University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Mobile Robot Assisted Navigation based on Collaborative Control

Ana Cristina Barata Pires Lopes

2012

Mobile Robot Assisted Navigation based on Collaborative Control

Ana Cristina Barata Pires Lopes

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Coimbra, Portugal

under supervision of

Prof. Dr. Urbano Nunes (advisor)
Prof. Dr. Aníbal Traça de Almeida (co-advisor)

To my parents

Acknowledgments

There are many people to whom I would like to express my appreciation and gratitude. First and foremost, I would like to thank my supervisors, Professor Urbano Nunes and Professor Aníbal Traça d'Almeida, for their guidance, encouragement, patience, and wisdom during these past few years. Professor Urbano's supervision work was outstanding throughout the PhD, and for his methodological contributions, fruitful discussions, continuous motivation, friendship, and for offering me all the necessary means to carry out this thesis, I am forever grateful. Professor Traça supervised me during my B.Sc, M.Sc and Ph.D. I owe him a great deal of my training as both student and researcher. It has indeed been a great privilege to work with such far-thinking and inspirational individuals. All I have learned and accomplished during the course of this thesis would not have been possible without their outstanding dedication.

During this Ph.D work, 11 people participated in the experiments, 1 of whom suffers from motor impairments. I'm deeply grateful to all of them. Special thanks go to Ricardo Parafita, Gabriel Pires and Vinicius Caridá, for their help during the experimental sessions.

I want to thank Dr. Carlos Barata, from the Associação de Paralisia Cerebral de Coimbra (APCC) board direction, who gave me the full support I needed. I would also like to thank the therapists and their indispensable assistance, their information, help, sympathy and patience during my many visits to the APPC.

I owe much to ISR for having provided the excellent conditions and resources that have allowed me to accomplish my Ph.D work, as I am also grateful to Instituto Politécnico de Tomar (IPT), where I have been Assistant Professor since 1999, for providing me with the conditions to conciliate, during the past five years, my lecturing activities with this Ph.D work. I acknowledge two partial research fellowships, one from Fundação para a Ciência e a Tecnologia (FCT) under Grant SFRH/BD/29655/2006, and the other from PROTEC, research fellowship under Grant SFRH/BD/49251/2008. This work has been in partially supported by Fundação para a Ciência e Tecnologia (FCT), under Grants FCT/POSI/EEA-SRI/58279/2004 (MTDTS04), PTDC/EEA-ACR/72226/2006 (PMITS06), and RIPD/ADA/109661/2009 (INTERFACE10).

Since joining the ISR in 1998 I have met many people from all over the world and I have worked with some of them. I am very grateful to my all friends, colleagues and professors from ISR, and I want to express a special gratitude for those with whom I have worked directly during my Ph.D, namely Gabriel Pires, Luís Vaz, Fernando Moita, and Razvan Solea. To my friends João Barreto and Lino Marques for their support and

fruitful discussions. To my friends and office mates Anabela Carvalho, Luís Conde, and Nuno Melo, for their support and friendship. To Augusto Figueiredo, Carlos Patrão, Pedro Martins, and Paula Fonseca for their encouragement and long-lasting friendship. To Ana Vieira, Pedro Correia, Luís Oliveira, Luís Almeida, Casimiro Baptista, and (again) Gabriel Pires, my friends and colleagues at IPT, for their support. I also want to express my gratitude to Paulo Coelho and Manuel Barros, whose help was decisive to conciliate the lecturing activities with the Ph.D work over the last year.

Finally, I would like to thank all my dearest friends and family, specially my mother Maria Isilda and my father Joaquim, whom I owe everything.

Abstract

This thesis proposes an Assisted Navigation System (ANS) for a Robotic Wheelchair (RW) relying on a sparse and discrete Human-Machine Interface (HMI), more precisely a P300-based Brain-Computer Interface (BCI), or, in alternative, a switch/multi-switch with a scanner interface that allows a selection of commands to steer a RW. The proposed ANS is primarily intended for people with severe motor disabilities who are not able to operate most of the commercial HMIs.

The ANS is based on a two-layer collaborative control approach that takes into account both human and machine information. The proposed collaborative controller complies with four essential design requirements: dialogue, awareness, self-reliance, and adaptiveness. The P300-based BCI, allowing a communication channel completely independent of any motor control requirement, appeared as a good choice for motor impaired users. However, due to the P300-based BCI system's low transfer rates, the user is only able to issue sparse and discrete commands over time. In this sense, to effectively use such HMI, we are proposing an ANS able to predict and execute user navigation intents with minimum information. This relates directly to two design requirements: awareness and self-reliance, meaning that the RW must be able to clearly identify situations where aid is required, and ask for help in these situations. Moreover, since users are only able to issue sparse information over time, the robot must be self-reliant and able to cope with dynamic changes in the environment without requiring any aid or, if it is really needed, it must be able to deal with unreliable and delayed information.

Our ANS includes a localization system, an obstacle detection module, and a local planner that were designed to deal with changes in the environment, providing some degree of self-reliance to the RW. We propose a planning strategy that can deal with low cluttered semi-structured environments. It comprises a global planner that provides a global path to reach a pre-defined goal, and a local planner that intervenes when changes in the environment are detected. We propose a modified VFH that is carried out in three stages: steering, path-planning and blending. Unlike the approaches proposed by [Borenstein 1991, Ulrich 2000], our proposal builds a polar histogram directly from laser scan information. Additionally, our approach is also able to blend global and local paths.

A Markov localization approach is also proposed. It was designed to fuse odometry with a new polar scan matching algorithm, composed by three main stages: scan preprocessing, virtual scan, and matching. The proposed matching algorithm uses the sample Pearson correlation coefficient to evaluate the similarities between current and virtual scans. The correlation factor is determined on polar coordinate space, leading to a reduction of computational complexity of this algorithm. We also propose an Extended Kalman Filter (EKF) to fuse the odometric data with the data from magnetic markers that are detected with a 3D algorithm. The latter is based on the Least Squares Fitting (LSF) of the measurement data with the 3D model of the vertical magnetic field.

The collaborative controller is also adaptive to the user's skillfulness in steering the assisted RW. To perform user characterization, we propose an Assistive Navigation training Framework (ANTF) that is able to characterize users, by sorting them into three different steering levels: beginner, average, and advanced. The ANTF is also intended for training users with disabilities to steer the RW.

Experimental results using RobChair, the RW platform developed at ISR-UC [Pires 2002, Lopes 2007] are here presented, showing the effectiveness of the proposed methodologies. The prototype was validated with ten able-bodied participants, and one disabled participant, in two different scenarios: a structured known environment, and a structured unknown environment with moving objects. Overall results have shown that all participants were able to successfully operate the device, showing a high level of robustness of both the BCI system and the navigation system.

Resumo

Esta tese apresenta um sistema de navegação assistida (ANS) para uma cadeira de rodas robotizada (RW) que se suporta numa interface homem-máquina (HMI) que providencia comandos esparsos e discretos, mais precisamente um interface cérebro-computador (BCI) baseado em P300, ou, em alternativa, um sistema interruptor/ multi-interruptores com uma interface de varrimento que permite a seleção de comandos para navegar uma RW. A ANS proposta destina-se a pessoas com deficiências motoras graves que não são capazes de operar a maioria das HMIs comerciais.

A ANS é baseada numa abordagem de controlo colaborativo de duas camadas que considera as informações provenientes do homem e da máquina. O controlador colaborativo proposto está em conformidade com quatro requisitos de conceção essenciais: diálogo, consciência, auto-suficiência, e capacidade de adaptação. A escolha da BCI baseada em P300, a qual nos oferece um canal de comunicação que é completamente independente de qualquer exigência de controlo motor, apresenta-se como uma boa escolha para utilizadores com deficiências motoras graves. No entanto, devido às baixas taxas de transferência do sistema BCI baseado em P300, o utilizador só é capaz de emitir comandos esparsos e discretos ao longo do tempo. Neste sentido, para efetivamente usar tal HMI, propõe-se um ANS capaz de prever e executar as intenções do utilizador, tendo por base informações mínimas. Esta questão está diretamente relacionada com dois requisitos de conceção: consciência e auto-suficiência, o que significa que a RW deve ser capaz de identificar claramente as situações onde a ajuda é necessária, e pedi-la nesses casos. Além disso, uma vez que os utilizadores só são capazes de emitir informaçõs esparsas ao longo do tempo, o robô deve ser auto-suficiente e capaz de lidar com mudanças dinâmicas no ambiente, sem necessidade de qualquer ajuda, ou caso esta seja necessária, ser capaz de lidar com informações não confiáveis e com atrasos.

O ANS inclui um sistema de localização, um módulo de detecção de obstáculos, e um planeador local, que foram projetados para lidar com as mudanças do ambiente, e providenciam um certo grau de auto-suficiência à RW. Nós propomos uma estratégia de planeamento que é capaz de lidar com ambientes semi-estruturados pouco congestionados. A abordagem de planeamento é composta por um planeador local, que fornece um caminho global para atingir uma meta pré-definida, e num planeador local que intervém sempre que se detetam mudanças no ambiente. Como planeador local propusemos um VFH modificado que é realizado em três etapas: direção, planeador de caminho e fusão. Ao contrário das abordagens propostas por [Borenstein 1991, Ulrich 2000], a nossa abordagem constrói um histograma polar diretamente do varrimento do laser. Adicionalmente, a abordagem proposta também é capaz de fundir caminhos globais e locais.

Como abordagem de localização propõe-se um algoritmo de Markov que se baseia na fusão da odometria com os dados provenientes de um novo algoritmo de correspondência de varrimentos em coordenadas polares. Este último é realizado em três fases: pré-processamento, varrimento virtual e correspondência. O algoritmo de correspondência proposto utiliza o coeficiente de correlação de Pearson amostrado para avaliar as semelhanças entre os varrimentos atuais e virtuais. O factor de correlação é determinado no espaço de coordenadas polares, conduzindo a uma redução na complexidade computacional. Propõe-se ainda um filtro de Kalman extendido (EKF) para fundir os dados de odometria com os dados de marcadores magnéticos, os quais são detetados com um algoritmo 3D. Este último é baseado no método de ajuste dos mínimos quadrados (LSF) dos dados de medição, com o modelo 3D do campo magnético vertical.

O controlador colaborativo é adaptável à capacidade do utilizador em conduzir a RW. Para realizar a caracterização do utilizador propomos uma plataforma de formação em navegação assistida (ANTF), que é capaz de caracterizar os utilizadores, e classificá-los em três níveis de navegação diferentes: principiante, médio e avançado. A ANTF também é destinada ao treino de navegação da RW de utilizadores com deficiência motora.

Os resultados experimentais com a RobChair, a plataforma RW desenvolvida no ISR-UC [Pires 2002, Lopes 2007] são apresentados, demonstrando a eficácia das metodologias propostas. O protótipo foi validado com 10 participantes sem deficiência, e um participante com deficiência motora grave, em dois cenários distintos: um ambiente estruturado conhecido, e um ambiente estruturado desconhecido com obstáculos em movimento. O resultado global mostra que todos os participantes foram capazes de operar com êxito o sistema, mostrando um elevado nível de robustez do sistema BCI e do sistema de navegação.

