preparation time.

Five healthy subjects (Subjects 1 to 5), all males, between 22 to 36 years, participated in our study. Note that no subject screening was conducted, and we used all the five volunteer subjects throughout the study.

5.3 Score Distributions

Figure 5.4 shows the distribution of scores in the three data sets mentioned above for one of the five subjects. We can see that scores are approximately normally distributed. Besides, scores in the *noncontrol* and *nontarget* sets are very similar, which is not surprising given that these EEG samples are all supposed to not contain a P300 signal.

We can also see on Figure 5.4 that the average value of the scores in the *target* set is higher than the average value of the scores in the *noncontrol* and *nontarget* sets. However, the two distributions are not clearly separated: for a given score value, we

cannot decide with a high confidence whether the EEG sample contains a P300 signal or not.

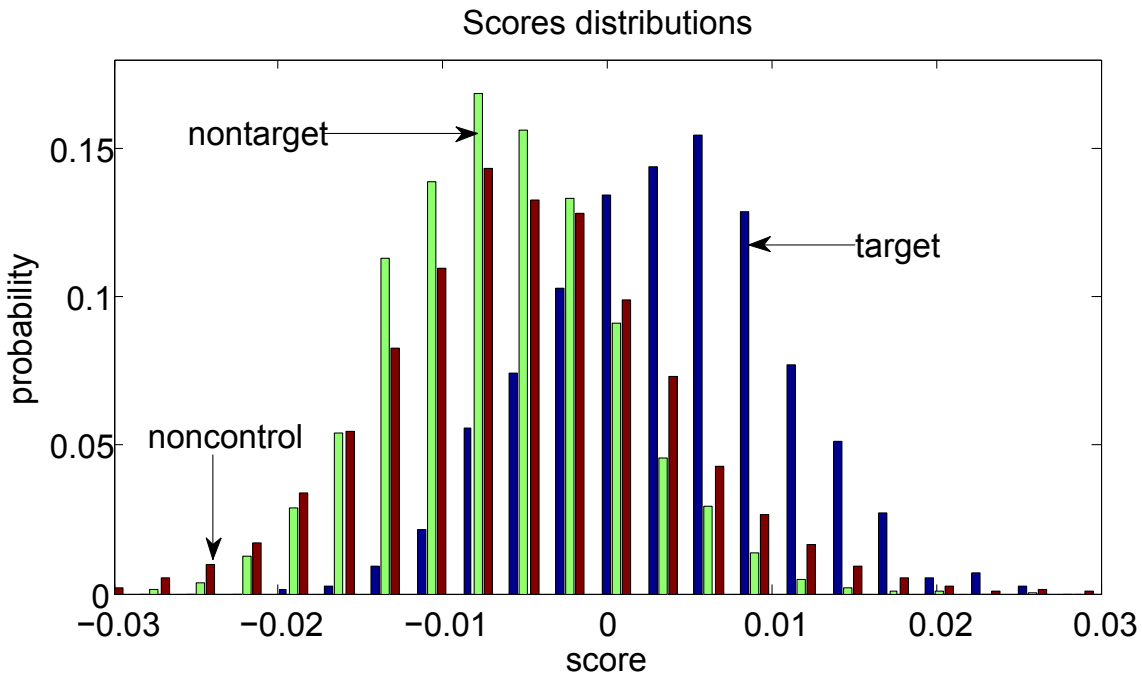


Figure 5.4: Distribution of scores for the *target*, *nontarget* and *noncontrol* sets for subject 1.

5.4 Performance Study

In this section we will evaluate the P300 BCI using the EEG data collected on our five subjects (see Section 5.3). Analysis will be done off-line with Matlab. Our performance evaluators will be the following three:

- Response time (RT). This tells us how long it takes on average to select an item. With our detection algorithm, this is the time before the (averaged) score of

a button reaches the decision threshold. Note that this measures also includes wrongly selected items (substitutions).

- Error rate (Err). This tells us how confident we can be that a selected item corresponds to the item the user intended to choose. It is defined as the ratio of wrongly selected targets by the total number of selections during an experiment, expressed in percentage.

- False acceptance rate (FA). This tells us how frequently items are selected when the subject is not intending to select any. This is an important factor since in a typical usage it is likely that the user will spend most of the time actually not using the wheelchair, i.e. doing something else. It is expressed in number of occurrences per hundred seconds.

For the remaining of this thesis, whenever we will use the word *performance* we will refer to a point in the three-dimensional space (RT x FA x Err).

To solve this multi-objective problem, we combine our three performance evaluators in one cost function:

$$C = \frac{RT}{[RT]} + \frac{FA}{[FA]} + \frac{Err}{[Err]} \tag{5.1}$$

where $[RT]$, $[FA]$ and $[Err]$ are normalizing factors.

We restrict the region of interest to $RT < 30$, $FA < 10$ and $Err < 10\%$: for values beyond those thresholds we set $C = \infty$, corresponding to an unacceptable performance. From preliminary experience with the interface, we know that RT will be in the order of 10 seconds, FA in the order of 1 occurrence per 100 seconds, and Err in the order of 0.1%. Hence, in order to balance order of magnitude of each term, we choose the following normalizing factors: $[RT] = 10s$, $[FA] = 1$ and $[Err] = 0.1\%$.

For each subject we first compute a threshold vector of 100 points that spans the range of scores in the three data sets. Then for each threshold point, and for values of depth from 1 to 16, we compute with Matlab the response time, false acceptance rate, error rate and cost. We then search the depth and threshold values that yield the smallest cost. Table 5.1 shows the obtained performances at that point for each subject. Figure 5.5 shows the distribution of response times obtained across all subjects at the optimal point (average response time is 15.4 seconds).

Table 5.1: Performances at lowest cost point.

Subject	depth	threshold	RT (s)	FA	Err (%)
1	8	0.003665	12.3	1.1	0.15
2	7	0.001055	17.3	1.6	0.11
3	8	0.004170	13.9	1.1	0.09
4	8	0.000402	18.4	1.2	0.14
5	9	0.000391	21.7	1.8	0.13
avg	-	-	15.4	1.2	0.12

Figure 5.6 shows the response time (RT) in seconds, the false alarm rate (FA) in number of occurrences per hundred seconds, and the error rate (Err) in percent, as a function of the threshold for subject 1 and for a value of depth $D = 8$ (which is the optimal value for this subject, as shown in Table 5.1). RT is bound by a minimum of eight corresponding to the averaging window's width. RT increases with the threshold: there are less samples with a high score therefore it takes a longer time until one of the scores reaches the threshold. Conversely, FA is high for low values of the threshold and tends to zero for high values. Err is below 10% and decreases for large values of the threshold.

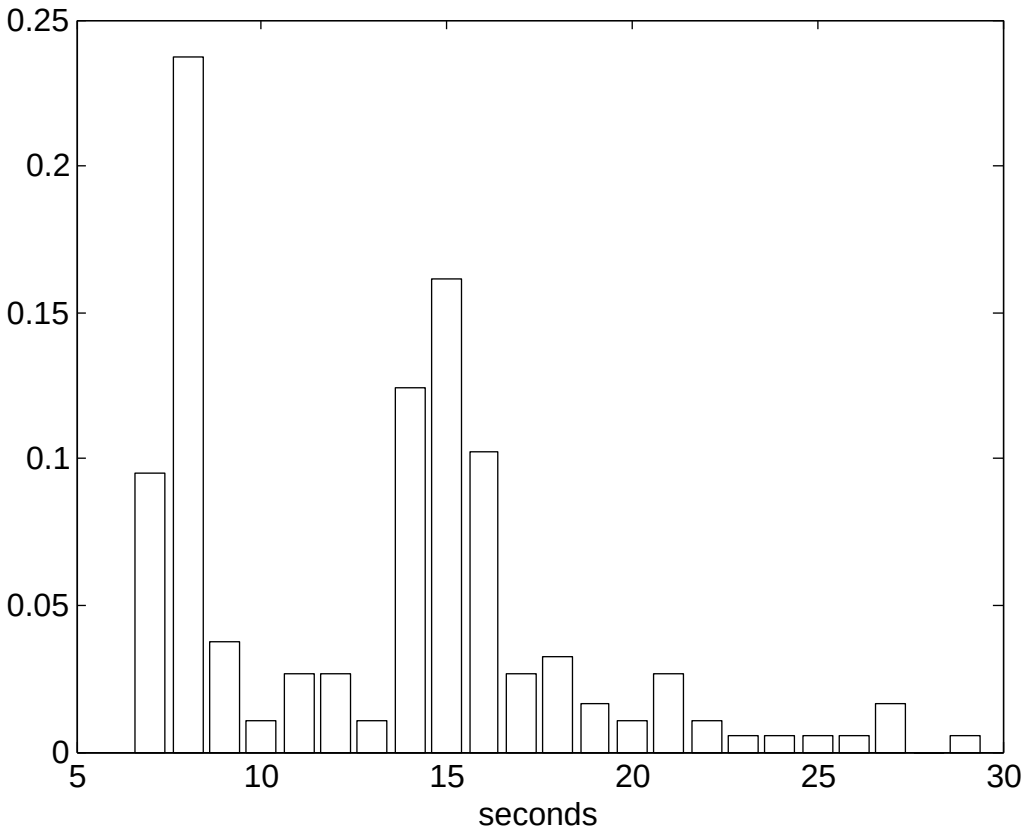


Figure 5.5: RT distribution for the P300 BCI. Times obtained across all subjects at the optimal point.

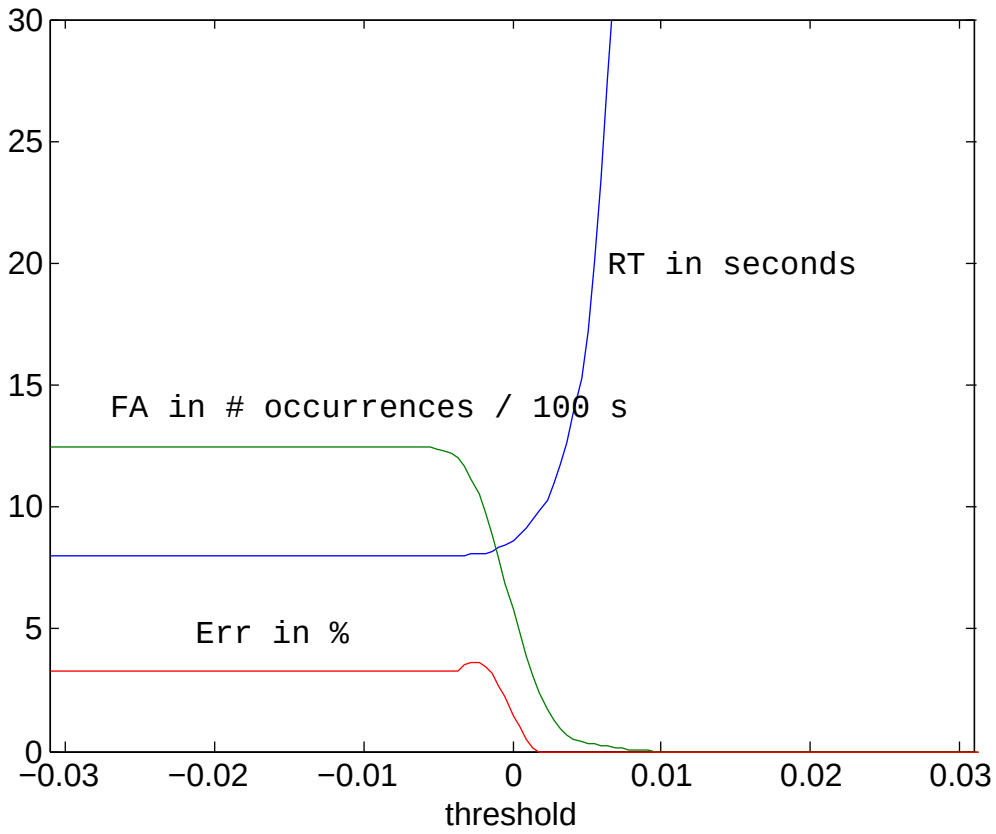


Figure 5.6: Influence of the threshold on performances of the P300 interface. RT (in seconds), Err (in percent) and FA (in number of occurrences per hundred seconds) for different values of the decision threshold (x axis). Results obtained for subject 1 with a depth value of 8.

Figure 5.7 shows the response time (RT) in seconds, the false alarm rate (FA) in number of occurrences per hundred seconds, and the error rate (Err) in percent, as a function of the threshold for subject 1 and for a value of the threshold $T = 0.003665$ (which is the optimal value for subject 1, as shown in Table 5.1). Again, RT increases as depth increases, while FA and Err decrease. For low depth values ($D \leq 4$), the error rate and false acceptance rate are very high.

5.5 Calibration

Automatic selection of the depth and threshold values can be performed though calibration. Data is first collected by asking the subject to select a few buttons on the interface as well as to relax. Then the classifier engine is trained using half of this data. The other half is used to compute RT, FA, Err and the cost, from which the optimal depth and threshold can be found.

These values vary from subject to subject and even from day to day. Therefore, re-calibrating may be necessary when performances decrease.

The depth and threshold values obtained from calibration are optimal in the sense that they minimize both the response time and the false alarm rate. However, optimality of obtained performances also depends on the application and on the user's preferences.