Contents

List of Tables

List of Figures

List of Abbreviations

ALS	Amyotrophic lateral sclerosis
ANS	Assistive navigation system
ANTF	Assistive navigation training framework
APCC	Associação de paralisia cerebral de Coimbra
BCI	Brain-computer interface
BLM	Brunswik's lens model
CAN	Controller area network
CFD	Cross-fitting detection
COG	Centre of gravity method
CP	Cerebral palsy
DPF	Derivative peak finder
DWA	Dynamic window approach
EEG	Electroencephalography
EKF	Extended Kalman filter
ERP	Event related potential
FAB-MAP	Fast appearance based mapping
FSC	Fuzzy shared controller
GBPC	Genetic-based policy capturing
GML	Grid Markov localization
GPS	Global positioning system
HA/UA	Human agent/User agent
HMI	Human-machine interface

HRC	Human-robot communication
HRI	Human-robot interface
HRS	Human-robot system
IML	Intent-matching layer
ISI	Inter-symbol interval
ISR-UC	Institute for systems and robotics - University of Coimbra
LAN	Local area network
LFD	Longitudinal-fitting detection
LRF	Laser range finder
LSF	Least squares fitting
MA	Machine agent
MCL	Monte Carlo localization
MHT	Multi-hypothesis tracking
MSR	Magnetic sensing ruler
MWL	Mental workload
PC	Personal computer
PDA	Personal digital assistant
PDF	Probability Distribution Function
PRM	Probabilistic road maps
PSM	Polar scan matching
RBL	Rule-based lens
RPP	Randomized path planner
RTT	Rapidly-exploring random trees

RW	Robotic wheelchair
SLAM	Simultaneous localization and mapping
SNR	Signal-to-noise ratio
SOA	Stimulus onset asynchrony
SPM	Symbols per minute
TCP/IP	Transmission control protocol/Internet protocol
TT	Trial time
TT-CAN	Time-triggered CAN
USB	Universal serial bus
VCL	Virtual-constraint layer
VFH	Vector field hystogram
VSLAM	Visual SLAM
WMR	Wheeled mobile robot

Chapter 1

Introduction

Contents

"You cannot hope to build a better world without improving the individuals. To that end each of us must work for his own improvement, and at the same time share a general responsibility for all humanity, our particular duty being to aid those to whom we think we can be most useful."
Marie Curie, in "Pierre Curie"

1.1 Motivation

Mobility is essential to human-beings.

Disability is part of the human condition. Almost everyone will be temporarily or permanently impaired at some point in life, and those who survive to old age will experience increasing difficulties in functioning [WRD 2011].

Limitations related to mobility can become critical to participate in nearly all activities of daily living [Noreau 2000, Chaves 2004]. Quality of life and perception of life satisfaction have also been shown to be affected [Leduc 2000]. The average life expectancy has had a significant increase in recent decades, leading to a significant aging of the population in developed countries. According to Census 2011, 19% of the Portuguese population was 65 years older in 2011, and this value is expected to increase in the near future. This trend may cause an increased need for mobility assistance, especially if the average increase in life expectancy of the population does not imply a weakening in the quality of life standards of older people. According to the world disability report [WRD 2011, Casanova 2008], older people are unevenly represented in disability populations. This means that rates of disabilities are much higher among those who are older, and this phenomenon is common to low and high income countries. According to Census 2001 [Cen 2002], 6.2% of the Portuguese population suffers from one or more types of disabilities. Around 1.5% of the Portuguese population suffers from motor disabilities and 0.1% suffers from cerebral palsy disorders. Moreover, according to the Observatory of Inequalities [of Inequalities 2010], based on data provided by Eurostat, unemployment and inactivity rates are much higher among those who are disabled.

The world report on disability [WRD 2011] makes several recommendations to assist stakeholders in overcoming the barriers that people with disabilities experience. One recommendation is related to strengthen and support research on disability, to life standards improvement of people with disabilities and to overcome barriers that prevent their further participation in society. Human-centered robots, and assistive robotics in particular, may contribute to help motor-impaired people to reach a better level of mobility, towards an improvement of their life standards. Furthermore, increasing the mobility levels of people with motor disabilities can also ultimately contribute to improve their social inclusion.

Human-centered robots will play a decisive role in the near future. These systems are Human-centered encompassed in a wide range of devices, from intelligent transportation systems, entertaining robots, rehabilitation or intelligent assisted robots. Most of these systems are designed to improve the performance of, or assist, human users in different tasks. However, the unstructured, unpredictable, hostile, and dynamically varying environments make it imperative, in some cases, to make use of distinctive human decision-making capabilities. Moreover, as all these technologies are meant to be "human centered", the human user is expected to intervene, to a varying extent, to operate such systems. A control framework is therefore needed to facilitate the collaboration between a human user (also designated as the human agent) and an intelligent machine (or a machine agent) taking advantage of the unique strengths of the human and the machine, and aiding each other in those areas of weakness [Tahboub 2001].

1.2 Challenges on assisted navigation

Assisted navigation belongs to a type of semi-autonomous control architecture that can be applied to certain mobile human-centered robots, such as Robotic Wheelchairs (RW), as the one proposed in this thesis. It requires at least two agents, a Human Agent (HA), also designated as the User Agent (UA), and a Machine Agent (MA), sharing the control of a robot, this meaning that the response of the system is always influenced by at least two agents [Enes 2010].

The proposed assisted robot, in this case a RW, is meant to be used by people with severe motor limitations who are only able to provide sparse information to steer the RW. Due to this fact, one must be careful in choosing of the most appropriate Human-Machine Interface (HMI), since it must fit the motor capabilities of each user. On the other hand, the Assistive Navigation System (ANS) must be able to clearly identify and respond to both user requests and commands provided by the HMI in the most accurate and safe manner. Another challenging issue related to assisted navigation and with the semi-autonomous controller in particular is how it can be made adaptive to user agents with different capabilities, while interacting with the RW. To increase the ANS effectiveness it is of major importance that users provide the system with reliable commands, and therefore it is imperative to train users to use the HMI to steer the RW in an appropriate manner. Moreover, a user characterization that is later used to adapt the navigation system to the user ability to steer the RW is also crucial to tailor the ANS to the steering capabilities of each user.

Safety and robustness are key issues when dealing with robots cooperating with humans. Safe navigation in more challenging scenarios is another fundamental issue of assisted navigation, where it is required to deal with new static and moving objects alike, being essential to perceive and interact with other humans.

1.3 Main goals and contributions

The main goal of this thesis is to propose an assistive navigation system based on collaborative control between a user agent with reduced motor capabilities, and a machine agent. This ANS was designed for people with severe motor disorders such as Amyotrophic Lateral Sclerosis (ALS), Cerebral Palsy (CP) and spinal cord injuries. In the case of neuro-degenerative motor disorders, the level of functionality depends on the stage of the disease. In non-degenerative motor disorders, such as CP, the symptoms vary sig-

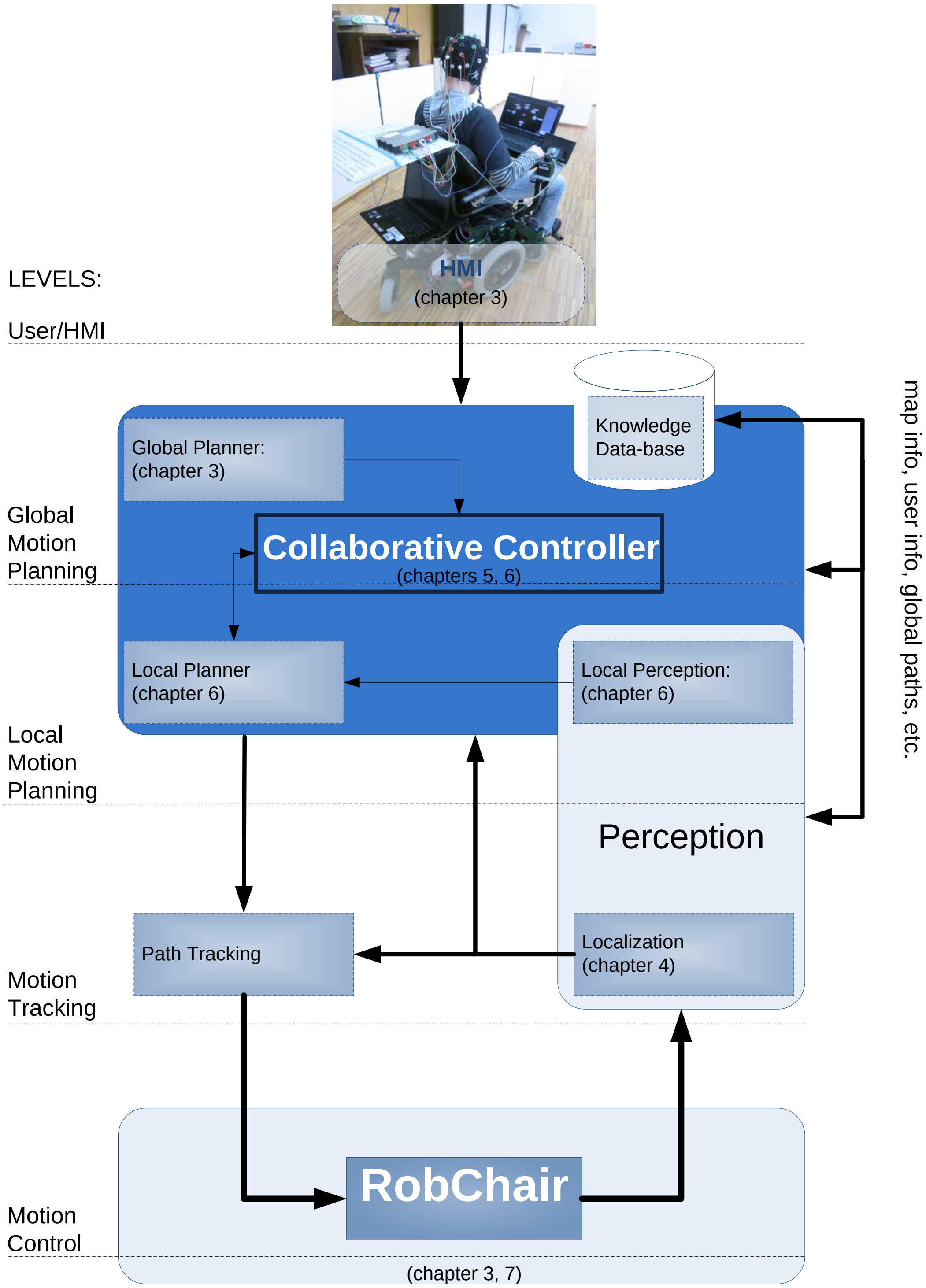


Figure 1.1: Thesis map with reference to chapters including main contributions.

nificantly among patients, and can go from a simple difficulty to walk to a total lack of control of muscular activity. These users are unable or have great difficulty to control conventional interfaces capable of providing a continuous reliable command over time. Due to this, users are only able to provide sparse commands over time. To overcome this problem, the RW must rely on the navigation system to validate user commands effectively. Additionally, and to ease user effort, low-level commands are only issued when there are dynamic changes of the environment, or when ambiguous situations occur. The development and implementation of an ANS adaptable to user's wheelchair-steering competence is also an important goal of this thesis. The following main contributions were given in this thesis:

- Complete assistive navigation architecture: the proposed assistive navigation architecture is structured in five main levels, namely: HMI, global planning, local planning, motion tracking, and motion control (see Fig. 1.1). It also includes a perception module, in charge of maintaining the local environment model, and of determining the correct pose of the robot. The "heart" of the system is the collaborative controller responsible for providing the final steering command, which was determined based on the steering command given by the user and the candidate steering commands provided by the local planner, taking context into consideration. The ANS also includes a knowledge database that stores the information regarding the working environment, and situation-based restrictions.

- Global and local planning: In our approach, we use the A* algorithm [Hart 1968], [Pearl 1982] as a global path planner to find the least-cost path from a given initial node to a goal node. An a priori grid map is provided to the global planner. The global path is determined online when the user selects a certain goal node he/she wishes to reach. In this approach, velocity and acceleration constraints are not considered, and the main goal is to find the least-cost path to a final node. We also propose to use the work developed in [Solea 2009] to smooth the path between an initial and a final node. For this approach, we provide a topological map that contains all possible subgoal nodes in the free space. All possible paths between nodes are determined offline and stored. A local planner based on the Vector Field Histogram (VFH) approach, determined by information provided by a Laser Range Finder (LRF), is also proposed. Unlike the approaches proposed by [Borenstein 1991, Ulrich 2000], our approach builds a polar histogram directly from laser scan information and does not require the construction of the Cartesian local map. Additionally, an algorithm to blend global and local paths is also proposed.