For instance, for a P300 based speller, we would give priority to speed and choose values that yield a small response time, at the cost of the higher false alarm rate. For the wheelchair's destination selection interface, we consider that it is more important to reduce the false alarm rate rather than the response time. Indeed, it is likely that the user will spend most of the time not using the interface than actually using it.

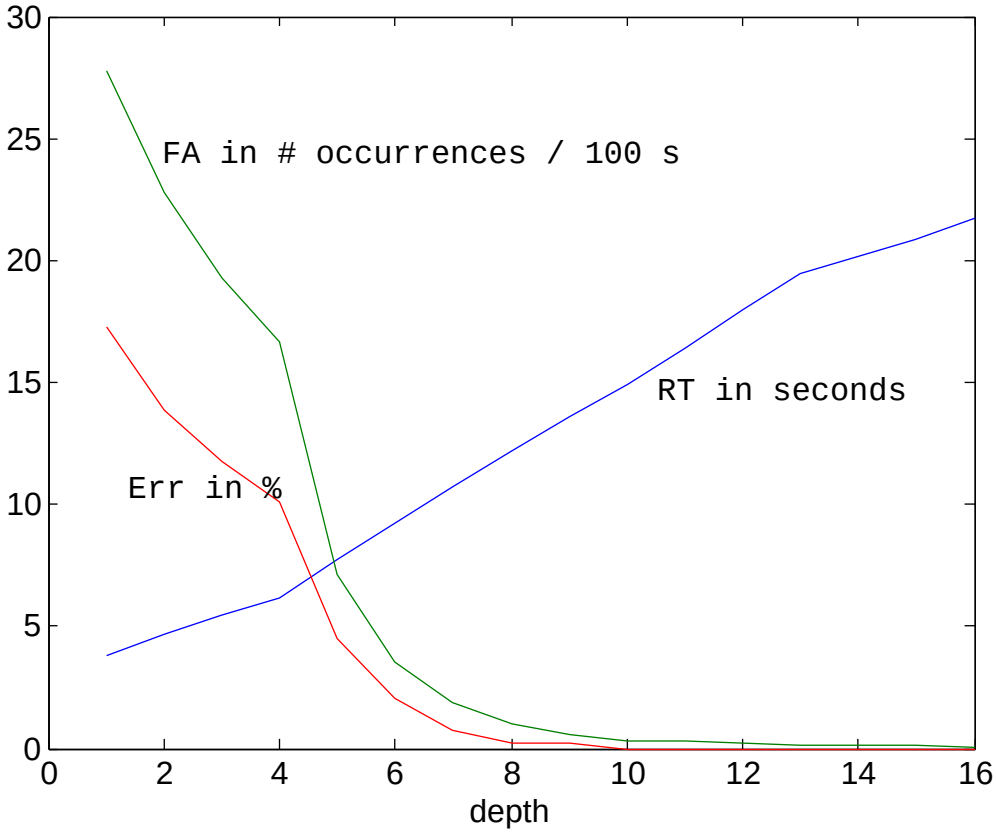


Figure 5.7: Influence of the depth on performances of the P300 interface for the 5 subjects. RT (in seconds), Err (in percent) and FA (in number of occurrences per hundred seconds) for different values of the depth (x axis). Results obtained for subject 1 with optimal threshold value.

Similarly, a particular user might prefer a low response time, while another would favor a low false alarm rate.

Hence, it is desirable that the user (or an aid) has the possibility to tune the performances of the interface. This can be done either off-line by observing RT and FA curves for different threshold and depth values, or on-line by adjusting the depth and threshold values until desired performances are obtained.

5.6 Evaluation of the P300 BCW

We conducted various navigation experiments in our lab building with healthy subjects. This environment included several floors connected by a lift. We assumed a smart environment where the lift could communicate with the wheelchair. Here the environment was simulated by manually operating the lift as well as the entrance and exit of the wheelchair in the lift.

At each floor, four destinations were interrelated by six guiding paths. These guide-paths were designed prior to the experiment using the walk through programming method, i.e. by tracing the paths with the wheelchair and coding the resulting data using B-splines.

Five subjects were asked to move between pairs of locations placed on different floors. All subjects succeeded at their first trial to reach all of the desired locations, taking in mean fifteen seconds (median = 12 s) to issue a command and no wrong command was selected. The subjects reported that it was "very easy" to activate the commands for selecting destinations and change the floor.

5.7 Locking the Interface

Due to the non-null value of FA, the interface is expected to generate random commands while the user is not using it. However, in a daily usage, the user would normally spend large amount of time not using the interface, for instance when resting, or performing another activity. To prevent a falsely detected P300 signal from accidentally setting the wheelchair in motion, we implemented an interface locking scheme similar to the keyboard locking facility on cellular phones.

The interface is locked using the menu's lock button. Once locked, no command can be issued before a sequence of keys is entered. Assuming N keys in the unlocking sequence, the number of false unlocking rate is $FA \cdot (\frac{1}{9})^N$. Using a key sequence of three characters ($N = 3$) and assuming $FA = 1.2$ per hundred seconds (averaged value reported in Table 5.1), the false unlocking rate is 0.00164 per hundred seconds, or one every 17 hours.

CHAPTER 6

Faster BCI for Stopping

Our experiments proved that selecting destinations with the P300-BCI is an efficient strategy to control a wheelchair: it is safe and predictable, and does not require the user to be constantly focused on the driving task. However, in our design constraints (Section 1.3) we established that the control burden must be as light as possible, yet allow certain freedom to the user, such as stopping at any time during motion or change course.

The off-line evaluation of the P300 interface (Section 5.4) showed that by changing the value of the decision threshold we modify the performances of the interface. Low thresholds lead to short response times (RT) and high false acceptance (FA) rate, while higher thresholds lengthen RT and reduce FA. This trade-off of FA and RT is due to the noisy nature of the EEG.

As a consequence it is not possible to use this interface to quickly issue a stop command, as a low RT would be accompanied by a high FA, and the wheelchair would stop too frequently along the way. Hence a faster BCI is needed for stopping, and it can be designed as this decision requires less information than a selection amongst many possible targets. In the course of this thesis we have evaluated two possible fast BCIs that we will present in this chapter.

In Section 6.1 we will present a fast P300 algorithm that works when there is only one item to choose from, and in Section 6.2 we will present a BCI based on motor imagery.

Note that other BCIs may be used for this task, and the two presented here do not form an exhaustive list of solutions.

6.1 Stopping with a P300-BCI

While in movement, the most relevant action is to stop, hence the P300 interface displays only a stop button. In this configuration if an EEG sample is falsely classified as containing a P300 signal, the result of a random phenomena because unrelated to the user's intention, there is only a probability of 1/9 for it to be the sample associated with the stop button. Therefore, the resulting FA should be 9 times smaller, which should allow us to reduce the decision threshold, hence RT, while maintaining FA low.

We will here present and compare six algorithms that were evaluated off-line. Each algorithm is based on the same pattern: it relies on a function that receives scores associated with the stop button and scores associated with the eight other buttons and decides whether the stop button was selected or not. A history of scores over the last D round is kept. The difference between the six algorithms lies in how the scores are processed and how the decision is taken.

The data used are the three data sets introduced in Section 5.2: the *target*, *nontarget* and *noncontrol* sets. The response time is evaluated by using the *target* and *nontarget* sets: the stop button scores are the scores in the *target* set, and is compared to the eight other buttons scores, the scores of the *nontarget* set. The false acceptance

rate is evaluated using the garbage set. Since for this data set the user is not trying to select any button, all buttons are considered as no target. In that condition, we process the data set nine times, using each button is turn as the stop button. This eliminates the bias that could be induced by choosing one button at random.

Similarly to the study conducted for the destination selection interface (Section 5.4), we will consider a cost function to compare those algorithms. In this case, we expect RT to be in the order of 5 seconds, and FA to be in the order of 2 occurrences per 100 seconds, hence we will choose the following normalization factors: $[RT] = 5s$ and $[FA] = 2$ (see Equation 5.1). Note that there is no possible error ($Err = 0$) since only the stop button can trigger a response.

6.1.1 Threshold-Based Algorithm

The four variants of the algorithm presented in this section follow the same principle: for each button, scores are averaged over the last D rounds. The stop button is selected if its averaged score is higher than the threshold and higher than the eight other buttons' averaged scores.

In the first variant (algorithm 1a), the scores are directly stored in the history array. The main problem with this algorithm is that it is very sensitive to threshold variations: tiny variations lead to large differences in performances. A proposed method to reduce the sensitivity is to normalize scores before comparing to threshold. We propose here three normalization methods:

- algorithm 1b: after each round the 9 scores are normalized between -1 and 1 with respect to the round's minimum and maximum values before going to the history array.

- algorithm 1c: scores are normalized between -1 and 1 with respects to the history's minimum and maximum values before comparing to threshold.

- algorithm 1d: scores are normalized with respects to the history's mean and standard deviation values before comparing to threshold.

Figure 6.1 shows the resulting performances for those 4 algorithms for one of the subjects. For each algorithm, using depth values between 1 and 8, we plotted the (FA,RT) points obtained for a range of threshold values.

We computed the cost for all subjects, all algorithms, all depth values and all threshold values. Table 6.1 gives for each subject which algorithm, which depth and which threshold value yields the smallest cost. We can see that for all five subjects, Algorithm 1d is the algorithm that yields the smallest cost.

Table 6.1: Cost analysis for algorithms 1a-d.

Subject	Algo	Depth	Threshold	RT	FA	cost
1	1d	2	1.9	4.0	1.4	1.50
2	1d	2	1.5	5.3	0.8	1.46
3	1d	3	0.5	6.1	1.6	2.02
4	1d	2	1.3	5.2	0.9	1.49
5	1d	3	1.8	4.3	1.2	1.46

6.1.2 Threshold-Less Algorithm

The previous algorithms all have the inconvenient of requiring a threshold, hence a calibration stage is required. We devised two algorithms to get rid of the threshold.

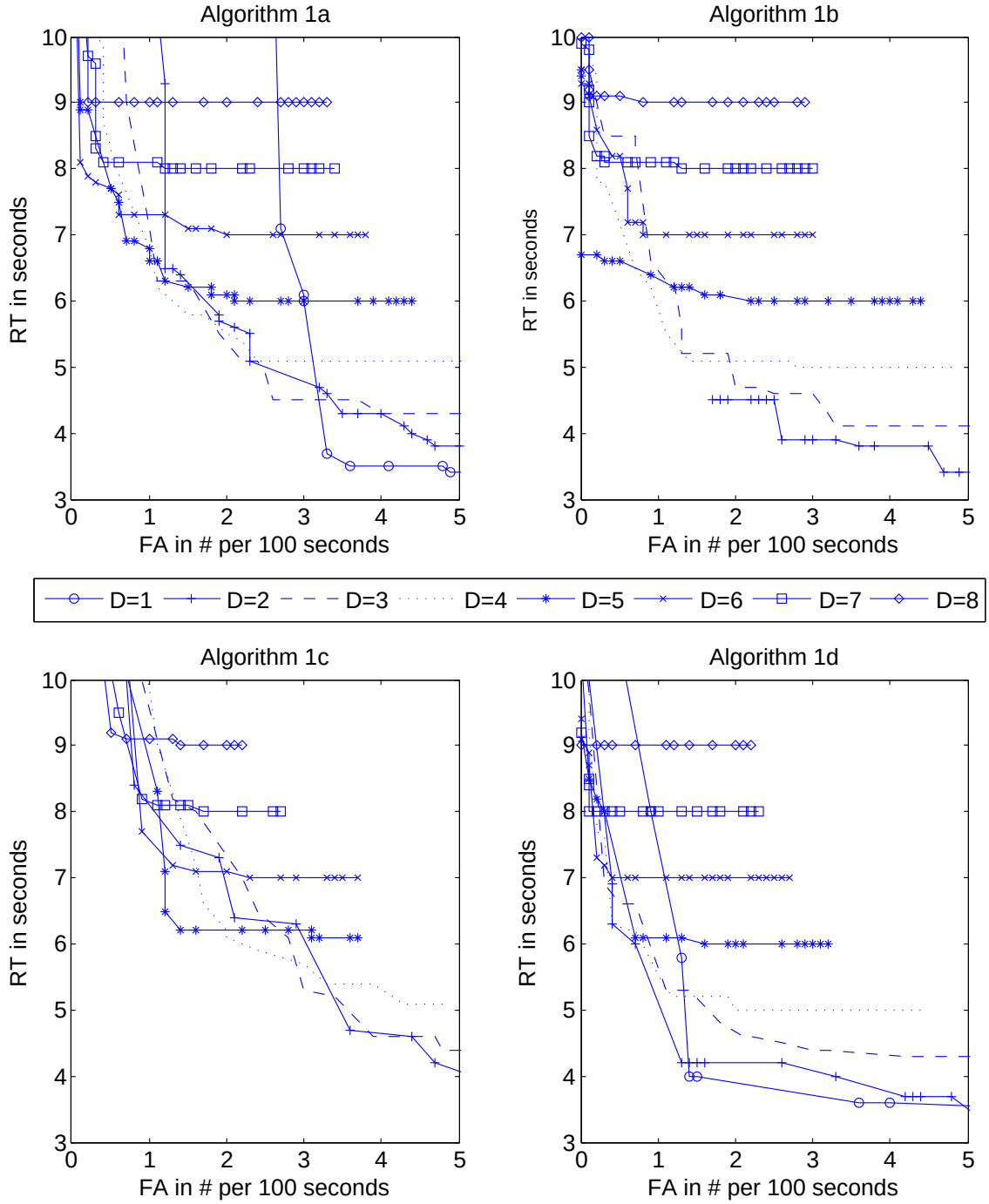


Figure 6.1: Evaluation of the threshold-based P300 stop algorithms. Results corresponding to subject 1. Different variants are proposed: a) no normalization, b), c) and d) scores are normalized following 3 different methods.