- Localization based on LRF: localization is performed using dead-reckoning data (odometry) for rough positioning, and laser data for polar scan matching. A Markov localization system was designed to fuse odometry with a new LRF Polar Scan Matching (PSM) algorithm, which follows three main stages: scan preprocessing, scan prediction, and matching. The proposed matching algorithm uses the sample Pearson correlation coefficient to evaluate the similarities between current and predicted scans. The correlation factor is determined on polar coordinate space, leading to decreased computational complexity of this algorithm, when compared with other 2D-localization algorithms.

- Localization based on the detection of magnetic markers: localization is performed using dead-reckoning data for rough positioning, and absolute positioning data provided by magnetic markers. The Extended Kalman Filter (EKF) was chosen for the data fusion process. The RW is equipped with a magnetic sensing ruler developed at ISR-UC that is able to perform a robust detection of magnetic markers. The detection is based on a 3-D algorithm that includes Longitudinal-Fitting Detection (LFD), and Cross-Fitting Detection (CFD). Both, the LFD and the CFD are based on the Least Squares Fitting (LSF) of the measurement data with the 3D model of the vertical magnetic field.

- Novel collaborative control algorithm: to decrease user effort, low-level commands are only issued when there are dynamic changes of the environment or ambiguous situations occur. A two-layer collaborative control approach is thus proposed to obtain an effective navigation of the RW, by receiving user commands that are sparsely issued. The collaborative controller includes a Virtual-Constraint Layer (VCL) and an Intent-Matching Layer (IML). The former is a traded controller in charge of enabling/disabling user commands, as a function of certain criteria, and the latter is a shared controller that determines the suitable maneuvers, taking into account the user's steering skills, and context restrictions. User characterization was incorporated in the collaborative controller, and therefore the navigation system is customized to the user's skills in steering the RW, increasing the overall navigation task efficiency.

- Assistive Navigation Training Framework (ANTF): this training framework was developed with two goals in mind. First, to train user ability to steer a powered wheelchair appropriately, given the restrictions of their limited motor capabilities, and secondly to characterize the user's steering of the RW, and user's skills in operating the HMI device. We propose to sort users in three steering levels: beginner, average, and advanced. User steering classification is performed using the Rule-Based Lens (RBL) paradigm [Yin 2006], for which three different user models were proposed, according to their steering skills, using the Genetic Based Policy Capture (GBPC) algorithm [Rothrock 2003].

- Assessment of the assistive navigation using P300-based Brain Computer Interface (BCI), with both able-bodied and disabled individuals: to the best of our knowledge, this thesis is the only work to date where a complete ANS for a robotic wheelchair, using a BCI for system interface, was tested by an individual suffering from cerebral palsy. The system was also tested with a significant group of able-bodied users. A complete assessment of the ANS was carried out, and new user metrics were proposed.

- The algorithms presented throughout this thesis have been tested in player/stage [Gerkey 2003] simulation environment, and in the real platform, RobChair [Lopes 2011, Lopes 2012a, Lopes 2012b].

1.4 Thesis outline

Figure 1.1 shows the thesis map, chapters comprising the main contributions are highlighted. The remainder of this thesis is organized as follows.

Chapter 2. This chapter provides ANS background for basic understanding of the work presented in this thesis. Additionally, a state-of-the-art allows the reader to frame the contributions of this thesis.

Chapter 3. A description of the complete assistive navigation architecture is presented, including details on the information flow. This chapter presents technical and scientific aspects related to the development and assessment of the ANS applied to a robotic wheelchair. It gives the necessary background to understand the methodologies used in the remaining chapters, including an overview of human-machine interfaces.

Chapter 4. The localization method used throughout the thesis is presented in this chapter. A 2D Markov localization approach is described here. Results from both simulation and real experiments with the RobChair are also presented in this chapter.

Chapter 5. This chapter describes the ANTF platform, which was developed to train and characterize user wheelchair-steering performance.

Chapter 6. This is the core chapter. It formally presents the proposed two-layer collaborative control algorithm. Methods and algorithms for ANS planning are also described here. The local planner is based on a modified vector field histogram approach determined by information provided by a laser scan.

Chapter 7. This chapter presents experimental results with RobChair using P300-based BCI to steer the RW, in two different navigation scenarios. The tests with the real platform were carried out with a significant group of able-bodied individuals and one disabled individual suffering from cerebral palsy disorders.

Chapter 8. This chapter draws some conclusions and explores some future work.

Appendice A. This appendice presents an extended Kalman filter that is used to fuse odometry with data from magnetic markers.

Chapter 2

State-of-the-art on semi-autonomous control for assistive robotics

This chapter introduces the concept of assistive robotics based on semi-autonomous control. It starts by presenting the most important concepts related to assistive robotics. The main strands on semi-autonomous control are then described, highlighting the methods and current potential applications, giving special focus to robotic wheelchairs, providing the current state-of-the-art.

2.1 Introductory concepts

2.1.1 Assistive robotics

Assistive robotics has largely been referring to robots that assist people with motor disabilities through physical interaction. This definition is nevertheless outdated because, currently assisted robots are used in a much broader scope, including assistance through non-contact interaction, such as those that interact with convalescent patients in a hospital or with senior citizens in nursing homes [Feil-Seifer 2005]. An assistive robot is a Human-Robot System (HRS) that obeys to a type of semi-autonomous control scheme, where both human and machine agents are able to influence the control of the system. Research in the fields of assistive robotics includes robotic wheelchairs [Levine 1999, Pires 2002, Taha 2008, A. Huntemann 2008], minimally invasive surgery [Nudehi 2003], surveillance, search and rescue [J. Shen 2004, Gao 2011], intelligent mobility assistants [McLachlan 2005, Law 2002, Aigner 1997, Aigner 1999], companion and rehabilitation robots [K. Wada 2002, Wade 2011, Ghosh 2011, Tani 2011, Gross 2011], manipulator arms for motor impaired humans[Kawamura 1995, Giminez 2003], or educational robots [Ghosh 2011]. The next sections give some fundamental background regarding semi-autonomous control and HRS.

2.1.2 Definition of semi-autonomous control

Semi-autonomous control can be defined as a type of control scheme that causes the output or response of a system to be influenced by two or more agents, as opposed to fully autonomous systems, where control belongs solely to the robot [Sheridan 1992, Ong 2005, Enes 2010].

Semi-autonomous controllers are used in assistive platforms, such as robotic wheelchairs, to aid human users with every-day-life tasks. In the case of assisted navigation, offering aid to physically impaired human users requires robotic systems to decide adequately when and to which degree corrective actions should be executed. This is a fundamental aspect in semi-autonomous control, since it can also be seen as a set of situations where control over a system is shared, traded or a combination of both among one or more humans, and among one or more robotic systems.

2.1.2.1 Shared control, and traded control

Shared and traded controllers can be classified as the primary forms of semi-autonomous control. At this point it might be useful to first provide a working definition of sharing and trading as a basis for further discussion. According to the Webster's online dictionary [Web 2012]:

Sharing: Using something jointly with others.

Trading: To exchange or give something in exchange.

In the case of sharing, human and robot influence in the execution of a certain task are not mutually exclusive. In an assistive robotics context *jointly* it means that both human and the robot cooperate together through the use of something to ensure task performance. In the case of trading, human and robot influence in the execution of a particular task are mutually exclusive. *to exchange* meaning that human and robot both give and receive something while working together [Ong 2005, Enes 2010]. Figure 2.1 illustrates the notions of sharing and trading between a human and a computer. According to Fig. 2.1, the roles of the computer (or the machine) can be classified according to how much task-load is carried compared to what the human operator alone can carry. In the case of sharing, the computer can extend the human's capabilities beyond what he/she can achieve by him/herself, or it can partially relieve the human, making his/her job easier. In the case of trading, the computer can backup the operator if he/she falters, or it can replace him/her completely [Sheridan 1992].

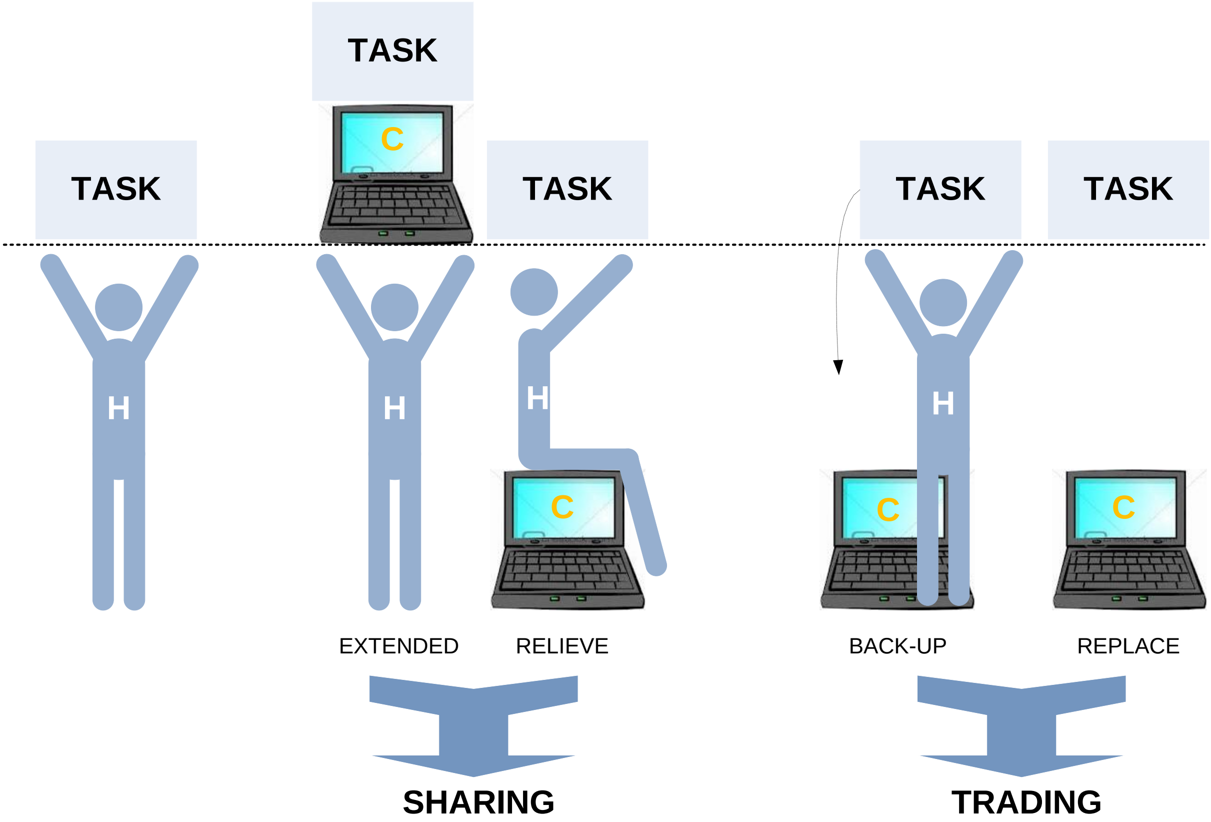


Figure 2.1: Notions of sharing and trading between a human and a computer. H stands for human and C for computer (adapted from [Sheridan 1992]).

The basic questions in sharing and trading are as follows [Sheridan 1992]. In sharing - ”which tasks should be assigned to human and which to the robot?” In trading - ”which aspects of the tasks to trade, and when should control be handed over, and when should it resume control during task execution?”