In the first variant (algorithm 2a), the stop button is selected if its score is higher than the eight other scores for N consecutive rounds. Figure 6.2-a shows the results for different values of N for subject 1.

In the second variant (algorithm 2b), the stop button is selected if its score is higher than the eight other scores for at least N of the last D rounds. Figure 6.2-b shows the results for different values of N and D for subject 1.

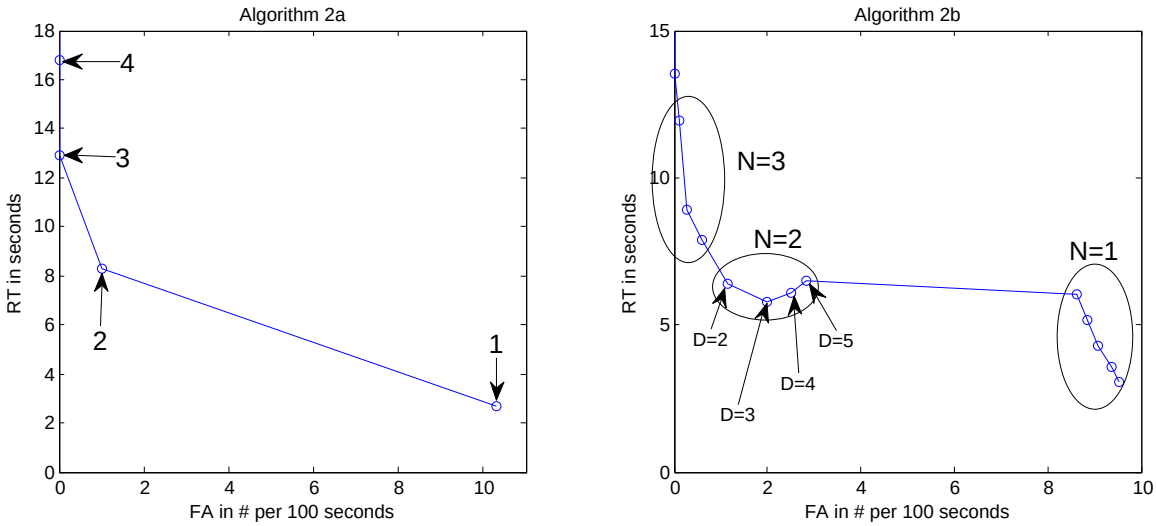


Figure 6.2: Evaluation of threshold-less P300 stop algorithm. Results corresponding to subject 1. Two variants are proposed: a) N consecutive hits, b) N hits over the last D rounds.

We computed the cost for all subjects, both algorithms and all depth values. Table 6.2 gives for each subject which algorithm and which depth yields the smallest cost, and compares with costs obtained for algorithm 1 ($cost1$). The last column shows the gain in cost ($gain = \frac{cost - cost1}{cost1}$).

Table 6.2: Cost analysis for algorithms 2a and 2b.

Subject	Algo	Depth	RT	FA	cost	$cost1$	gain
1	2b	2/3	5.7	2.0	2.14	1.50	42%
2	2b	2/3	6.3	1.2	1.86	1.46	27%
3	2b	2/4	6.9	2.1	2.43	2.02	20%
4	2b	2/3	6.1	1.3	1.87	1.49	25%
5	2b	3/4	4.7	2.1	1.99	1.46	36%

6.1.3 Conclusion

Of all threshold-based algorithms, Algorithm 1d gives the best performances. Algorithms 2a&b get rid of the threshold, hence of the need for calibration. Algorithm 2b yields a smaller cost than Algorithm 2a, yet costs remain higher than with Algorithm 1d.

For the following of this thesis, Algorithm 1d will be used. Corresponding RT and FA distributions are shown on Figure 6.3 (obtained using for each subject the best depth and threshold value).

As with the P300 BCI for destination selection, automatic calibration can be performed to select the best threshold value. Once EEG data has been collected (as in Section 5.5) and the P300 detection engine trained, remaining EEG data can be processed to compute RT, FA and the cost function, from which the optimal depth and threshold can be found.

6.2 Stopping with a μ/β-BCI

In Section 2.2.3, we saw that people can learn to regulate the EEG power in the μ (8-12Hz) and β (18-26Hz) bands, and that this can be used to continuously control

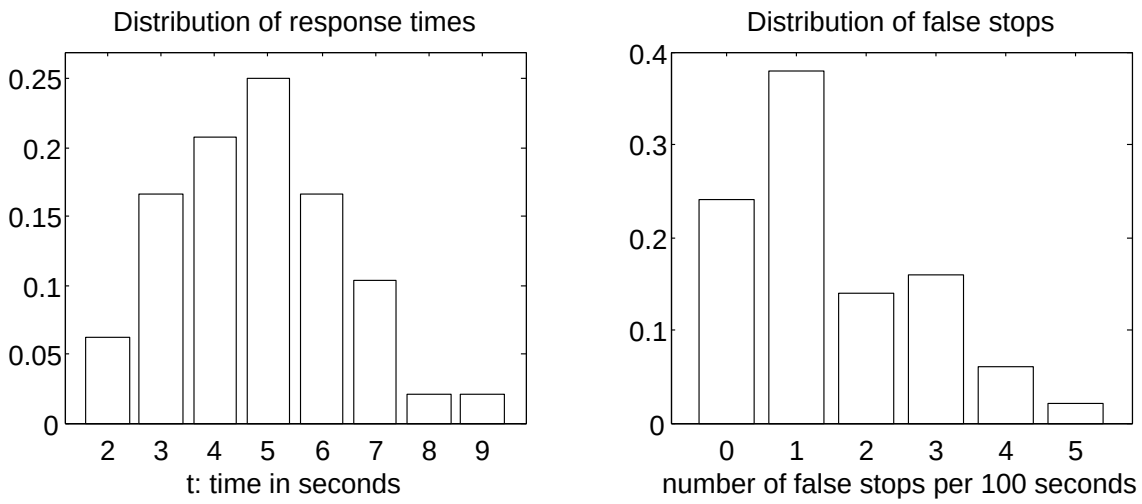


Figure 6.3: Distribution of RT and FA for P300 stop BCI. Best algorithm (Algorithm 1d), all subjects, best depth and best threshold.

a cursor in one or two dimensions. In this section we will describe a BCI based on μ/β analysis to stop the wheelchair.

EEG is recorded over the sensorimotor cortices (electrodes FC3, FC4, C3, C4, CP3, CP4, which are a subset of the electrodes used for the P300 BCI). Before a user starts using the system, the baseline μ and β powers for each channel must be computed, which will be used as reference. To obtain the baseline, the user is asked to stay in "idle" state, in which the user does not move or try to regulate his EEG power, during three minutes. The collected baseline EEG signal is then divided into segments of 250 millisecond, and the μ and β powers in each segment are calculated by using the Fast-Fourier-Transform (FFT). The system computes the average μ and β powers.

In the control stage, the computer calculates the EEG powers in μ and β bands in each 250 millisecond EEG and form a joint feature vector consisting of 12 variables (6

channels and 2 frequency bands): $\vec{x} = \{x_1, \ldots, x_{12}\}$. The control output y is given by $y = \vec{w} \cdot (\vec{x} - \vec{x_0})$, where $\vec{x_0}$ denotes the vector of baseline powers. The vector $\vec{w}$ can be learned using empirical data [81]. Visual feedback in the form of a cursor which position is proportional to the value of the control output y can be presented to the user.

We evaluated this BCI with two young healthy subjects. They were selected for their rare ability to use the μ/β BCI almost without training, hence avoiding the need for a lengthy training program. The first subject reported that he does not know how he is controlling the cursor or at least he could not explain it. The second subject reported that he was imagining himself walking in a straight line and turning to the left or to the right to control the cursor.

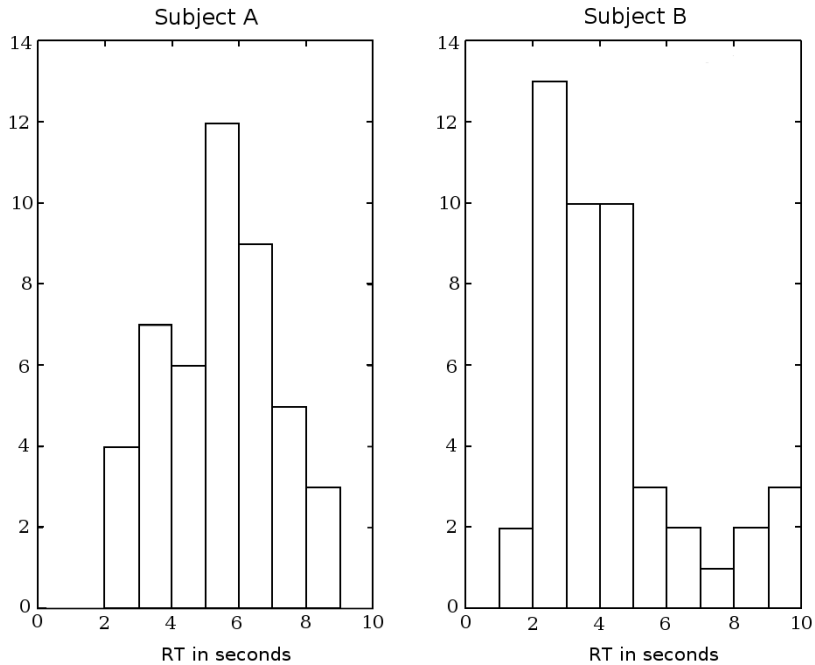


Figure 6.4: Distribution of response times for the μ/β stop BCI with our two healthy subjects.

We then recorded the time needed to move the cursor beyond a threshold, fixed at 30% of maximal power on each side, upon presentation of an audio clue. Figure 6.4 shows the distributions of response times for the two subjects. Table 6.3 gives the parameters of these distributions.

Table 6.3: RT statistics (in seconds) with the μ/β stop BCI.

Subject	min	max	mean	std
A	2.2	8.7	5.4	1.7
B	1.1	9.8	4.3	2.6

To evaluate the false acceptance rate, i.e. the number of times the cursor moves beyond the threshold while the user is not intending to, the subjects were asked to relax and to not try to control the cursor for 5 minutes. For none of them the cursor moved beyond the threshold, leading to an estimated false acceptance rate of 0.

These initial results were obtained with healthy subjects that were only briefly trained. Wolpaw & McFarland [45] obtained better results with 4 disabled subjects that were extensively trained: the subjects managed to move a cursor from the center of the screen to one of eight targets in less than 3 seconds (see Figure 2.5). Therefore, it can be expected that with appropriate training, disabled subjects could obtain similar response times using our μ/β-BCI.

6.3 Hybrid Interface

We propose to combine two modalities in an hybrid BCI: the P300-BCI for destination selection, and a faster modality for stopping. The BCI switches between these two modalities depending on the state of the wheelchair: we use the P300 modality

to allow the user to select a destination and start motion when the wheelchair is stopped, and when the wheelchair is moving we use the faster modality to give the user the ability to stop it. This control strategy is illustrated on Figure 6.5.

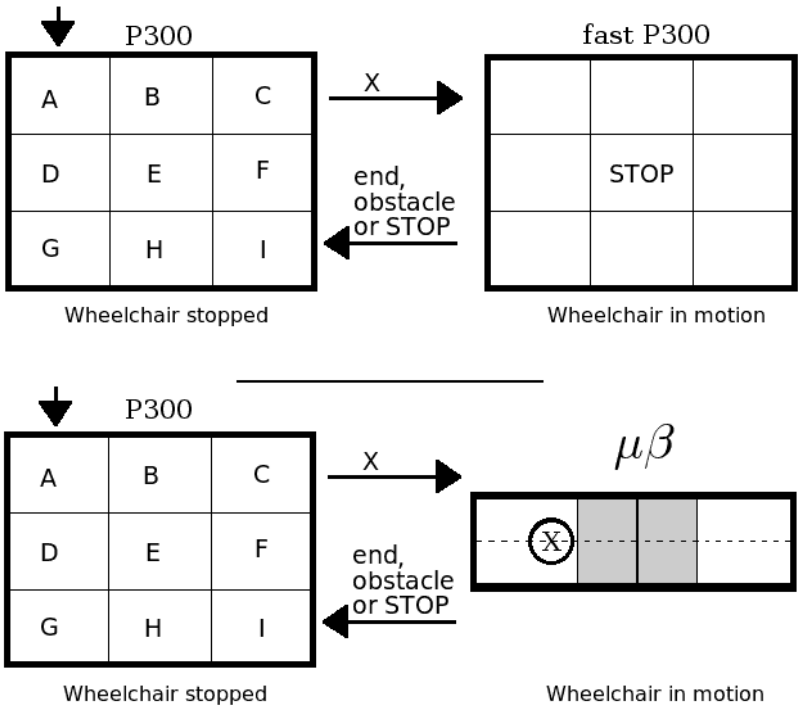


Figure 6.5: Control diagram with the hybrid BCI. When stopped, the BCI shows the list of destinations. Upon selection, the wheelchair starts moving and the BCI switches to the fast BCI for stopping (fast P300 or μ/β).

We presented two fast BCIs that can be used for stopping. Both have a response time of about 5 seconds. The fast P300 BCI has a relatively high false acceptance rate (1 every 30 seconds) whereas the μ/β BCI's FA is null. On the other hand, the P300 paradigm is easy to use and does not require training, whereas the μ/β BCI requires a lot of training and concentration to use it. Therefore we decided to offer

both BCIs and let the user decide which is more satisfying according to his or her abilities.

6.4 On-line Evaluation

We conducted on-line experiments with healthy subjects (5 for P300, and 2 for μ/β) seating on the wheelchair and moving on a circular guiding path.

In a first experiment [82] the wheelchair was launched and the subject was required to issue a stop command as fast as possible after the apparition of a clue. Top panel of Figure 6.6 shows the CDF of P300 response times for all five subjects. Figure 6.7 shows the CDF of μ/β RTs for both subjects. The subjects managed to stop the wheelchair in all trials both for P300 and μ/β.

A second experiment examined the occurrence of false acceptance FA, i.e. when a STOP command was issued involuntarily. For this purpose, the subjects were required to not activate the STOP command and were observed during 2 minutes. Bottom panel of Figure 6.6 shows the CDF of P300 false stop times (i.e. time without false stops) for all five subjects. With the μ/β interface there was no false stop.

6.5 Comparison of Off-line and On-line Results

The perception of motion by the brain and the stress induced by seating on a moving robot are factors that might prevent the usage of our BCI to issue a stop command. Generally, we do not know of any study proving that EEG patterns would remain similar during motion as at static position, such that the same model can be used in the two conditions. To investigate this we compared the on-line and off-line data collected.

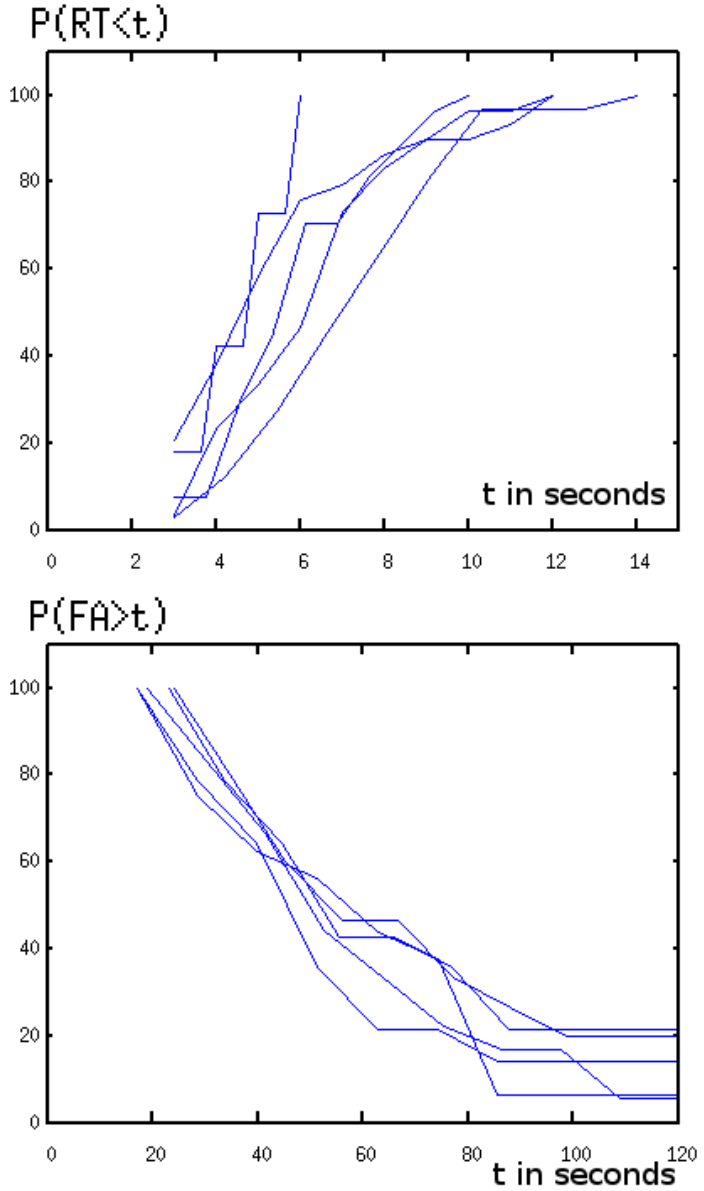


Figure 6.6: On-line evaluation of response time and false acceptance rate for the P300 stop BCI for the 5 subjects.

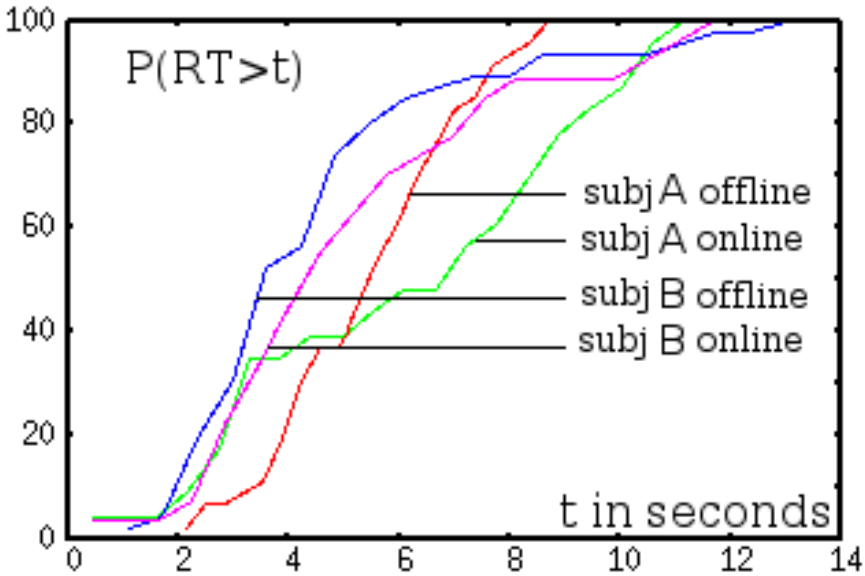


Figure 6.7: Off-line vs on-line comparison for the μ/β stop BCI.

Figure 6.8 shows the RT CDF (top) and the FA CDF (bottom) averaged over the five subjects, off-line and on-line. Figure 6.7 shows the μ/β RT distributions for our 2 subjects. These curves are very close one another and thus suggest that there is no much difference between on-line and off-line conditions.

To verify that point we conducted statistical comparison between the 2 conditions. We also ran a t-test between each pair of distribution. Table 6.4 shows the result of this comparison. It can be seen that mean times and standard deviations are close one to another and that p-values are well above 5%.

This analysis thus confirms the similarity between on-line and off-line results, hence showing that motion does not induce significant modification of the EEG that could prevent using a BCI to issue a stop.

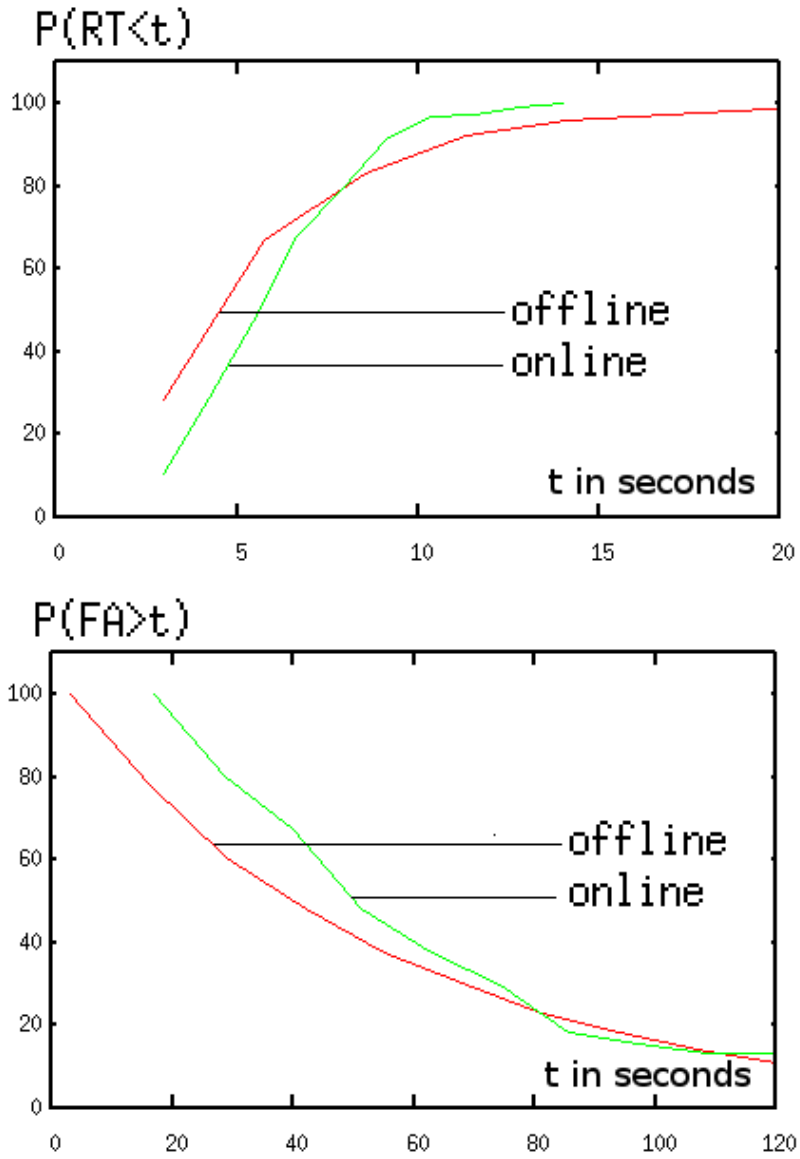


Figure 6.8: RT and FA for the P300 stop BCI, comparison of on-line and off-line data averaged on the 5 subjects.

Table 6.4: Off-line and on-line performances for the P300 and μ/β stop BCIs.

		RT (in seconds)			FA (in number of occurrences per 100 seconds)		
		off-line $\mu \pm \sigma$	on-line $\mu \pm \sigma$	p-value	off-line $\mu \pm \sigma$	on-line $\mu \pm \sigma$	p-value
P300	subject 1	4.3 ± 1.7	4.7 ± 1.1	0.21	2.0 ± 1.5	2.4 ± 1.3	0.34
	subject 2	5.3 ± 2.0	5.8 ± 1.9	0.27	2.5 ± 1.5	2.6 ± 1.5	0.77
	subject 3	7.4 ± 4.3	7.0 ± 2.3	0.60	1.6 ± 1.2	2.1 ± 1.2	0.13
	subject 4	6.1 ± 2.8	6.6 ± 2.3	0.32	2.1 ± 1.2	2.5 ± 1.6	0.30
	subject 5	6.7 ± 4.5	5.6 ± 2.4	0.19	2.1 ± 1.4	2.0 ± 1.1	0.80
	averaged	6.0 ± 3.4	5.9 ± 2.2	0.93	2.1 ± 1.4	2.3 ± 1.4	0.13
μ/β	subject A	4.3 ± 2.6	4.9 ± 2.7	0.330	0.0 ± 0.0	0.0 ± 0.0	1.00
	subject B	5.4 ± 1.7	7.9 ± 2.2	0.224	0.0 ± 0.0	0.0 ± 0.0	1.00
	averaged	4.9 ± 2.2	5.5 ± 3.0	0.159	0.0 ± 0.0	0.0 ± 0.0	1.00

6.6 Comparison of the μ/β and P300 BCIs for Stopping

Finally, we would like to compare the P300 interface with the μ/β interface. On Figure 6.9 we plot the RT CDF with both interfaces. These curves combine the data from all subjects, on-line and off-line. We can see that the responses are quite similar. However, a t-test between these 2 distributions returns a p value of 2% only.

Hence in term of response time, the μ/β and fast P300 BCIs are equivalent. There are two main differences however:

- The fast P300 BCI suffers from a non null false alarm rate which will cause the wheelchair to stop involuntarily. The μ/β does not have this problem ($FA = 0$).
- The μ/β BCI is difficult to use and requires long training, whereas the P300 BCI relies on a natural mechanism on the brain and as such requires almost no training.