2.1.3 Human-robot systems

Current HRS take many different forms. These can range from manually controlled systems, such as teleoperation, to autonomous robotic systems that employ artificial intelligence, machine perception, and advanced control [Giralt 1997]. Table 2.1 illustrates a brief description of the six main types of HRS, presented in increasing order of autonomy.

2.1.3.1 Requirements for a HRS

The design of a HRS should comply with three main requirements:

Human roles: establishes how the robot is commanded and controlled in an HRS from

Table 2.1: Basic description of the six main types of human-robot systems.

HRS	Basic description	Applications
Type 1: Teleoperation system (not computer-aided)	The human is located remotely from the robot. The robot is directly controlled by human supervisor's own visual senses (line of sight). The robot extends the human's manipulation capability to a remote location.	Underwater cleaning of reactor vessels, pipe inspection, etc.[Roman 1993]
Type 2: Teleoperation System (computer-aided)	An extension of Type 1, but the human controls the robot through artificial sensing, computer, and displays. The robot extends both the human sensing and manipulation capabilities.	Robotics Surgery [Grimbergen 2004, Nudehi 2003], underwater operation [Lee 2006], etc.
Type 3: Telerobotics System (an advance form of teleoperation system)	The robot is normally equipped with high level of intelligence (such as safe navigation, path planning, etc.) while receiving higher-level instructions from the human instead of exercising continuous manual control as in Type 1 and 2.	Space exploration [Pedersen 2003], pipe inspection [Roh 2005], search and rescue [Casper 2003, J. Shen 2004], etc.
Type 4: Intelligent Mobility System	A variant of Type 3, but the human and the robot are close to each other.	Intelligent wheelchair [A. Huntemann 2008, Grasse 2010, Lopes 2011], or mobility support system [McLachlan 2005]
Type 5: Work Partner	Robot is equipped with powerful and versatile mechanisms to communicate, interact and cooperate with human in a natural and intuitive way.	Caretakers [Gross 2011, Haegele 2001], educational robots [Ghosh 2011],etc.
Type 6: Autonomous Robot	Robot replaces the human and performs the desired tasks autonomously.	iRobot Roomba Intelligent vacuum cleaner, Minerva: museum tour guide [Thrun 1999].

the perspective of the human interacting with the robot [Sheridan 1992, Roh 2005, Burke 2004].

Robot roles (robot autonomy): establish the required degree of robot autonomy in an HRS from the perspective of the robot interacting with the human. This consideration is directly linked to the degree of human intervention (i.e. degree of control) required for the robot to perform a task [Sheridan 1992, Roh 2005, Burke 2004].

Human-robot communication: establishes how human and robot communicate, being directly related with the choice of HMI [T. Fong 2001, Kosuge 2004].

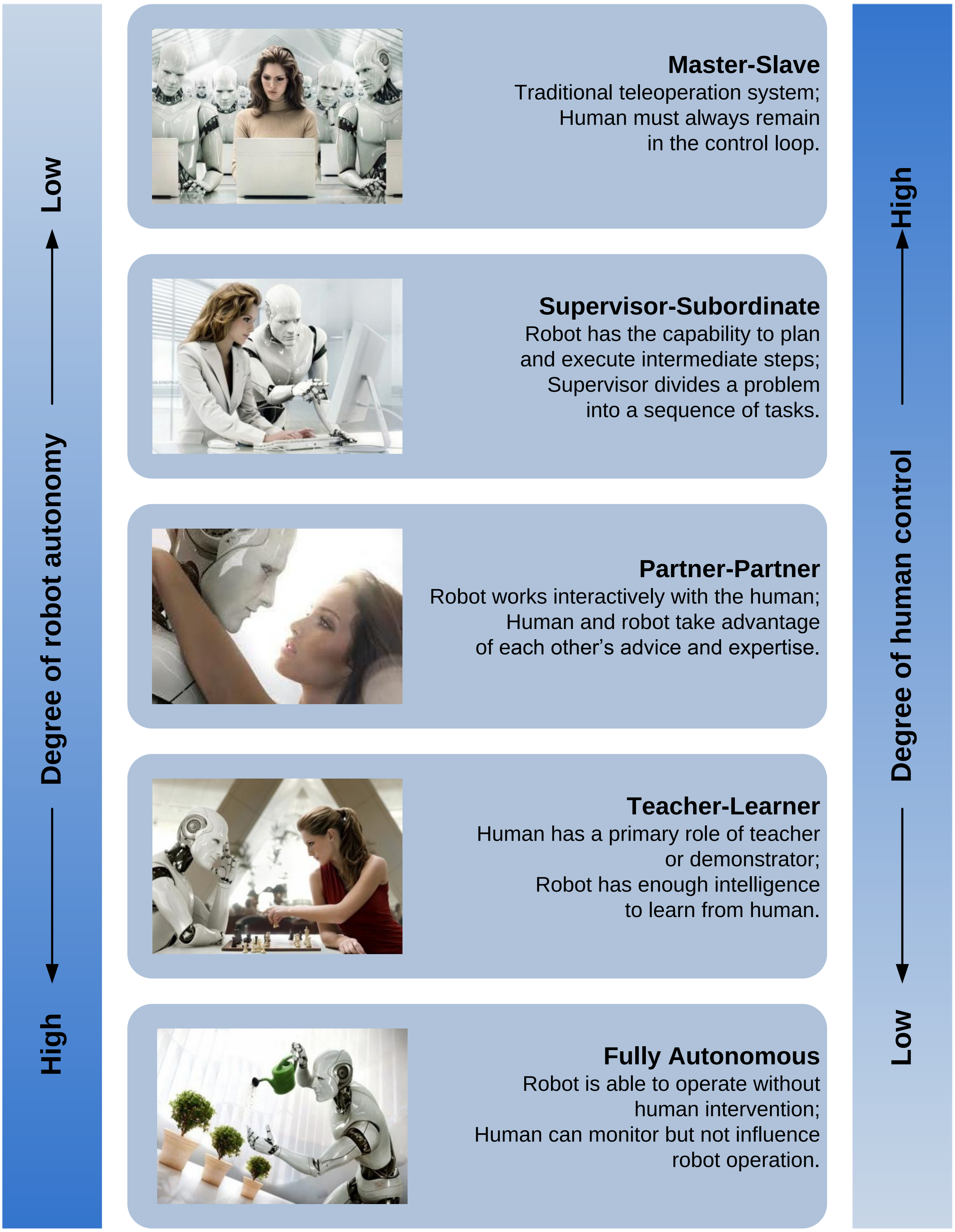
2.1.3.2 Human roles

The roles of a human in an HRS are application-specific ([Sheridan 1992, Burke 2004]). Many questions can be raised regarding the roles of the human in an HRS. Should he/she be an active, serial element in the control loop? Should he/she be a supervisor monitoring the progress of the system [Curry 1976]? Is the human a necessary system element in the control loop? Which effective control method should be used to determine how the human and robot interact in order to increase the system performance? The roles and relationships of human and robot in the different types of HRS depicted in Table 2.1 are classified as illustrated in Fig. 2.2:

Master-slave (type 1 and 2): This describes the traditional teleoperation system [Sheridan 1992]. The master-slave is the most basic form of operation, where the human must always remain continuously in the control loop. The operating principle is simple; the human (master) has full control over the robot (slave), e.g. all control decisions will depend on the human. When human stops, control stops.

Supervisor-subordinate (type 3 and 4): Here, the robot does not simply mimic the human's movements as in the master-slave role. Instead, the worker robot has the ability to plan and execute all the necessary intermediate steps, taking into account all events and situations with minimum human intervention. On the other hand, the human as a supervisor divides a problem into a sequence of tasks, which the robot performs on its own [Sheridan 1992]. If a problem occurs, the human supervisor is responsible for finding a solution and devising a new task plan.

Partner-partner (type 3-5): Here, the robot is viewed as the human's work partner and is able to work interactively with the human. Both human and robot are able



Master-Slave
Traditional teleoperation system;
Human must always remain
in the control loop.

Supervisor-Subordinate
Robot has the capability to plan
and execute intermediate steps;
Supervisor divides a problem
into a sequence of tasks.

Partner-Partner
Robot works interactively with the human;
Human and robot take advantage
of each other's advice and expertise.

Teacher-Learner
Human has a primary role of teacher
or demonstrator;
Robot has enough intelligence
to learn from human.

Fully Autonomous
Robot is able to operate without
human intervention;
Human can monitor but not influence
robot operation.

Figure 2.2: Roles and relationships between a human and a robot. Photographs were obtained in [Steiner 2012].

to take advantage of each other's skills and to benefit from each other's advice and expertise [Grasse 2010, Wasson 2001]. Unlike the supervisor-subordinate-type system, if a problem occurs, the robot may provide the necessary assistance to find a solution [T. Fong 2001].

Teacher-learner (type 3-6): The human acts a primary role of teacher or demonstrator. For this type of system, it is assumed that the learning robot has sufficient intelligence to learn from the user [Nicolescu 2001]. Once the robot is able to handle the task, it can replace the human completely or work together with the human depending on the context of the application.

Fully autonomous (type 6): Here, the aim is to develop robotic systems with the ability to operate without any human intervention once the control is delegated to the robot [Giralt 1997, Thrun 1999]. This implies that the human can only monitor but not influence the robot operation. The only intervention possible is to stop the robot operation when a potentially serious error occurs.

2.1.3.3 Robot roles - robot autonomy

Robot autonomy encompasses two basic attributes [Giralt 1997]: operating autonomy and decisional autonomy. Operating autonomy refers to the basic operational capability (i.e. the technological considerations) of a physical robot. For instance, to be "operational", a mobile robot must be equipped with the following basic components: adequate navigation sensors (e.g. sensors for obstacle avoidance, detection, and localization), appropriate communication links to interface with the human, embedded computation and program storage for local control systems. Decisional autonomy refers to the level of intelligence imbued in a robot. This includes an internal representation of the world and of the task, and the abilities to act reasonably in an unstructured/semi-structured environment. This encompasses the ability to reason about its own action, learn, and adapt to some extent based on human feedback or from its own environment, over a given period of time [Roh 2005].

To respond to a range of different control modes and to turn control mode transitions easier, the robot must have the required autonomy to interact with the human [Braynov 2003]. By stating that a robot is autonomous, it does not mean that the robot is thoroughly self-governing and capable of completing self-planning and self-control. However, it can operate with some known (to the human) level of capability in the absence of human supervision/management for a defined period of time [Roh 2005].

2.1.3.4 Human-robot communication

To ensure that the robot responds to the correct control mode when varying its own degree of autonomy, some aspects belonging to Human-Robot Communication (HRC) have to be addressed. In Human-Human communication, humans communicate with each other easily through the same language. They can communicate effectively face-to-face or through electronic communication devices. However, in the case of HRC, it is not that straight forward, because humans cannot communicate with robots directly. A well-defined communication channel is therefore required to address the different modes of interactions between the human and the robot. Some of the basic considerations in HRC are [Roh 2005]:

Methods of communication: This relates to how information is transferred from the human to the robot (or vice versa). The current state of HRC encompasses a large spectrum of methods, such as Personal Computer (PC)-based control interfaces, Personal Digital Assistant (PDA) as interface devices [Fong 2001b], haptic interface [Ellis 1996], voice [Pires 2002], gesture [Hu 2010], eye-trackers [C. Lin 2006], BCI [Pires 2011b, Lopes 2011], etc. The use of these methods is problem-specific or application-specific. However, regardless of the method being used, effective human-robot communication is paramount.