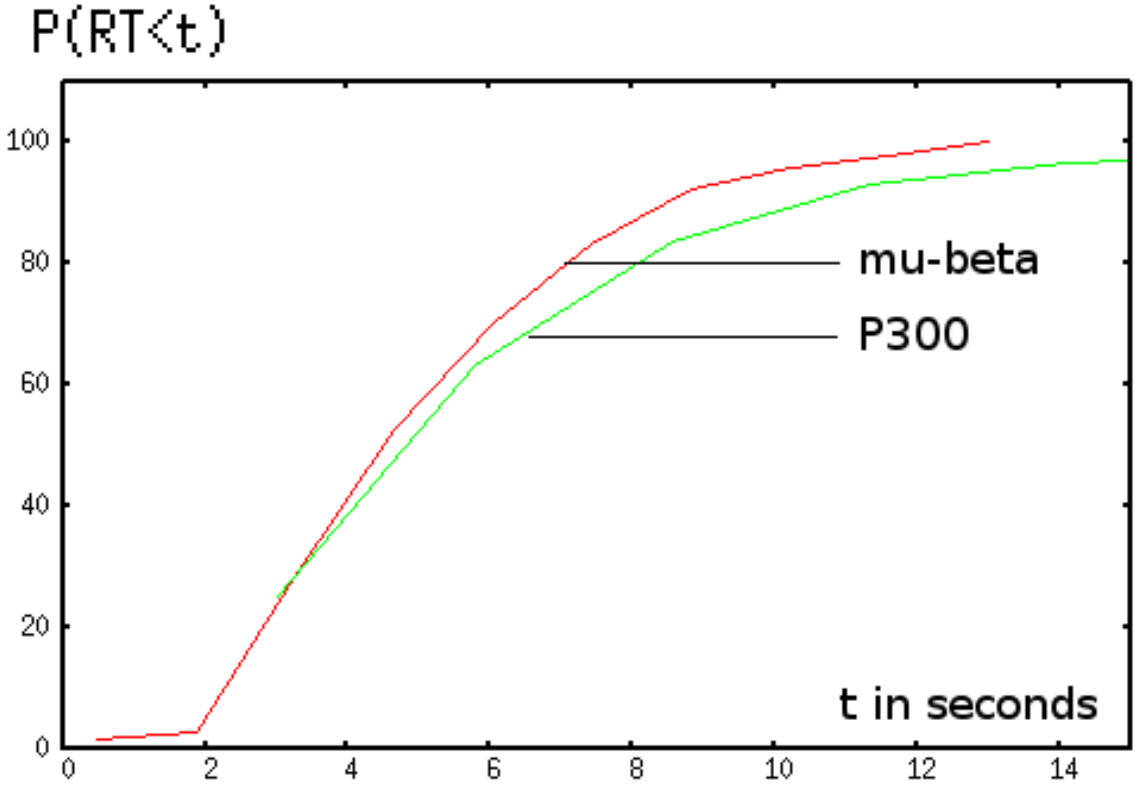


Figure 6.9: Comparison of μ/β and P300. RTs are averaged over all subjects, on-line and off-line together.

As a conclusion, the choice between the μ/β BCI and the fast P300 one will be based on user preference and ability.

CHAPTER 7

Overall Strategy Evaluation

The aim of this chapter is to evaluate our strategy and compare it with the brain controlled wheelchair projects presented in Section 2.3.

7.1 Experimental Setup

Depending on the option chosen for the stopping the wheelchair, we have the two possible scenarios. In the first one (Scenario A), we consider that there is no false stop. This corresponds to either using the μ/β interface ($FA = 0$), or to disabling the stop feature. In the other one (Scenario B), the P300 stop interface is active, hence we will have some false stops from time to time. If a false stop occurs before the wheelchair has completed the navigation, the user has to reselect the destination again.

We will run simulations based on measured performances of our BCIs (selection times and false stop rates):

- results from off-line evaluation of the P300 selection interface for the 5 subjects (Figure 5.5).
- results from off-line evaluation of the P300 stop interface (Section 6.1), algorithm 1d with a depth of 2 (Figure 6.3).

- results from on-line evaluation of the P300 stop BCI (Section 6.4)

The benchmark task is to navigate between locations connected by $50m$ long paths, i.e. approximately the length of a smooth and direct trajectory in MAIA's evaluation tests (see Figure 7.2). The wheelchair's nominal velocity is $0.5m.s^{-1}$, hence the nominal travel time is 100 seconds.

7.2 Evaluation Metrics

Our first metric will be the *mission time*, i.e. the total traveling time, which is computed as follows. First, a selection time is picked randomly according to the RT distribution. For scenario A, the *mission time* is simply the selection time plus the nominal time (100s).

For scenario B, we choose randomly a false stop rate from the FA distribution, and compute the corresponding time to stop $T = 100/FA$ as well as the distance traveled by the wheelchair in this time $D = V * T$, with $V = 0.5m.s^{-1}$. If this distance is greater than $50m$, the wheelchair has reached the destination without stopping. If the distance is smaller than $50m$, the wheelchair has stopped before reaching its destination and has to be restarted, so we again pick a selection time. The simulation is completed when the total distance reaches $50m$.

In an attempt to measure the *mental workload* required to control the wheelchair, we choose our second metric to be the *concentration time*, i.e. the time spent controlling the BCI.

Finally, in order to compare the different projects independently of the used scenario, we introduce the following ratios, which added together form a cost function.

mission time ratio : ratio of mission time to nominal time ($100s$).

concentration time ratio : ratio of the concentration time (i.e. time spent controlling the BCI) to nominal time.

Other important factors mentioned in the introduction, such as the comfort and safety factors could be added to this cost function, however their measurement would be somehow arbitrary, thus we decided to not consider them in this evaluation.

7.3 Results

Figure 7.1a shows the CDF of the *mission time* over 500 trials. In scenario A, the mission time is 112 seconds on average; in scenario B, it is 128 seconds with an average of 1.2 false stops. Table 7.1 shows the value of our two metrics and the cost function for our two scenarios.

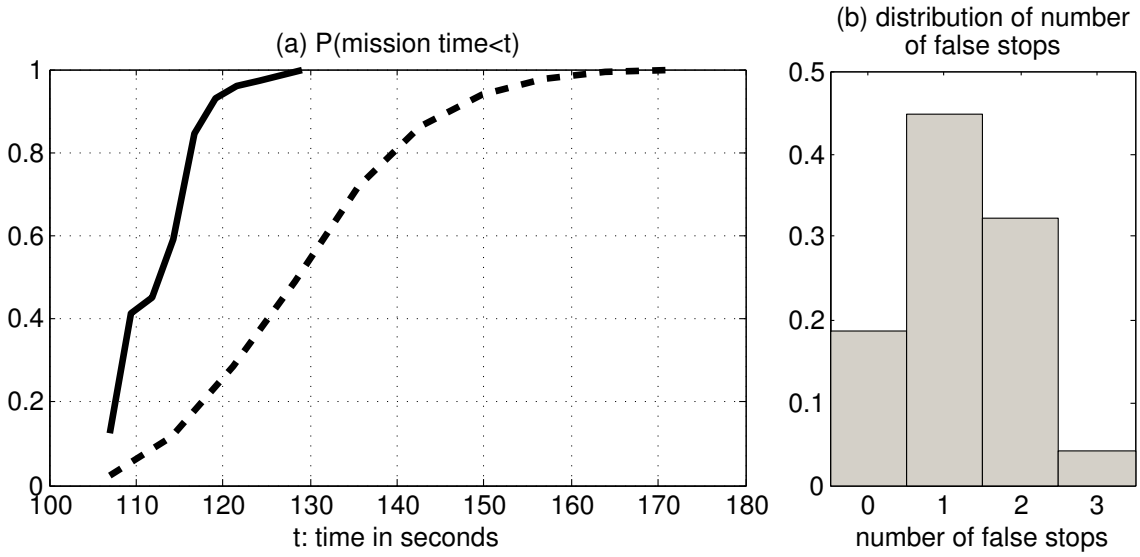


Figure 7.1: a) Mission time CDF in scenario A (no false stop, solid line) and B (with false stops, dashed line). b) Distribution of number of false stops per run in scenario B.

Table 7.1: Metrics to evaluate the overall strategy.

	Scenario A				Scenario B			
	min	max	mean	std	min	max	mean	std
mission time	107	129	112	5.0	107	177	128	13.2
miss. time ratio	1.07	1.29	1.13	0.05	1.07	1.77	1.28	0.13
concentration time	7	29	12.6	5.0	7	77	28.3	13.2
conc. time ratio	0.07	0.29	0.13	0.05	0.07	0.77	0.28	0.13
total cost	1.14	1.58	1.25	0.10	1.14	2.54	1.57	0.26

7.4 Comparison

Results from the MAIA project [56] are shown on Figure 7.2. Due to the nature of their strategy (exposed in [56] and in Section 2.3.3), trajectories are far from being optimal. Average trajectory times range from 130 to 270 seconds (bottom panel of Figure 7.2) depending on subject and active behaviors. We will take $200s$ as their mean mission time, which yields a mission time ratio of 2. Since the control of the wheelchair requires continuous concentration, we will take 200 seconds for the concentration time, yielding a concentration time ratio of 2. Finally, the total cost is 4.

It is possible to evaluate the cost of the Toyota/Riken strategy (see [59] and Section 2.3.4). Their subject managed to drive the wheelchair on an eight-shaped course in $22.88s$ on average (standard deviation $\pm 0.16s$). While driving on the same course with a joystick, the average time was $16.96s$ (standard deviation $\pm 0.086s$), hence the mission time ratio is 1.35. Since the control of the wheelchair requires continuous concentration, we will take $22.88s$ for the concentration time, yielding a concentration time ratio of 1.35. Finally, the total cost is 2.70.

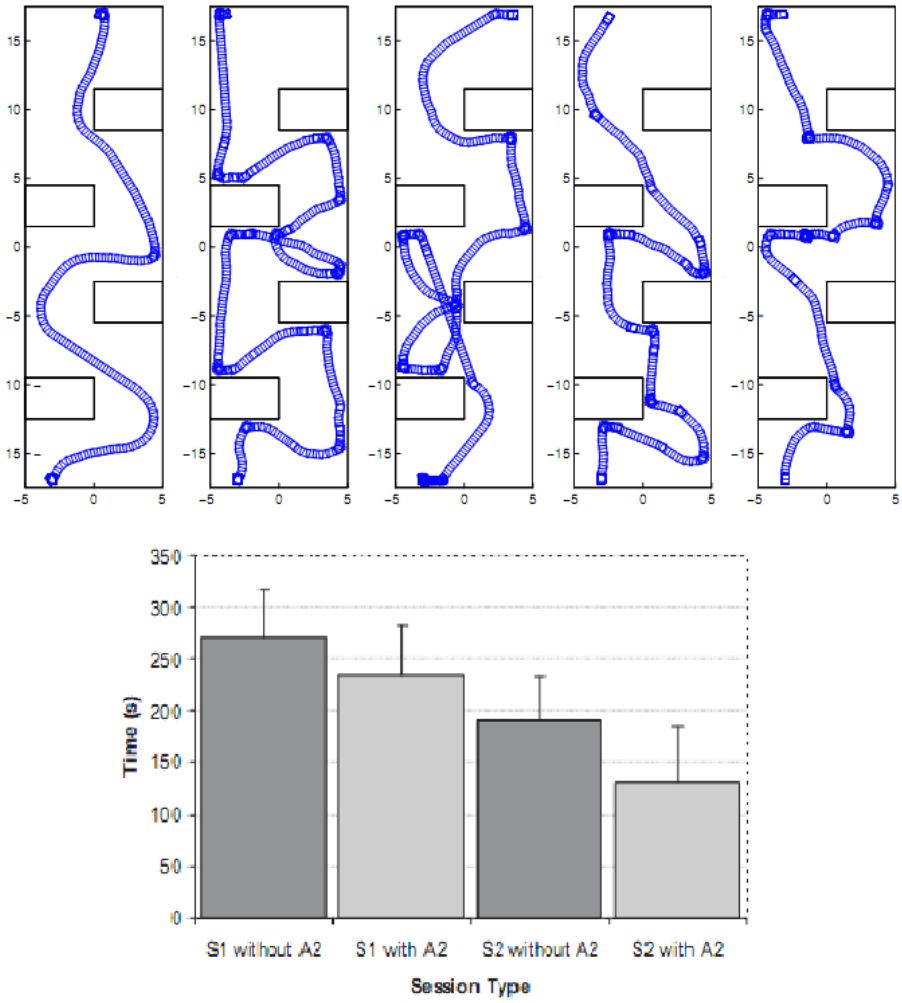


Figure 7.2: Results from MAIA project for 2 able subjects (from [56]). Top: wheelchair trajectories. Bottom: averaged elapsed time.

Data for the Minguez brain controlled wheelchair (see [55] and Section 2.3.2) is also available. They evaluated their system on two different circuits. The first circuit was designed to accomplish complex maneuverability tasks and avoidance of obstacles in constrained spaces (length of optimal path: $12m$). The second circuit involved navigation in open spaces (length of optimal path: $32m$). The respective mean mission times were 571 and 659 seconds, and the respective concentration time were 447 and 439 seconds. From this data, the mission time ratios can be computed as 25 and 10.3, and the concentration time ratios as 18.6 and 6.8. Finally, the total costs are 43.6 and 17.1.

Table 7.2 summarizes for each of those strategy the evaluated cost, and compares with our strategy. In this table, the third (respectively fourth) column is relative cost difference with Scenario A (respectively B):

$$Gain_A = \frac{C - C_A}{C_A}$$

$$Gain_B = \frac{C - C_B}{C_B}$$

Table 7.2: Comparison of strategy costs.

	cost	gain w.r.t A	gain w.r.t. B
Scenario A	1.25	0%	-20%
Scenario B	1.57	25%	0%
MAIA	4	220%	155%
Toyota	2.7	116%	72%
Minguez (1)	43.6	3388%	2677%
Minguez (2)	17.1	1268%	989%

7.5 Discussion

We compared mission times between our project and the MAIA project to put the two projects in perspective. We showed that, in term of mission time, the two projects are equivalent (although we are slightly faster). However, the difference between the two strategies is more fundamental: MAIA's wheelchair allows more freedom of movement but requires constant concentration, whereas in our strategy emphasis was put on minimizing the control input hence user fatigue.

Using a cost function that takes into account mission time and mental effort we showed that our project is the most efficient one. Again, this is only a partial comparison, and it would be absurd to compare those projects based only on that measure. Other factors should be taken into consideration, such as the freedom of movement allowed by the strategy (in our case, the wheelchair can only go to predefined locations), comfort and safety factors, financial cost of the equipment, amount of training required to use the BCI, etc.

Finally, we acknowledge that we are comparing simulation results with results obtained from real experiments. However, we believe that our simulation is fair for two reasons. First, to compute the operation time, we draw selection times and false acceptance rates from experimental data. Second, our wheelchair follows pre-defined guiding paths, ruling out the presence of obstacles on the way (same as with the other projects).

CHAPTER 8

Conclusion

8.1 Summary

This work developed a brain controlled wheelchair for navigation in familiar environments. It is intended as a realistic mobility recovery tool for locked-in people for whom a BCI is the only way to communicate or control a device. Since using a BCI requires a concentration effort that may exhaust the user, emphasis was put on minimizing the control input. To provide security the navigation is based on following guiding paths that were designed by a human helper, hence hazard free.

Locations of interest are first defined in the familiar environment of the user. A helper traces the guiding paths between them by pushing the wheelchair along the desired trajectories while the on board computer records them. The guiding paths are virtual, which allows easier maintenance and avoids modifying the environment.

Using the BCW is straightforward. The wheelchair offers a list of locations to the user. The selection process is simplified by reducing the amount of choices: only the locations connected to the current one, together with few relevant commands, are displayed. The user selects the desired destination using the BCI, and the wheelchair autonomously follows the appropriate guiding path to its end.

For the BCW, destination selection is done with a P300 EEG BCI. An electrode cap records the electrical activity of the brain while a small and portable acquisition device amplifies and digitalizes the signal for computer analysis. The items to be selected are displayed on screen in a table and flashed one by one in a random order. To select an item, the user simply counts the number of times this item is flashed while ignoring other items. A peak appears in the EEG approximately 300 milliseconds after the item of interest has been flashed while it remains unchanged for other items. This peak, the P300 signal, can be detected by a trained support vector machine algorithm, and the corresponding item is traced back as the one that was flashed 300 ms earlier.

This P300 based selection process takes 15 seconds on average (median = 12 s), which makes it impossible to use it to stop the wheelchair. For that purpose two faster BCIs were proposed: a fast single item P300 algorithm and a binary $\mu\beta$ BCI. Both allow to stop in 5 seconds on average, but vary in their false positive rate and their ease of use. A user chooses one of them according to his or her ability and preferences.

Finally, the overall strategy was evaluated using a cost function and results were compared to data from other brain controlled wheelchair projects. The cost function takes into account the total time of the motion and the time spent controlling the BCI was used. We saw, in terms of that cost function only, our strategy is clearly superior. However, this is only a partial comparison: other factors should be taken into consideration, such as the freedom of movement allowed by the strategy, comfort and safety factors, financial cost of the equipment, amount of training required to use the BCI, etc.

8.2 Discussion

In Section 1.3 we stated that in our opinion, a robotic wheelchair, to be successful, had to comply with the following constraints:

- Safety: especially since it transports a particularly vulnerable person.
- Ergonomy: the wheelchair should provide intuitive and efficient navigation with a minimum of efforts.
- Low cost: so that people who need it can afford it.

Safety is insured by the use of human designed paths. We believe that following human designed guiding path is much safer and more reliable than relying on artificial sensing and reasoning. Due to the large variety of traps and hazards in a home environment, there could always be a case when the AI would fail, no matter how many sensors used. However, if an unexpected situation occurs, such as an obstacle on the guiding path, the safety depends on the ability of the user to issue a stop command in time.

The ergonomy constraint is also fulfilled by the use of human designed paths which provide smooth and natural trajectories. Moreover, since navigating with the BCW is reduced to destination selection, the strategy minimizes the control effort. However, the user is constrained to a finite number of locations, which may prove frustrating at times.

The overall cost of the equipments required by the BCW, excluding the BCI and the wheelchair itself, is very low. One only needs a laptop, a DAQ card, two rotary encoders and a bar-code scanner. Only few modifications need to be done on the wheelchair.

Experiments in this thesis were conducted solely with young healthy subjects. Experiments with locked-in patients could not be conducted, due to the extreme difficulty of finding suitable volunteers. Nevertheless, we argue that our results should transfer, possibly with a minor performance decrease, to locked-in subjects. Indeed, we saw in Section 2.2.6 that various studies show that P300- and $\mu\beta$-based BCIs are usable by severely disabled patients and that their performances are similar to those of able subjects.

8.3 Recommended Directions

A few sensors could be added on the wheelchair to prevent colliding with obstacles when the user fails to issue a stop command in time. Bumpers could be a good start. Using proximity sensors might cause more problems that it solves: it may prevent going under a desk or close to a wall while the user desires so.

The real time part of the control and communication with the hardware could be ported to a small micro chip board with a serial link. This would allow control from any laptop, and would probably facilitate integration of the BCW with other BCI related applications (web surfing, games, ...).

On the BCI side, new technology is available every year, which should lead to interesting improvements. A much awaited one is the apparition soon on the market of new sensors for EEG acquisition, such as Emotiv's EPOC (Figure 8.1), designed for the video game market. These devices do not require conductive gel, hence should be easier to use and more comfortable. Besides they should be smaller and cheaper than current devices. However, the quality of the measurement will have to be assessed first.

Figure 8.1: Emotiv EPOC sensor for brain control of video games (to be released soon).

Recently the market saw the apparition of micro computers such as Beagleboard or Gumstix (see Figure 8.2). These take advantage of cell phone technology to deliver laptop-like computing power from a tiny chip, at a very low price and with a very low power consumption. It could be possible to use such computers to control the wheelchair and process the EEG data. This would allow to produce a simpler, neater and cheaper system.

Finally, we have been considering for quite some time the possibility of creating a mobility platform to transform a manual wheelchair into a powered wheelchair. The starting point of the discussion is that locked in people may already have a wheelchair, expensively customized to their morphology and needs, but most probably a manual one, pushed by a relative or care giver, since they do not have the ability to drive

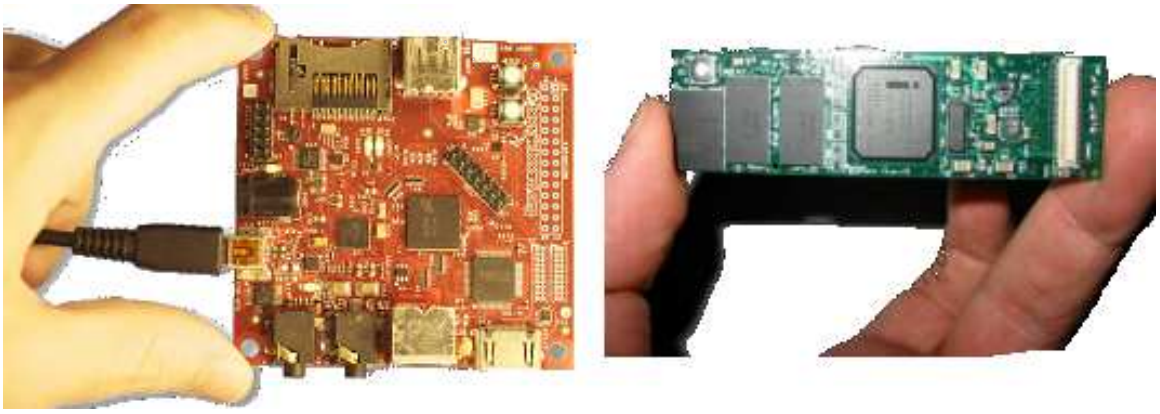


Figure 8.2: The Beagleboard (www.beagleboard.org) and Gumstix (www.gumstix.com) mico-computers could be used to replace the laptops on a future version of the prototype.

it themselves. In that situation, we would like to provide some minimally invasive way of powering their wheelchair. One idea would be to use a mountable power module such as the one on Figure 8.3 developed at Ngee Ann Polytechnic, Singapore. This module includes two motorized wheels, a battery and a controller, and provides enough power to drive the wheelchair indoor and on flat ground. It is easily mounted on a wheelchair and is quite inexpensive.

Figure 8.3: A wheelchair powering module by Ngee Ann Polytechnic, Singapore.

BIBLIOGRAPHY

[1] R. Ficke, "Digest of data on persons with disabilities," tech. rep., Washington, DC: US Department of Education, National Institute on Disability and Rehabilitation Research, 1991.

[2] N. A. B. on Medical Rehabilitation Research, "Report and research plan for the national center for medical rehabilitation research," tech. rep., National Institute of Child Health and Human Development, National Institute of Health, 1992.

[3] H. A. Yanco, *Shared user-computer control of a robotic wheelchair system.* PhD thesis, Massachusetts Institute of Technology, september 2000.

[4] E. Demeester, M. Nuttin, D. Vanhooydonck, and H. V. Brussel, "Assessing the users intent using bayes rule: Application to wheelchair control," *proceedings of the ASER international workshop on advances in service robotics*, pp. 117–124, 2003.

[5] D. Vanhooydonck, E. Demeester, M. Nuttin, and H. Van Brussel, "Shared control for intelligent wheelchairs: an implicit estimation of the user intention," *proceedings of the international workshop on advances in service robotics*, pp. 176–182, 2003.

[6] R. Barea, L. Boquete, E. López, and M. Mazo, "Guidance of a wheelchair using electrooculography," *Advances in intelligent systems and computer science*, 1999.

[7] Y. Kuno, N. Shimada, and Y. Shirai, "Look where you are going: A robotic wheelchair based on the integration of human and environmental observations," *IEEE Robotics and Automation Magazine*, vol. 10, no. 1, pp. 26–34, 2003.

[8] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, and J. Borenstein, "The navchair assistive wheelchair navigation system," *IEEE transactions on rehabilitation engineering*, vol. 7, pp. 443–451, December 1999.

[9] K. A. Tahboub and H. H. Asada, "A semi-autonomous control architecture applied to robotic wheelchairs," *IEEE/RSJ International Con- ference on Intelligent Robots and Systems*, pp. 906–911, 1999.

[10] J. Connell and P. Viola, "Cooperative control of a semi-autonomous mobile robot," *IEEE International Conference on Robotics and Automation*, pp. 1118–1121, 1990.

[11] G. Bourhis, K. Moumen, P. Pino, S. Rohmer, and A. Pruski, "Assisted navigation for a powered wheelchair," *IEEE International Conference on Systems Man and Cybernetics*, no. 3, pp. 553–558, 1993.

[12] J. Chapin, K. Moxon, R. Markowitz, and M. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, vol. 2, pp. 664–670, 1999.

[13] M. Nicolelis, "Brain-machine interfaces to restore motor function and probe neural circuits," *Nature Rev. Neurosc.*, vol. 4, pp. 417–422, 2003.

[14] M. J. Black, E. Bienenstock, J. P. Donoghue, M. Serruya, W. Wu, and Y. Gao, "Connecting brains with machines: the neural control of 2d cursor movement," *proceedings of the 1st international IEEE EMBS conference Neural engineering*, pp. 580–583, 2003.

[15] P. R. Kennedy, R. Bakay, M. Moore, K. Adams, and J. Goldwaithe, "Direct control of a computer from the human central nervous system," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, pp. 198–202, June 2000.

[16] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, July 2006.

[17] N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann, "Neurophysiological investigation of the basis of the fmri signal," *Nature*, vol. 412, pp. 150–157, 2001.

[18] G. Gratton, M. Goodman-Wood, and M. Fabiani, "Comparison of neuronal and hemodynamic measures of the brain response to visual stimulation: an optical imaging study," *Human brain mapping*, vol. 13, no. 1, 2001.

[19] V. A. and B. Chance, "Non-invasive optical spectroscopy and imaging of human brain function," *Trends in Neuroscience*, vol. 20, pp. 435–442, 1997.

[20] E. Gratton, S. Fantini, M. Franceschini, G. Gratton, and M. Fabiani, "Measurements of scattering and absorption changes in muscle and brain," *Philosophical Transactions: Biological Sciences*, vol. 352, no. 1354, pp. 727–735, 1997.

[21] V. Toronov, S. Walker, R. Gupta, J. Choi, E. Gratton, D. Hueber, and A. Webb, "The roles of changes in deoxyhemoglobin concentration and regional cerebral blood volume in the fmri bold signal," *Neuroimage*, vol. 19, no. 4, pp. 1521–1531, 2003.

[22] W. J. Freeman, *Mass action in the nervous system: Examination of the neurophysiological basis of adaptive behavior through EEG.* Academic Press, New York, N.Y, U.S.A., 1975.

[23] D. Cohen, "Magnetoencephalography: evidence of magnetic fields produced by alpha rhythm currents," *Science*, vol. 161, pp. 784–6, 1968.

[24] D. Cohen, "Magnetoencephalography: detection of the brain's electrical activity with a superconducting magnetometer," *Science*, vol. 175, pp. 664–6, 1972.

[25] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography¯theory, instrumentation, and applications to noninvasive studies of the working human brain," *Rev. Mod. Phys.*, vol. 65, pp. 413–497, Apr 1993.

[26] H. Berger, "Uber das electrenkephalogramm des menchen," *Arch Psychiat*, 1929.

[27] E. Niedermeyer and F. Lopes da Silva, eds., *Electroencephalography. Basic principals, clinical applications, and related elds.* Williams & Wilkins, 1999.