Communication format: This pertains to the communication language used for information exchanging between the human and the robot. In [Zhai 1992], the notion of "continuous" and "discrete" languages as two different coding mechanisms to describe human-robot information exchange, is proposed. According to [Zhai 1992], continuous language is used to represent information that is distributed continuously in quantitative or qualitative form, either along a spatial or a temporal dimension. Examples include sending of continuous signal (e.g. via input devices such as a joystick) to control the robot. On the other hand, discrete language is used to represent information which consists of separate or distinct elements. Examples of discrete language are signs, symbols, written text, etc. used for communicating with the robot. As compared to continuous language, discrete language is normally used when the available information bandwidth is low or the communication delay is high. As stated before, the choice of the method for communication exchange is problem-specific, and the choice of a discrete interface is sometimes related to the capabilities of the human (e.g. a motor impaired person may not be able to control a continuous interface). This also implies that the robot must have enough autonomy to perform the tasks, since the commands provided by the human are sparse and discrete.

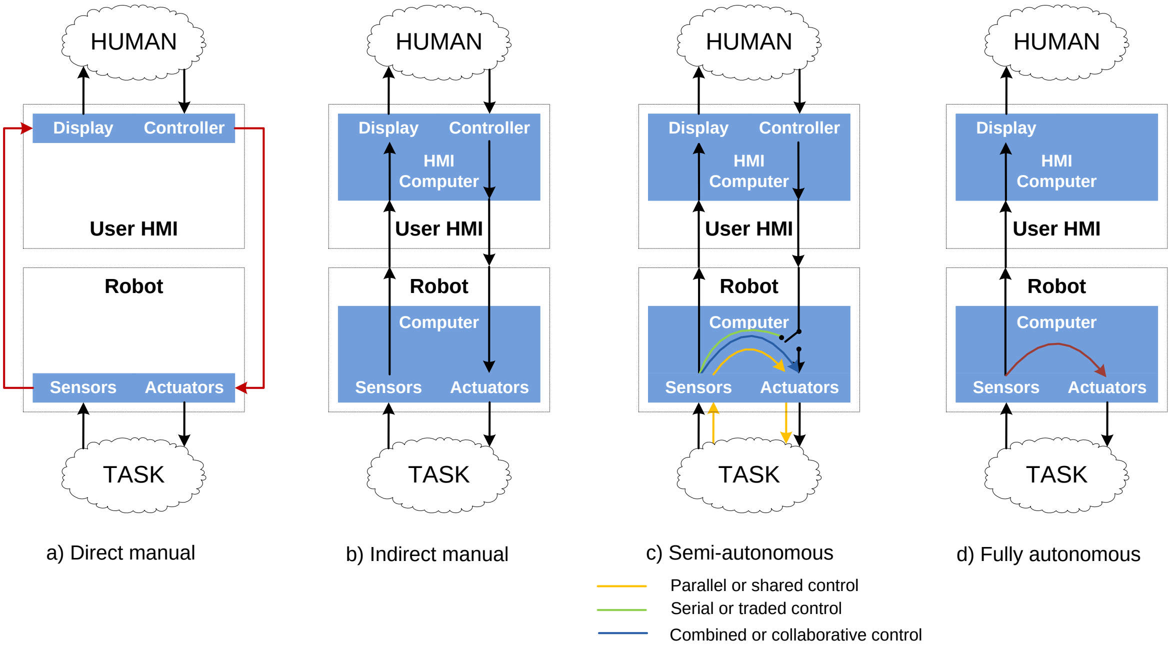
Communication bandwidth: This relates to the amount of HRC required to perform a given task. A good communication system is two-way (full duplex) with high data rate so that command data can be transferred from the human to the robot, and at the same time information from the robot can be conveyed back to the human. The amount of communication can be assessed by the information quantity, measured in bits, and the information transfer bandwidth, measured in bits per second. For instance, high communication bandwidth is normally required in manual control (such as teleoperation), because the human must control each movable function of the robot in real time. On the other hand, lower communication bandwidth is required in semi-autonomous control because continuous control of the robot is not required.

Purpose of communication: This pertains to the type and the purpose of the information being exchanged between human and robot. In performing a task, the human must provide the robot accurate information about the task at hand. Likewise, the robot should communicate to the human any information regarding its state and provide a feedback of the current status of the task, allowing him/her to evaluate the task's successes and faults. In addition, it is important that the robot conveys any difficulties encountered while executing the task.

2.2 Semi-autonomous control

The concept of semi-autonomous control comes from joining two main strands: the teleoperation concept [Sheridan 1992] and the autonomous robot concept [Giralt 1997]. According to [Giralt 1997], in the teleoperation concept, both human and machine interact at the human operator station level. On the other hand, in the autonomous robot concept, the focus is to have on-board, in-built intelligence at machine level so that the robot can adapt its actions autonomously to the task conditions during Human-Robot Interaction (HRI).

Although the semi-autonomous control concept may emerge from the two mentioned concepts, the basic objective remains the same. That is, in order to go further and beyond simple human control of a robot there is a need to provide the robot basic competence and a certain degree of autonomy. This leads to a reduction in the degree of human supervision [Sheridan 1992, Giralt 1997]. Based on the roles and relationships defined in subsection 2.1.3.2, Fig. 2.3 illustrates the nature of the task-performance interactions between human and robot under different control modes.



a) Direct manual

b) Indirect manual

c) Semi-autonomous

d) Fully autonomous

Figure 2.3: A spectrum of control modes (adapted from [Sheridan 1992, Ong 2005]).

As the human cannot perform the task directly, he/she must perform the task via two main interaction loops. One loop defines the interaction between human and robot via an interface. The second loop defines the interaction between the robot and the task via its sensors and actuators. The ”intermediary“ facilitating the interaction between these two loops is the control mode.

Here, each control mode is viewed as a way for the human to interact with the robot task performance. Fig. 2.3 (a) represents traditional master-slave manual control system (Type 1). Fig. 2.3 (b) represents indirect (i.e. computer-aided) master-slave manual control system (Type 2). Fig. 2.3 (c) represents semi-autonomous control system (Type 3-5). Fig. 2.3 (d) represents autonomous control for a fully autonomous robot (Type 6).

2.2.1 Shared control

Semi-autonomous control can be further classified into parallel type, serial type or a combination of both parallel and serial types [Yoerger 1987]. In the parallel type (Fig. 2.3 (c)), both manual and autonomous control loops coexist. The parallel type is normally referred to as shared control, an approach to combine the strengths of both the human and the robot by letting them control different aspects of the system simultaneously in situations that require teamwork [Arkin 1991, Papanikolopoulos 1992, Sheridan 1992, Brunner 1993, Lee 1993, Wasson 2001]. It is normally used in situations where the task is too difficult to be achieved by either the human (via manual control) or the robot (via autonomous control) alone.

In [Tahboub 1999] shared control is compared to a horse-and-driver cooperation. The driver excels in global planning and is responsible for high-level steering of the horse, whereas the horse excels in fine motion control and avoids obstacles. This way, the weaknesses of both cooperating agents are mitigated, and their strengths combined. Shared control is mutually not exclusive approach, allowing simultaneous intervention of the machine and the human operator. The execution of each task relies on both, user command signals and machine signals, since both have some degree of control over the system output. More specifically, sharing control means that the human and the machine control different aspects of the system at the same time [Sheridan 1992].

Shared control has been studied in different ways in both fields of telemanipulation and teleoperation of mobile robots. Examples include position-compliance control [Brunner 1993, Lee 1993], vision-based perceptual guidance control [Papanikolopoulos 1992, Hoppenot 2002], safeguarding control [T. Fong 2001, Wasson 2001] and behavioural control [Arkin 1991, Levine 1999, Grasse 2010]. Most shared control approaches have been based upon some kind of coordination/fusion strategy regarding human inputs and robot assessment of the environmental task. As compared to manual control, shared control frees the human's attention from directly controlling nominal activities, while allowing direct control during more perceptually intensive activities such as manipulation of parts (e.g. [Brunner 1993, Lee 1993]) and navigation in cluttered area (e.g. [Grasse 2010, Carlson 2011a, Carlson 2011b]).

2.2.2 Traded control

In traded control, also denoted as a serial type of semi-autonomous control (see Fig. 2.3 (c)), either manual control or machine control can be selected as the operating mode at any given time. Traded control is a mutually exclusive approach allowing the human and the machine to exchange control on some basis [Papanikolopoulos 1992, Sheridan 1992, Lee 1993]. Control is assigned to the human or to the machine based on the degree of task demand, environment constraints, or user and machine ability to execute the task. In this control approach, a machine can perform part of a task on its own, thus freeing the user to give more attention to the rest of the task, a process referred to as task partitioning. In performing a navigation task, the human may intervene and take control from the robot (e.g. to give a new movement direction) if it moves in the wrong direction. On the other hand, the robot may override undesired commands (e.g. decelerates or stops) coming from the human, if these commands may cause damage. From this perspective, this control mode may allow both human and robot to "assist" each other in a partner-partner like manner.

2.2.3 Collaborative control

In the combined configuration (Fig. 2.3 (c)), both serial and parallel types interact to an extent, where the tasks within each mode may also be shared and traded [Sheridan 1992]. A classical example is the sharing and trading of control in aircraft autopilot systems [Billings 1997]. During the cruise phase, in order to engage the autopilot system, the pilot trades the control over to the machine. While the autopilot system holds the altitude, the pilot may adjust the heading, thereby sharing control at the same time. An example of the combined type is the Supervisory Control (SC) based on the supervisor-subordinate role [Sheridan 1992].

Another recent example is the collaborative control (CC), an extension of SC based on the Partner-Partner model by [Fong 2001b] for the teleoperation of mobile robots. According to [Fong 2001b], the essential difference between CC and SC is that CC can adjust its method of operation based on situational needs so as to enable "fine-grained sharing and trading of control". More specifically, in a situation where the robot does not know what to do or is performing poorly, it has the option to hand over the control (e.g. in decision making) to the human. In other words, CC may enable work to be dynamically allocated to the robot or the human throughout a particular task performance. Situations where machines and humans cooperate to achieve a common goal fall within the field of collaborative control [Poncela 2009]. Figure 2.4 presents a summary of all control modes presented so far.

2.2.4 Adaptiveness

Semi-autonomous control provides a framework for integrating users with varied experience, skills, and training. As a consequence, the machine has to be able to adapt to different operators, meaning that the degree of autonomy of the machine is variable [Fong 2001a].