[28] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. MacFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, "Brain-computer interface technology: A review of the first international meeting," *IEEE transactions on rehabilitation engineering*, 2000.

[29] N. Birbaumer, T. Hinterberger, A. Kubler, and N. Neumann, "The thought-translation device (ttd): Neurobehavioral mechanisms and clinical outcome," *IEEE transactions on Neural Systems and rehabilitation engineering*, vol. 11, pp. 120–123, June 2003.

[30] S. Sutton, M. Braren, J. Zubin, and E. R. John, "Evoked-potential correlates of stimulus uncertainty," *Science*, vol. 150, no. 3700, pp. 1187–1188, 1965.

[31] M. Fabiani, G. Gratton, D. Karis, and E. Donchin, "Definition, identification, and reliability of measurement of the p300 component of the event-related brain potential," *Advances in Psychophysiology*, vol. 2, pp. 1–78, 1987.

[32] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, no. 70, pp. 510–523, 1988.

[33] D. Lange, H. Pratt, and G. Inbar, "Modeling and estimation of single evoked brain potential components," *IEEE transactions on BioMedical Engineering*, vol. 44, no. 9, pp. 791–799, 1997.

[34] E. Donchin, K. Spencer, and R. Wijesinghe, "The mental prosthesis: Accessing the speed of a p300-bases brain-computer interface," *IEEE transactions on rehabilitation engineering*, 2000.

[35] S. Makeig, T. Jung, A. Bell, D. Ghahremani, and T. Sejnowski, "Blind separation of auditory event-related brain responses into independent components," *Proceedings of the National Academy of Sciences*, vol. 94, no. 20, pp. 10979–10984, 1997.

[36] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[37] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "Bci competition 2003 data set iib: Enhancing p 300 wave detection using ica-based subspace projections for bci applications," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1067–1072, 2004.

[38] H. Serby, E. Yom-Tov, and G. Inbar, "An improved p300-based brain-computer interface," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 13, no. 1, pp. 89–98, 2005.

[39] J. Aunon, C. McGillem, and D. Childers, "Signal processing in evoked potential research: averaging and modeling.," *Critical reviews in bioengineering*, vol. 5, no. 4, p. 323, 1981.

[40] S. Georgiadis, P. Ranta-aho, M. Tarvainen, and P. Karjalainen, "Single-trial dynamical estimation of event-related potentials: a kalman filter-based approach," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 8, pp. 1397–1406, 2005.

[41] W. Singer, "Synchronization of cortical activity and its putative role in information processing and learning," *Annual Review Physiology*, vol. 55, pp. 349–374, 1993.

[42] W. Kuhlman, "Eeg feedback training: enhancement of somatosensory cortical activity," *Electroencephalography and Clinical Neurophysiology*, vol. 45, pp. 290–294, 1978.

[43] J. R. Wolpaw and D. J. MacFarland, "Multichannel eeg-based brain-computer communication," *Electroencephalography and Clinical Neurophysiology*, vol. 90, pp. 444–449, 1994.

[44] G. E. Fabiani, D. J. MacFarland, J. R. Wolpaw, and G. Pfurtscheller, "Conversion of eeg activity into cursor movement by a brain-computer interface (bci)," *IEEE Transactions Neural Systems and Rehabilitation Engineering*, vol. 12, no. 3, pp. 331–338, 2004.

[45] J. R. Wolpaw and D. J. MacFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *PNAS*, vol. 101, pp. 17849–17854, December 2004.

[46] M. Cheng, X. Gao, and D. Xu, "Design and implementation of a brain-computer interface with high transfer rates," *IEEE transactions on biomedical engineering*, vol. 49, p. 1181, 2002.

[47] O. Friman, I. Volosyak, and A. Gräser, "Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces," *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 742–750, 2007.

[48] J. D. R. Millán, J. Mourino, M. Franze, F. Cincotti, M. Varsta, J. Heikkonen, and F. Babiloni, "A local neural classifier for the recognition of eeg patterns associated to mental tasks," *IEEE transactions on Neural Net.*, 2002.

[49] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *proceedings of the IEEE*, 2001.

[50] C. Neuper, G. Muller, A. Kubler, N. Birbaumer, and G. Pfurtscheller, "Clinical application of an eeg-based brain–computer interface: a case study in a patient with severe motor impairment," *Clinical Neurophysiology*, vol. 114, no. 3, pp. 399–409, 2003.

[51] A. Kubler, F. Nijboer, J. Mellinger, T. M. Vaughan, H. Pawelzik, G. Schalk, D. McFarland, N. Birbaumer, and J. R. Wolpaw, "Patients with als can use sensorimotor rhythms to operate a brain-computer interface," *Neurology*, vol. 64, no. 10, pp. 1775–1777, 2005.

[52] E. W. Sellers and E. Donchin, "A p300-based brain-computer interface: initial tests by als patients.," *Clin Neurophysiol*, vol. 117, pp. 538–48, Mar. 2006.

[53] F. Nijboer, E. W. Sellers, J. Mellinger, M. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, D. Krusienski, T. M. Vaughan, *et al.*, "A p300-based brain–computer interface for people with amyotrophic lateral sclerosis," *Clinical Neurophysiology*, vol. 119, no. 8, pp. 1909–1916, 2008.

[54] K. Tanaka, K. Matsunaga, and H. Wang, "Electroencephalogram-based control of an electric wheelchair," *IEEE Transactions on Robotics*, vol. 21, pp. 762–766, August 2005.

[55] I. Iturrate, J. Antelis, A. Kübler, and J. Minguez, "Non-invasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 614–627, 2009.

[56] J. Philips, J. D. R. Millán, G. Vanacker, E. Lew, F. Galán, P. Ferrez, H. V. Brussel, and M. Nuttin, "Adaptive shared control of a brain-actuated simulated wheelchair," *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 408–414, 2007.

[57] J. D. R. Millán, A. Hauser, and F. Renkens, "Adaptive brain interface-abi: Simple feature, simple neural network, complex brain-actuated devices," *Digital Signal Processing*, pp. 297–300, 2002.

[58] G. Dornhege, T. Hinterberger, J. D. R. Millán, D. J. MacFarland, and K.-R. Müller, "Toward brain-computer interfacing," *MIT Press*, Sept. 2007.

[59] K. Choi and A. Cichocki, "Control of a wheelchair by motor imagery in real time," in *Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning*, pp. 330–337, Springer, 2008.

[60] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," pp. 167–193, 1990.

[61] J. Leonard and H. Durrant Whyte, "Mobile robot location by tracking geometric beacons," *IEEE Trans. Robotics and Automation*, vol. 7, pp. 376–382, 1991.

[62] Q. Zeng, B. Rebsamen, E. Burdet, and C. L. Teo, "Design of a collaborative wheelchair with path guidance assistance," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 877–882, 2006.

[63] A. Micaelli and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," tech. rep., INRIA, 1993.

[64] C. Samson, "Path following and time-varying feedback stabilization of a wheeled mobile robot," *Proceedings of the international conference on advanced robotics and computer vision*, vol. 13, pp. 1.1–1.5, 1992.

[65] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of robot Control*, ch. 9, p. 241. Springer, 1996.

[66] B. Gillespie, J. E. Colgate, and M. Peshkin, "A general framework for cobot control," *IEEE transactions on Robotics and Aut.*, pp. 391–401, 2001.

[67] M. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. Moore, and B. Gillespie, "Cobot architecture," *IEEE transactions on Robotics and Aut.*, pp. 377–390, 2001.

[68] Q. Zeng, E. Burdet, B. Rebsamen, and C. L. Teo, "A collaborative wheelchair system," *IEEE Transactions Neural Systems and Rehabilitation Engineering*, vol. 16:2, pp. 161–170, 2008.

[69] E. S. Boy, E. Burdet, and C. L. Teo, "Collaborative wheelchair assistant," *proceedings of the international conference on Intelligent Robots and System (IROS)*, vol. 2, pp. 1511–1516, 2002.

[70] E. S. Boy, E. Burdet, C. L. Teo, and J. E. Colgate, "Investigation of motion guidance with scooter cobot and collaborative learning," *IEEE Transactions on Robotics*, vol. 23, pp. 245–255, April 2007.

[71] Q. Zeng, E. Burdet, B. Rebsamen, and C. L. Teo, "Evaluation of the collaborative wheelchair assistant system," *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 601–608, 2007.

[72] B. Long, B. Rebsamen, E. Burdet, H. Yu, and C. L. Teo, "Elastic path controller for assistive devices," *proceedings IEEE international conference of the Engineering in Medicine and Biology Society (EMBS)*, 2005.

[73] B. Long, B. Rebsamen, E. Burdet, and C. L. Teo, "Development of an elastic path controller," *proceedings of IEEE international conference on Robotics and Automation (ICRA)*, pp. 493–498, 2006.

[74] L. Zhou, C. L. Teo, and E. Burdet, "Development of a novel elastic path controller," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 1596–1601, Oct. 2007.

[75] L. Zhou, C. L. Teo, and E. Burdet, "Analysis and parameter optimization of an elastic path controller," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 789–794, 29 2007-Nov. 2 2007.

[76] L. Zhou, C. L. Teo, and E. Burdet, "A nonlinear elastic path controller for a robotic wheelchair," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, pp. 142–147, June 2008.

[77] Q. Zeng, C. L. Teo, B. Rebsamen, and E. Burdet, "Collaborative path planning for a robotic wheelchair.," *Disability and Rehabilitation: Assistive Technology*, vol. 3, pp. 315–324, Nov 2008.

[78] Q. Zeng, E. Burdet, B. Rebsamen, and C. L. Teo, "Experiments on collaborative learning with a robotic wheelchair," *iCreate*, pp. 57–62, 2007.

[79] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, pp. 297–298, 1999.

[80] H. Zhang, C. Guan, and C. Wang, "A statistical model of brain signals with application to brain-computer interface," *Proceedings of the 27th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 5388–5391, 2005.

[81] C. Guan, M. Thulasidas, and J. Wu, "High performance p300 speller for brain-computer interface," *IEEE International Workshop on Biomedical Circuits and Systems*, pp. S3/5/INV–S3/13–16, 2004.

[82] B. Rebsamen, E. Burdet, C. Guan, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier, "Controlling a wheelchair using a bci with low information transfer rate," *IEEE international conference on Rehabilitation Robotics (ICORR)*, pp. 1003–1008, 2007.

APPENDIX A

Mathematics of the Path Following Controller

We will here detail the mathematical derivation of the path following controller. The first step is to express the kinematic of the wheelchair in path coordinates. The second step is to obtain time-independent equations.

A.1 Coordinates Transformation

We will first study the kinematic of a point, then of the wheelchair.

Kinematic of a point

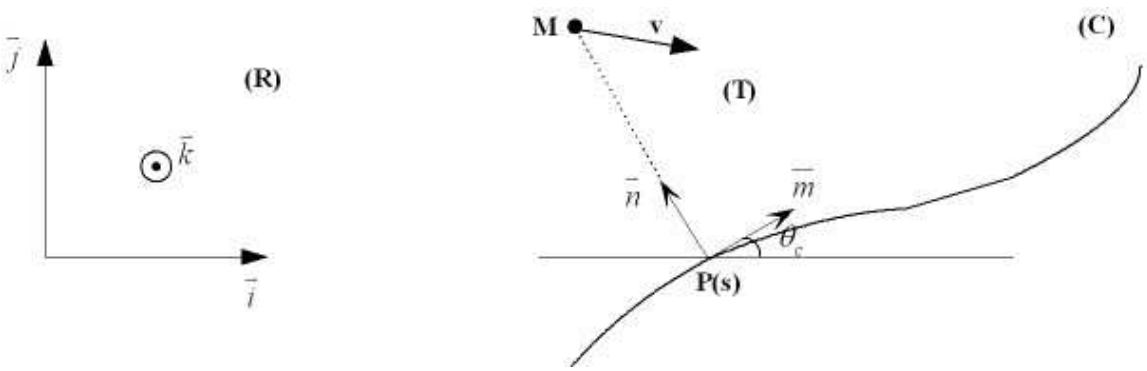


Figure A.1: Frames and notations for the controller.

Consider a moving point $\mathcal{M}$ and the associated Frenet frame $(\mathcal{T})$ defined on the curve $(\mathcal{C})$ as indicated in Figure A.1. The point $\mathcal{P}$ is the orthogonal projection of the point $\mathcal{M}$ onto the curve $\mathcal{C}$. And $\mathcal{O}$ is the origin of the global frame $\mathcal{R}$.

$$\left(\frac{d\overrightarrow{OM}}{dt}\right)_R = \left(\frac{d\overrightarrow{OP}}{dt}\right)_R + \left(\frac{d\overrightarrow{PM}}{dt}\right)_T + \overrightarrow{\Omega} \times \overrightarrow{PM} \tag{A.1}$$

We express this equation in frame (T) by multiplying by the rotation matrix R_T^R:

$$R_T^R = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Besides we have the following equalities:

$$R_T^R \left(\frac{d\overrightarrow{OP}}{dt}\right)_R = \left(\frac{d\overrightarrow{OP}}{dt}\right)_T = \overrightarrow{0},$$

since P is the origin of frame (T),

$$\left(\frac{d\overrightarrow{PM}}{dt}\right)_T = \begin{bmatrix} \dot{s} \\ \dot{y} \\ 0 \end{bmatrix},$$

$$\overrightarrow{\Omega} \times \overrightarrow{PM} = \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} -y\dot{\theta}_c \\ 0 \\ 0 \end{bmatrix},$$

$$\dot{\theta}_c = \frac{d\theta_c}{dt} = \frac{d\theta_c}{ds} \cdot \frac{ds}{dt} = c_c\dot{s},$$

where c_c is the curvature.