One possible approach for controlling the degree of adaptation is to delegate this

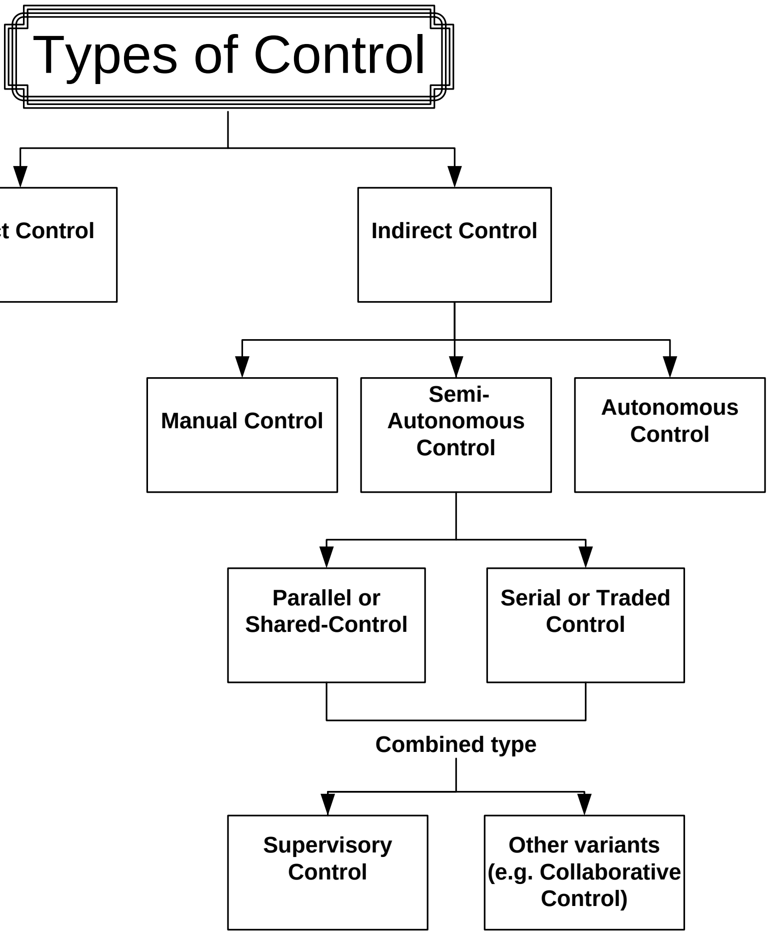


Figure 2.4: Classification of different types of control in a Human-Robot System.

authority to the human agent. Unfortunately, as the complexity of a system increases, the human might not always have enough available resources to perform adaptation. This suggests the need for automatic adaptation. Since semi-autonomous control allows for variable human response (in terms of accuracy, speed, etc.), automatic adaptation means that the machine agent should be able to dynamically adjust its degree of autonomy. In particular, the level of autonomy should adapt to fit situational needs and operator input (e.g., human decision making). If the machine is unable to cope with a particular situation and the human is able to help, the level of autonomy should be reduced. On the other hand, if the human is unable to respond, the level of autonomy needs to be increased [Fong 2001a].

Nevertheless, automatic adaptation introduces in the process a degree of uncertainty that must be weighted. An example of an automatic adaptation for a shared control system, called the NavChair Assistive Navigation System, can be found in [Simpson 1998, Levine 1999]. In [Carlson 2012] a collaborative control mechanism that assists users as and when they require help, was proposed. The system uses a multiple-hypothesis method to predict the driver's intentions and, if necessary, to adjust the control signals to achieve the desired goal safely.

2.2.4.1 User modeling

In a HRI context, a user model can be defined as a representation of the relevant attributes of a user. The relevant attributes can vary according to the type of task that is being executed. For instance, the user model for a navigation task must capture a set of attributes related to that task, such as the user's driving skills. The degree of adaptation in a semi-autonomous controller depends on the user model, and how the relevant attributes vary for each user.

The work of [Fong 2001a] conceptualizes user modeling and how it can be incorporated in a semi-autonomous controller. This is done through the use of predefined, stereotype user profiles. Each profile describes a class of users through a set of attributes. To create the user profiles, three design issues were taken into account, namely:

- Attributes: What information about a user is required and how should it be represented?
- Definition: What subgroups of the users (the stereotypes) are going to be defined? How do we obtain information about each type of user?
- Use: How will the system use information about different types of users (i.e., what useful behavioral adaptation can be made)?

In [Fong 2001a] three attributes were considered for the example of a vehicle teleoperation in unknown environments: response accuracy, which gives an estimate of how accurately the user can answer a question (or make a decision); expertise, which is concerned with an estimate of task skill / domain knowledge the user possesses; and query interval, which indicates how often the user is available to answer questions (or make decisions). Three user profiles (stereotypes) were also defined, namely: novice, scientist, and expert. Each profile was then used to configure the human-robot interaction in three ways: to fit the interface to user needs; to filter dialogue messages according to each type of user; and to modify how the robot acts by configuring robot modules.

Examples of relevant works on user modeling applied to semi-autonomous navigation of a robot are [Poncela 2009, Li 2011, Demeester 2008, A. Huntemann 2008]. The work of [Poncela 2009] incorporates an evaluation of both, human and robot, into the collaborative control approach to steer a robot. The efficiencies of human and robot are evaluated at each time instant, and their commands are combined, according to their efficiency, to provide the shared motion commands (rotational and translational velocities) to the wheels' controllers. The efficiencies of human and robot are continuously evaluated in terms of softness, trajectory length and security. In the work of [Li 2011], a dynamic shared control method is proposed to adapt wheelchair's assistance to the variations of user performance and the environmental changes. Three evaluation indexes, namely safety, comfort and obedience, are designed to evaluate wheelchair performance in real time. A minimax multi-objective optimization algorithm is adopted to determine the user's control weight. In [Demeester 2008, A. Huntemann 2008] a general user modeling technique was developed to assist people with a certain driving pattern caused by aged-related motor disorders. The last three approaches use a conventional joystick to steer the robot, and are mainly suited for users with some motor capabilities and able to provide a continuous reliable command.

Another important issue raised by user modeling is concerned to the evaluation of user's skills for each attribute defined in the user model representation. The standard lens paradigm based on multi-regression analysis [Brunswik 1955, Hammond 1975] is able to compare human judgments and the environmental information. The work of [T. Sawaragi 2003] uses this methodology to compare the performance of humans in steering a robot with the performance of an autonomous simulated robot. In [Rothrock 2003] a technique called GBPC is proposed to generate environment and human rule-based models using propositional logic.

2.2.4.2 Mental workload and user signal degradation

Mental Workload (MWL) and user signal degradation over time are also two important aspects for system adaptation in semi-autonomous control. The underlying assumption is that each person has a range of MWL in which performance is optimal and that adapting the system, to maintain the operator's MWL within his/her own range, maximizes the total human-machine system performance. This is also related to the fact that there is a limit to the number of tasks a human operator is able to perform simultaneously in a given time.

The work in [Crandall 2002] presents a theoretical characterization of the efficiency of human-robot interaction, with the purpose of designing a HRS with adjustable robot autonomy. To enable a human to manage multiple tasks (including interacting with multiple robots), it is necessary to know how long a human can give attention to one robot before the performance of other neglected tasks deteriorate. The proposed framework was built on the intuition that the likely performance of a human-robot interaction degrades as the human neglects the robot to perform other tasks and as world complexity increases. In the approach proposed by [Crandall 2002], two different robot control schemes (shared control teleoperation and traditional manual control teleoperation) were analyzed in terms of system performance and MWL demands on the human operator.

2.3 Robotic wheelchairs

In recent years different assistive navigation architectures based on semi-autonomous control, for intelligent wheelchair platforms, were developed, such as RobChair developed at the Institute for Systems and Robotics at University of Coimbra [Pires 2002, Lopes 2011, Lopes 2012a, Lopes 2012b]. Most of the intelligent wheelchairs described in the literature incorporate a semi-autonomous controller that belongs to one of three main groups of control approaches.

The first group leaves control mostly to the user, and automatic navigation is only triggered when a given situation is detected (e.g. imminent collision). These systems do not normally require a prior knowledge of the operation area or any specific change in the environment. On the other hand they require more planning and continuous effort by the user.

In the second group, the system works like an autonomous robot, for which the user simply provides a destination, and the robot is in charge of getting there. To reach their destination, these systems typically use a complete map of the environment, or some sort of modifications are required to their environment (e.g., tape tracks placed on the floor or markers placed on the walls). They are usually unable to compensate for unplanned obstacles or travel in unknown areas. This type of robotic wheelchairs are appropriate to users that are unable to plan and execute a path to a destination, and for operations in structured known environments.

A third group of approaches offers semi-autonomous navigation assistance. There is a subset based on variations of the subsumption architecture. This subset relies on a basic set of primitives (e.g. avoid obstacles, pass doorway, following wall) that can be used to assist the user in difficult maneuvers. The responsibility for selecting the most appropriate operating mode can be performed by the user (manual adaptation) or by the robotic wheelchair (automatic adaptation). Another subset is based on shared control approaches, for which the control is distributed between robot and user at all times,

Table 2.2: Relevant research works in the field of robotic wheelchairs.

Project	Description of main features
NavChair (University of Michigan) [Levine 1999]	It uses multiple-task specific operating modes: obstacle avoidance, door passage, and wall following. Automatic adaptation is based on a Bayesian network that combines the information of the Navchair's surrounding with the information provided by a topological map.
ROLLAND (University of Bremen) [Rofer 2000]	RW with three operating modes: turn-in-place, wall following, and trajectory playback. The user teaches trajectories using turn-in-place and wall-following behaviors. The user is also able to drive the wheelchair that modifies its velocity to avoid obstacles.
WATSON (Nara Institute of Sciences and Technology) [Matsumoto 2001]	Uses machine vision to interpret user's gaze to control the wheelchair.
CWA (National Institute of Singapore) [Boy 2002]	Uses a collaborative wheelchair assistant that keeps the wheelchair on a pre-defined path. The user is able to leave the path to avoid obstacles, and is able to control the speed along the path. Paths can be defined by means of a graphic user interface.
RobChair (University of Coimbra) [Pires 2002]	Provides local obstacle-avoidance assistance. User manually switches between general collision-avoidance and wall-following modes.

increasing and decreasing the weight of each one depending on the user's driving ability. Another subset is related to the prediction of user's intention. This subset of approaches allow users with limited motor capabilities to drive a wheelchair with minimum, and possibly unreliable, command effort. A list of some RWs described in literature are presented in Table 2.2 and Table 2.3. Most of these architectures are based on semi-autonomous control approaches of the forms presented in subsections 2.2.1-2.2.3.

Brain-actuated wheelchairs are robotic wheelchairs guided by the use of a BCI. This type of robotic wheelchairs have been studied recently by some research groups. A summary of the main relevant works is presented in Table 2.4. These systems are mainly based on two types of brain control: 1) modulation of sensory-motor rhythms by performing

Table 2.3: (Continued) Relevant research works in the field of robotic wheelchairs.

Project	Description of main features
WAD (University of Marseille) [Mallet 2002]	Provides obstacle avoidance behavior based on infrared distance sensors. The attractor dynamics approach is used to integrate the obstacle avoidance behavior to a user defined target acquisition behavior, in which the direction and the distance to the target are indicated by the user using a graphic user interface.
POMPD (ARC Centre of Excellence for Autonomous Systems) [Taha 2008]	The RW relies on minimal user input (obtained from a standard wheelchair joystick) in conjunction with a learned Partially Observable Markov Decision Process (POMDP), to estimate and subsequently to move the user to his destination.
VAHM (LASC, University Paul Verlaine-Metz) [Grasse 2010]	The VAHM is based on a semi-autonomous controller that uses a multi-agent control architecture with agents of three types: environmental, cognitive, and behavioral. It is able to provide assistance to the user during navigation by proposing the direction to be taken when a path has been recognized. It uses a particle filtering approach to implement the recognition of the most frequent paths according to a topological map, which was constructed offline. This approach will spare the user in determining a new direction.
SHARIOTO (Katholieke Universiteit Leuven) [Vanhooydonck 2010]	An assistive navigation system with a human-machine shared control that includes user's intention estimation, and determines whether the user needs assistance to achieve that intention. An implicit user model is introduced and incorporated in the framework, in order to make the execution of both tasks adaptable to a specific user.

mental tasks (e.g. motor imagery) [Pfurtscheller 1998]; 2) detection of P300 event-related potentials through the design of oddball paradigms [Farwell 1988]. While in motor imagery the number of commands is limited to 3 or 4, P300-BCIs can provide a significant number of commands, but they depend on external visual stimuli (see [Pires 2009] for a practical comparison of some neural mechanism approaches).

A BCI system can work synchronously or asynchronously. Using sensory-motor modulation, the asynchronous operation means that the onset of imagery tasks is not time-cued by the system, but self-paced by the user instead. In P300-based systems, the asynchronous operation assumes the detection of a non-control state, for which no commands are sent. In both cases, it is the user who decides when to send a command. The exper-

Table 2.4: Summary of relevant research works in the field of brain-actuated wheelchairs.