Equation A.1 becomes:

$$R_T^R \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{s} \\ \dot{y} \\ 0 \end{bmatrix} + \begin{bmatrix} -c_c y\dot{s} \\ 0 \\ 0 \end{bmatrix}$$

Expanding and rearranging we obtain the expression of $\dot{s}$ and $\dot{y}$:

$$\begin{cases} \dot{s} = (\cos\theta_c \quad \sin\theta_c) \cdot \binom{\dot{X}}{\dot{Y}} / [1 - c_c(s)y] \\ \dot{y} = (-\sin\theta_c \quad \cos\theta_c) \cdot \binom{\dot{X}}{\dot{Y}} \end{cases}$$

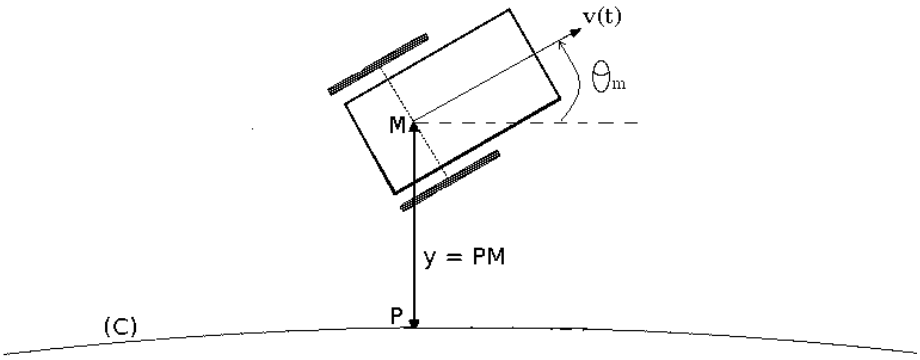


Figure A.2: Wheelchair's kinematics.

Kinematic of the wheelchair

Our wheelchair platform has two actuated wheels on a common axis and the reference point $\mathcal{M}$ at mid-distance of these two wheels (see Figure A.2), so the kinematic equations of this unicycle-type vehicle are as follows:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = v \cdot \begin{pmatrix} \cos\theta_m \\ \sin\theta_m \end{pmatrix} \tag{A.2}$$

From the above two functions, we have the following expression of unicycle expressed in coordinates $\{s, y\}$:

$$\begin{cases} \dot{s} = v\cos(\theta_m - \theta_c)/(1 - c_c y) \\ \dot{y} = v\sin(\theta_m - \theta_c) \end{cases} \tag{A.3}$$

For simplicity, we make $\theta = \theta_m - \theta_c$, so we have the kinematics function of the wheelchair expressed as (using $c_c = d\theta_c/ds = \dot{\theta}_c/\dot{s}$):

$$\begin{cases} \dot{s} = v\cos\theta/(1 - c_c y) \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \dot{\theta}_m - c_c\dot{s} \end{cases} \tag{A.4}$$

A.2 Time independent equations

To derive the control variable, we express the equations of motion with respect to the new variables $\eta = \int_0^t |\dot{s}|\, d\tau$ instead of the time-index t. This new variable has the physical meaning of the distance travelled by the vehicle along the path. From now on we will denote $\dfrac{d}{d\eta}$ as $(\ )'$.

s' is computed as follow:

$$s' = \frac{ds}{d\eta} = \frac{ds}{dt}\,\frac{dt}{d\eta} = \frac{\dot{s}}{d\eta/dt}.$$

Since:

$$\frac{d\eta}{dt} = \frac{d}{dt}\left(\int_0^t |\dot{s}|\, d\tau\right) = |\dot{s}|\,,$$

we have:

$$s' = sign(\dot{s}) = sign\left(\frac{v\cos\theta}{1 - c_c y}\right).$$

y' is computed as follow:

$$\begin{aligned}
y' &= \frac{dy}{d\eta} \\
&= \frac{dy}{dt}\,\frac{dt}{ds}\,\frac{ds}{d\eta} \\
&= \frac{\dot{y}}{\dot{s}}\, s' \\
&= \frac{v\sin\theta}{v\cos\theta/\left(1 - c_c y\right)}\, s' \\
&= \tan\theta\ \left(1 - c_c y\right)\ sign\left(\frac{v\cos\theta}{1 - c_c y}\right)
\end{aligned}$$

Similarly, we have:

$$\begin{aligned}
\theta' &= \frac{\dot{\theta}}{\dot{s}}\, s' \\
&= \left(\frac{w\left(1 - c_c y\right)}{v\cos\theta} - c_c\right)\ sign\left(\frac{v\cos\theta}{1 - c_c y}\right) \\
&= w\left|\frac{1 - c_c y}{v\cos\theta}\right| - c_c\ sign\left(\frac{v\cos\theta}{1 - c_c y}\right)
\end{aligned}$$

Finally, the kinematic model of the wheelchair can be expressed as below in term of the distance traveled:

$$\begin{cases} s' = sign(\frac{v\cos\theta}{1-c_c y}) \\ y' = \tan\theta \ \ (1 - c_c y) \ \ sign\left(\frac{v\cos\theta}{1-c_c y}\right) \\ \theta' = w\left|\frac{1-c_c y}{v\cos\theta}\right| - c_c \ sign\left(\frac{v\cos\theta}{1-c_c y}\right) \end{cases}$$

APPENDIX B

Code for Setting Up the Real Time Architecture

We will give here an overview of the code used for the setting up the RTAI system and creating the various components of the system, as described in Section 3.3.2.

The main function (see Listing B.1) forks off a child process which will be responsible of the real time tasks. The parent process handles all other tasks. Communication between the two processes is realized through a RTAI shared memory.

Listing B.2 shows the function realized by the real time process. The structure `struct wheelchairDataRT` contains all the data to be shared between the real time process and the non real time process (user process). It is allocated as a shared memory by the call to `rtai_malloc` which creates a shared memory referenced by the key `WCHDRT`. The function `setupRTAI` (see Listing B.3) creates a real time task

Listing B.1: The main function

```
1 int main(){
2     pid_t pid = fork();
3     if( pid>0 ) mainUser();
4     else        mainRTAI();
5     return 0;
6 }
```

Listing B.2: Main function of the real time process

```
struct wheelchairDataRT *wchDRT = NULL;
void mainRTAI(){
    //register a handler for CTRL-C
    signal(SIGINT, killprog);

    setupRTAI(); //see below

    // create a shared memory with key 'WCHDRT'
    wchDRT = rtai_malloc(nam2num("WCHDRT"),
        sizeof(struct wheelchairDataRT));

    //initialize the DAQ card and the Comedi API
    initHardware();

    //start the control loop (see below)
    wchDRT->RTAItime = 0;
    while ( wchDRT->RTAItime >= 0 ){
        if( !hardware_loop() ) break;
        rt_task_wait_period();
        wchDRT->RTAItime += 0.005
    }
    killprog(0);

    //free the shared memory
    rt_shm_free(nam2num("WCHDRT");

    // terminate the real time stuffs properly
    rt_make_soft_real_time();
    stop_rt_timer();
    rt_task_delete(mytask);
}
```

Listing B.3: The code to set up a real time task in RTAI

```
unsigned long mytask_name;
RT_TASK *mytask = NULL;
RTIME period;
struct sched_param mysched;

void setupRTAI(){
    mytask_name = nam2num("MASTER");
    mysched.sched_priority
            = sched_get_priority_max(SCHED_FIFO)-1;
    sched_setscheduler(0,SCHED_FIFO,&mysched );
    mlockall(MCL_CURRENT | MCL_FUTURE);
    mytask = rt_task_init(mytask_name,1,0,0);
    rt_set_periodic_mode();
    period = nano2count(5E6);
    start_rt_timer(period);
    rt_make_hard_real_time();
    rt_task_make_periodic(mytask,
            rt_get_time()+period, period);
}
```

set with the highest priority on a FIFO scheduler. It is configured as a periodic task, and the period is set as 5ms.

Once the real time task is set up properly and hardware initialized, the code enters the control loop. At each iteration the code

1. calls the `rt_task_wait_period` function which causes the process to sleep until the real time period has elapsed.

2. It then calls the `hardware_loop` function which reads the counters associated with the glide wheels rotary encoders, from which it computes the current position and velocity, and performs PD velocity control.

3. Finally it increments the value of `wchDRT->RTAItime` by the period, hence it holds the time elapsed since the loop started.

The computed position and velocity, as well as the desired velocity, are stored in `wchDRT` (the structure in the shared memory). The real time process writes the position and velocity, the user process writes the desired velocity.

Since the `hardware_loop` function is called periodically by RTAI, hence with a small jitter, the velocity is derived from the counter reading with good precision. This is also important for the D term of the PD controller which required good timing.

Listing B.4 shows the main function of the user process. Like the RT main function it starts by declaring an interrupt signal handler, connect to the shared memory, and do some initializations. Then it launches a couple of utility threads: one for an OpenGL visual feedback, one for logging data to a a file, one for handling the Ethernet communication with the BCI computer, and two to monitor the ultrasonic proximeter and the bar code scanner. Then it enters the main loop, which calls periodically the main control function `controlWheelchair`. Synchronization is realized with the `usleep` function, which suspends the thread for approximately 100ms. This function realizes a state machine which will execute at each iteration the control corresponding to the current state (for instance follow a guiding path), and check for the relevant events (i.e. a message from the BCI, or end of path reached).

Listing B.4: Main function of the user process

```
void mainUser(){
   //register a handler for CTRL-C
   signal(SIGINT, killprog);

   // create and connect to shared memory
   wchDRT = rtai_malloc(nam2num("WCHDRT"),
         sizeof(struct wheelchairDataRT));

   // do some system initialization
   initSyst();

   // launch some threads for visual feedback (OpenGL)
   // and monitoring of non-RT sensors
   pthread_create(&id,NULL,glThreadFunc,NULL);
   pthread_create(&id,NULL,sensorThreadFunc,NULL);
   pthread_create(&id,NULL,scannerThreadFunc,NULL);
   pthread_create(&id,NULL,dataLoggingThreadFunc,NULL);
   pthread_create(&id,NULL,ethernetThreadFunc,NULL);

   while( wchDRT->userProcessLoop ){
      if( !controlWheelchair() ) break;
      usleep(.1E6); //sleep for 100ms
   }

   killprog(0);
   //free the shared memory
   rt_shm_free(nam2num("WCHDRT"));
}
```

APPENDIX C

List of Publications

Many publications resulted from this Ph.D. work:

- B. Rebsamen, C. Guan, H. Zhang, C. Wang, C.L. Teo, M. Ang and E. Burdet (2009), *A Brain Controlled Wheelchair to Navigate in Familiar Environments*, IEEE Trans. on Neural System and Rehabilitation Engineering (TNSRE, submitted).

- B. Rebsamen, E. Burdet, C. Guan, C.L. Teo, Q. Zeng, C. Laugier and M. Ang (2007), *Controlling a wheelchair indoors using thought.* IEEE. Intelligent Systems 22: 18-24 (feature article).

- B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C.L. Teo, Q. Zeng, C. Laugier and M. Ang (2007), *Controlling a Wheelchair Using a BCI with Low Information Transfer Rate.* Proc IEEE. International Conference on Rehabilitation Robotics (ICORR).

- B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C.L. Teo, Q. Zeng, M. Ang and C. Laugier (2006), *A Brain-Controlled Wheelchair Based on P300 and Path Guidance.* Proc IEEE. / RAS-EMBS Int. Conf on Biomedical Robotics and Biomechatronics (BioRob) 1101-6.

Publications that this Ph.D. work has contributed to:

- B. Long, B. Rebsamen, E. Burdet, C.L. Teo (2005) *Elastic Path Controller for Assistive Devices.* Proc IEEE. Engineering in Medicine and Biology Conference (EMBC) 6239-42.

- B. Long, B. Rebsamen, E. Burdet and C.L. Teo (2006), *Development of An Elastic Path Controller.* Proc IEEE. International Conference on Robotics and Automation (ICRA) 493-8.

- Q. Zeng, E. Burdet, B. Rebsamen and C.L. Teo (2008), *A Collaborative Wheelchair System.* IEEE. Transactions on Neural Systems and Rehabilitation Engineering 16(2): 161-70.

- Q. Zeng, E. Burdet, B. Rebsamen and C.L. Teo (2008), *Collaborative Path Planning for a Robotic Wheelchair.* Disability and Rehabilitation: Assistive Technology (in press).

- Q. Zeng, C.L. Teo, B. Rebsamen and E. Burdet (2006), *Design of a Collaborative Wheelchair with Path Guidance Assistance.* Proc IEEE. International Conference on Robotics and Automation (ICRA) 877-82.

- Q. Zeng, E. Burdet, B. Rebsamen and C.L. Teo (2007), *Experiments on Collaborative Learning with a Robotic Wheelchair.* Proc International Convention for Rehabilitation Engineering and Assistive Technology (i-Create).

- Q. Zeng, E. Burdet, B. Rebsamen and C.L. Teo (2007), *Evaluation of the Collaborative Wheelchair Assistant System.* Proc IEEE. International Conference on Rehabilitation Robotics (ICORR).