BCI approach	Asynch. control	Description	N. of Participants
Left and right imagery [Tanaka 2005]	No	Control of real wheelchair (commands: left, right)	6 able-bodied
Motor imagery [Vanacker 2007, Philips 2007]	Yes	Control of real wheelchair (commands: forward, left, right)	5 able-bodied
Motor imagery (left hand and rest) and words association [Galán 2008]	Yes	Control of a simulated wheelchair in 3D environment (commands: left, right, forward).	2 able-bodied
P300 visual paradigm built in a virtual 3D reconstruction of environment [Iturrate 2009]	No	Control of real wheelchair and simulated wheelchair in virtual environment (selection of local surrounding points).	5 able-bodied
P300 visual paradigm combined with motor imagery [Rebsamen 2010]	Yes	Control of real wheelchair (selection of high-level destination goals with P300 (e.g. kitchen) and stop detection with motor imagery).	5 able-bodied
Motor imagery (left and right hand) [Huang 2012]	Yes	Control of 2D simulated wheelchair (commands: go, stop, right, left).	5 able-bodied
P300 visual paradigm [Lopes 2012b]	No	Control of real wheelchair (Visual arrow-paradigm encoding 7 symbols for steering control, e.g. FORWARD, RIGHT45, etc.).	10 able-bodied and 1 disabled

imental results achieved with brain-actuated wheelchairs have so far been very positive, but show that its effective application in real-world scenarios is very difficult. The low transfer rate and low robustness of BCIs, as well as demanding requirements of human-centered robots still pose many research challenges, some of them being pursued in this work.

Chapter 3

Assistive navigation system: materials and methods

Contents

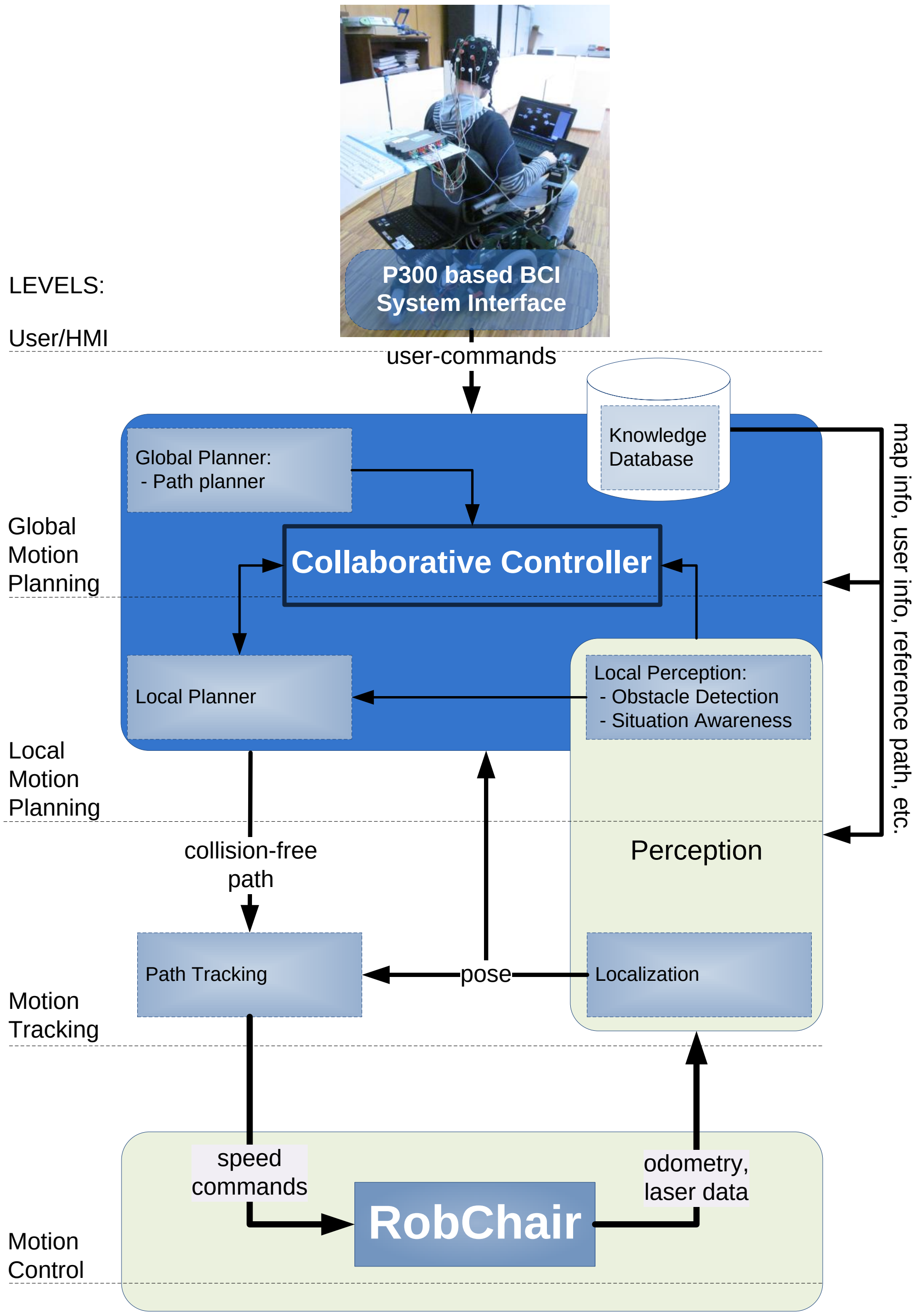


Figure 3.1: ANS - Assistive navigation system architecture.

3.1 ANS architecture

The ANS has been tested in player/stage [Gerkey 2003] simulation environment, and in RobChair [Lopes 2011]. It is structured in five main levels, as shown in Fig. 3.1:

- HMI: provides user intent (sparse steering commands) to the system through a P300-based BCI. Inputs: ElectroEncephaloGraphy (EEG) signals; Outputs: user command (also denoted by brain-command) to the global motion planning and collaborative controller.

- Global Motion Planning: is responsible for determining reference paths to predefined goals. Two different approaches were adopted for global planning. In the first approach the user provides a goal using the system HMI. For this case we use the A* algorithm to determine the global least-cost path [Hart 1968], [Pearl 1982] to the selected goal. This method is implemented online, and it was used in the simulations in player/stage environment. In the second approach, goals and waypoints are predefined, and multiple paths between waypoints are determined offline, using the method proposed in [Solea 2009]. The second approach was used in the experiments with RobChair. Inputs: map info from knowledge database, goal position from HMI (first approach), waypoints (second approach); Outputs: global path (also denoted by reference path) provided to local motion planning.

- Local Motion Planning: is responsible for providing the motion tracking level with a collision-free path that should converge to the reference path after obstacle-avoidance. The local planner links to the knowledge database, perception, and collaborative control modules. Inputs: pose and admissible openings from perception module; final steering command from the collaborative controller; and a subgoal in the reference path provided by knowledge database; Outputs: steering command candidates provided to the collaborative controller; and a collision-free path provided to the motion tracking level.

- Motion Tracking: determines the speed reference commands for the motion controllers. Inputs: collision-free path; Outputs: linear and angular speed commands to the motion control level.

- Motion Control: is responsible for the robust velocity servo control. Inputs: linear and angular speed commands that are converted by inverse kinematics to angular speed commands to the wheels' controllers; Outputs: odometry determined by direct kinematics based on wheel encoders data, and laser data.

The ANS also integrates a knowledge database that stores information regarding the working environment, namely, situation-based restrictions, and driving rules. It also stores the global paths provided by the global planner. When required, this information is provided to several levels of the ANS, as depicted in Fig. 3.1.

The perception module is in charge of maintaining a local environment model. It is also responsible for determining the current pose of the RW, using a localization system that fuses dead-reckoning data with map-based information.

The key function of the collaborative controller is to determine a set of maneuvers to reach a predefined goal. This module relies on both, the ANS and sparse user commands, resulting in a human-machine collaborative control system.

3.1.1 HMI

The ANS proposed in this thesis was developed to provide the RW with a level of autonomy, by using discrete and sparse HMIs. An ANS with these characteristics can be used by subjects suffering from severe motor disorders, such as CP or ALS. At least one specific motor skill is required to operate most of HMIs. In this thesis, we propose to use a BCI, as the system HMI, to steer the RW, since it opens a new communication channel for users that are severely motor impaired. However, the number of decoded Symbols Per Minute (SPM) is still low for a continuous control of a wheelchair. Due to this fact, users are only able to provide sparse commands over time. To overcome this problem, the RW also relies on the ANS to achieve a safe and effective navigation. Additionally, and to ease user effort, low-level commands are only issued when there are dynamic changes of the environment, or when ambiguous situations occur. The BCI system used in the ANS proposed in this thesis was developed in the work of [Pires 2012a]. It is based on EEG that translate brain patterns into low-level commands to steer a RW [Lopes 2011, Lopes 2012b, Lopes 2012a]. In alternative to the BCI, a scanning interface combined with a single-switch / multiple-switch is also proposed. This type of HMI is also able to provide the ANS with low-level sparse commands, similar to those provided with the BCI system.

3.1.2 Global motion planning

3.1.2.1 Global path-planner

We use the A* algorithm [Hart 1968], [Pearl 1982] to find the least-cost path from a given initial node to a goal node. This approach is an exploration algorithm in the graph theory, and it fits well to grid space modeling. The choice of the next node to be analyzed is determined by heuristics that uses the distance from the current node to the goal.

An a priori occupancy grid map is provided to the global planner. The global path is determined whenever the user selects a global goal node he/she wishes to reach. In this approach, velocity and acceleration constraints are not considered, and the main goal is to find the least-cost path to a final node.

3.1.2.2 Path smoother

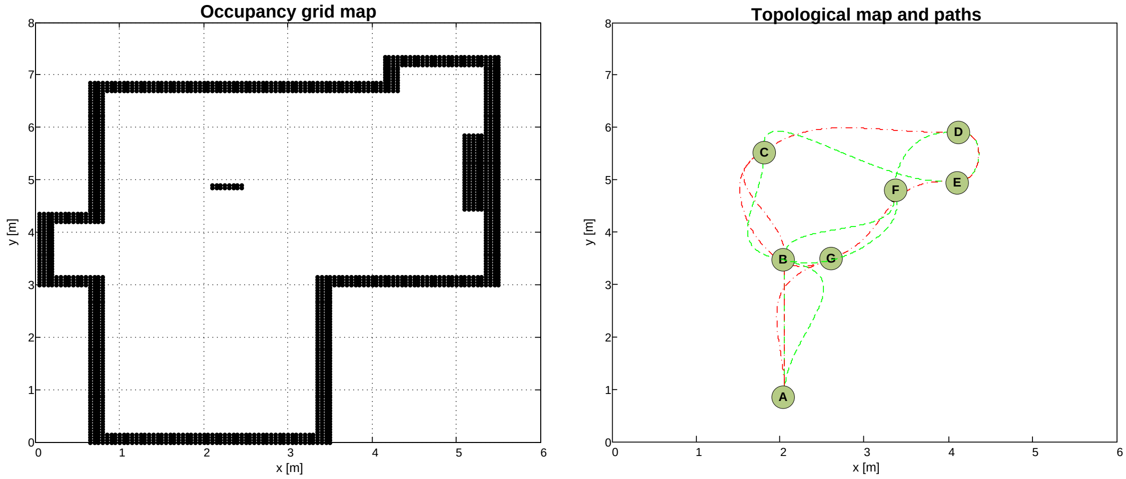
The path-planner proposed in [Solea 2009] was used to establish multiple reference paths between a set of predefined waypoints. For this approach, we provided a topological map to the global planner that contains all waypoints in the free space. All possible paths between waypoints are then determined and stored. This approach was implemented offline in the experiments presented in this thesis.

3.1.2.3 Grid and topological maps

At the current stage of RobChair, the ANS is provided with a type of location-based map, normally known as an a priori occupancy grid map of the environment, such as the one depicted in Fig. 3.2 a) that was used as the base scenario for the experimental tests. Occupancy grid maps are location based, since they assign to each x-y coordinate a binary occupancy value, which specifies whether or not a location is occupied with an object. Each grid node has a fixed dimension and an associated binary occupancy value that depends of the presence of obstacles. Grid maps have several advantages for mobile robot navigation, since they make it easy to find objects through the unoccupied space. However, map construction, and adjusting the position of objects in the grid map may become a difficult task. RobChair is also provided with a topological map, constructed offline, containing all possible goals and waypoints (bifurcations, and final destinations) of the environment, as shown in Fig. 3.2 b).

3.1.3 Local motion planning and motion tracking

The local planner is based on an adaptation of the VFH+ algorithm [Ulrich 2000]. The local planner is provided with a binary histogram by the obstacle-avoidance module. It then calculates the admissible openings in the free space, in a way to ensure safe navigation of the RW. A set of candidate directions is then derived from the previous admissible openings. A cost function is applied to each candidate in order to determine the steering angle that minimizes it. A blending algorithm is applied to merge local and global paths, once the obstacles are overcome. Details on this subject are given in section 6.2.



a) b)

Figure 3.2: A priori map used as the base scenario for the experimental tests: a) grid map; b) topological map with possible paths.

A sliding mode path-following controller, which uses the lateral error, was adopted as the system path tracker [Solea 2009] that generates the velocity vector (v_c, ω_c).

3.1.4 Perception

The perception module is constituted by a localization module and an obstacle detection module, as shown in Fig. 3.1.

3.1.4.1 Obstacle detection

The obstacle detection module detects new obstacles in the environment using a matching algorithm, which compares a current laser scan with a predicted laser scan determined from a priori occupancy grid map. When a new obstacle is closer than a certain threshold, the obstacle detection module constructs a polar field histogram that assigns to each scan sector a value that varies with the proximity to the detected obstacle. A binary histogram that assigns the value zero if the sector is free of obstacles, and a value one if the sector is occupied is then constructed. A sector is considered free of obstacles if the correspondent value in the polar field histogram is below a certain threshold.

3.1.4.2 Localization

Localization is performed using dead-reckoning data (odometry) for rough positioning, and laser data for polar scan matching. Two approaches were used to fuse odometry with laser polar scan matching: a Markov grid localization and Monte Carlo particle filter. Details on this subject can be found in chapter 4.

3.1.5 Collaborative controller

The key function of the collaborative controller is to determine a set of maneuvers to reach a predefined goal. This module relies on both human and machine agents' commands, resulting in a human-machine collaborative control system, whose effectiveness depends on both the machine agent design and on the user ability to steer the RW. Details on this subject are given in chapter 6.

3.2 Human-machine interface

A robot is commonly viewed as a tool: a device which performs tasks on commands. As such, a robot has limited freedom to act and will perform poorly whenever its capabilities are ill-suited for the task at hand [Fong 2002]. However, many times the only thing the robot needs to get out of trouble is to ask and receive advice (even a small amount) from a human.

Consider the situation in which a RW is navigating to a goal destination and suddenly faces a deadlock due to unexpected obstacles. Depending on the RW's autonomy, it may be unable to proceed or may decide to take a long detour. If, however, the robot is able to discuss the situation with a human, a better solution can be found. It seems clear that there are benefits to be gained if human and robots work together. In particular, if we treat a robot not as tool, but rather as a partner. To do this, it is necessary to enable collaboration between the humans and the robot: to engage in dialogue, to ask questions to each other, and to jointly solve problems [Fong 2002, Takeda 1997].

"Human-robot interaction can be defined as the study of the humans, robots, and the ways they influence each other" [Fong 2001b]. As a discipline HRI regards the analysis, design, modeling, implementation, and assessment of robots for human use. HRI can be done directly, with proximal interaction (e.g. physical contact) or can be mediated by an HMI. The latter translates human intentions to robot commands, and may provide feedback via displays. If the human and the robot are separated by a barrier, then the interaction is called teleoperation.

3.2.1 Taxonomy for HRI: the specific case of robotic wheelchairs

In [Yanco 2004] an extended taxonomy for HRI is proposed. Whereas classifications include measures of the social nature of the task (human interaction roles and human-robot physical proximity), task type, and robot morphology. Moreover, the amount of intervention required for controlling a robot is also part of the proposed taxonomy, since it is considered as one of the defining factors for human-robot interaction, and is directly related to the level of autonomy of the robot.

Robot control ranges from teleoperation to full autonomy, and, consequently, the amount of required intervention varies accordingly to the level of control. A constant interaction is required at the teleoperation level, and less interaction is required as the robot has increased autonomy.

The taxonomy proposed by [Yanco 2004] measures the autonomy level and amount of intervention required. The autonomy level measures the percentage of time that the robot is carrying out its task on its own; the amount of intervention required measures the percentage of time that a human operator must be controlling the robot. These two measures sum to 100%. A teleoperated robot is fully controlled by a robot operator. This type of robots has 0% of autonomy and 100% of required intervention. At the other end of the spectrum are robots with full autonomy. For this case robots have 100% of autonomy and 0% of required intervention. In between these two levels of autonomy is a large continuum of robot control, often called semi-autonomous control. RWs mostly belong to a variable type of autonomy that varies according to one decisive factor: the human agent.

3.2.1.1 Classification of HMIs according to [Zhai 1992]

Robotic wheelchairs are intended to increase mobility of people that are motor impaired, in a permanent or temporary manner. When designing the ANS for the RW the primary focus is the human agent. A characterization of the typical RW user has to be done previously. Understanding the capabilities of the users that will use the RW is paramount. For instance, It is not reasonable to expect that a user suffering from severe motor disorders, not able to control the movement of arms and legs, can use a conventional joystick to steer the RW. The first step in the designing process of the ANS consists in analyzing the capabilities of potential users for the RW, and select the most appropriate(s) HMI(s). The ANS proposed in this thesis was developed to provide the RW with a level of autonomy, by using discrete and sparse HMIs, and it can be used by subjects suffering from severe motor disorders, such as CP or ALS. In the case of neuro-degenerative motor disorders, the level of functionality depends on the stage of the disease, and can go to complete locked-in states [Kubler 2008]. In non-degenerative motor disorders, such as CP, the symptoms vary significantly among patients, and can go from a simple difficulty to walk to rough motor control, or total lack of control of muscular activity.

Potential HMIs for severe motor control impaired users include those that only provide a small set of sparse and discrete commands, such as: go forward, turn right, etc. Possible HMIs of this type include: switch/multiple-switch with scanning interface, and some paradigms for BCI. These types of HMIs lead us to a taxonomy for human-robot communication in telemanipulation proposed by [Zhai 1992] that classifies the HMIs according to language, in particular, continuous and discrete languages.

In [Zhai 1992] a general notion of ”continuous” vs. ”discrete” language was used to describe the two different coding mechanisms in human-robot information exchange. Continuous language is a means of representing information which is distributed continuously, along either a spatial or a temporal dimension. In human-to-human communication, painting, dancing, music all are examples of continuous language. In human-machine communication, tool using and analogue displays exemplify this continuous format. In contrast, discrete language consists of independent elements. It is a superset of verbal language. In general, written text, oral conversation, computer programming language and symbols and signs in every day life fall under the category of discrete language.

Can information which is encoded in a discrete or continuous language ”translated” to the opposite format? An initial hypothesis on this relationship between discrete and continuous languages is a ”generalized sampling theorem”. In communication engineering, an analogue signal can be converted to a digital signal without losing any information, provided the sampling frequency is at least twice as high as the highest frequency component in the original signal. An analogy to this applies to the notion of continuous vs. discrete languages. For instance, when one verbally describes (using discrete language) another person’s appearance (which comprises continuous, spatial information), usually only primary characteristics (low frequency components) are transferred to the listeners, but the details (high frequency components) are lost without sufficient sampling.

The taxonomy proposed by [Zhai 1992] is easily extended to mobile robotics, and to RWs, in particular. RWs are inherently continuous and spatial. In light of the generalized sampling theorem, discrete (command) languages can be used in human robot communication only when the information bandwidth required for any particular task is sufficiently low. In other words, only when the robot has sufficient autonomy. The classification proposed by [Zhai 1992] was adopted to classify the HMIs that are used in the ANS described in this thesis. In this context, a discrete HMI is considered as a HMI capable of providing discrete and sparse commands.

3.2.2 HMI commercial solutions

Most of powered wheelchairs are operated through joystick by hand. However, the hand function may be limited or even not available in some patients with severe disabilities. An alternative HMI may use other part of the body to operate it. The work of [Fehr 2000] shows that 10% of the patients with motor disabilities find it extremely difficult or impossible to use the wheelchair for activities of daily living. When asked specifically about steering and maneuvering tasks, the percentage of patients that found these difficult or impossible jumped to 40%. Most of the patients analyzed in this study, 81% to be precise, used a conventional joystick to steer the wheelchair. From the remaining patients, 9% used head or chin HMIs, 6% used sip and puff, and 4% used other interfaces such as eye gaze, tongue pad, head, hand or foot switches. A problem immediately arises from these numbers: a severe motor impaired user, such as those suffering from ALS or severe forms of CP disorders, are not able to control most of these HMIs (joystick, head or chin controllers, and sip and puff) because they lack fine motor control. According to the taxonomy proposed by [Zhai 1992] these HMIs would be classified, in terms of language, as continuous since they provide a continuous input over time. For these users an HMI solution that provides a discrete input over time seems to be the most appropriate. There exist some commercial solutions of this type including: scanning interface combined with single-switch / multiple-switch, eye gaze, and voice control. The latter presents some drawbacks because many of these users are also speech impaired.

The scanning interface is a possible HMI solution, since it does not require fine motor control of a specific part of the body, such as arms, legs or head. Scanning is the technique of successively highlighting items on a computer screen or other electronic device and pressing a switch when the desired item is highlighted [Biswas 2011]. This solution has some advantages: is simple and is flexible, once the switch, or multiple switches, can be used by the part(s) of the body the user controls better. However it presents some important disadvantages. One of the most relevant is the level of fatigue and stress caused by the use of this HMI. For instance, the combination of a head-switch with a scanning interface for a user that is only able to control the head movement seems to be a good choice. However, since he/she needs to do the same head movement repeatedly, in order to select among the options provided by the scanning interface, he/she usually gets tired rapidly. Using multiple-switches can help to prevent fatigue, since different parts of the body could be used to select the options provided by the scanning interface. Nevertheless, many users are bound to use only a part of his/her body. Moreover, these systems are slow to operate. For the ANS proposed in this thesis we use a graphical scanning interface also designated as the visual arrow paradigm, which is depicted in Fig. 3.3 b).

Another HMI solution for people with motor disabilities include eye tracking based interfaces that use the eye gaze as a binary input like a switch press input through a blink [Duchowski 2007]. This type of interface is normally used to substitute the mouse and computer keyboard, but can also be used to steer a RW. The resulting system remains as slow as the scanning system combined with a single-switch / multiple-switch. Additionally, it is quite strenuous to control the cursor through eye gaze for long time as the eye muscles soon become fatigue. In the study of [Fejtová 2009] eye strain in six out of ten able-bodied participants was reported. Another solution may be to use the eye gaze to directly control a pointer position in a screen. The work of [Zhai 1999] presents a detailed list of advantages and disadvantages of using eye gaze-based pointing devices. In short, using the eye gaze for controlling the cursor position pose several challenges, namely: strain, as posted before; the eye gaze tracker does not always work accurately; it often makes clicking on small target difficult. In [Donegan 2009] problems in precision and speed of an eye gaze-based systems were also reported.

A multi-modal solution that alternates the use of eye gaze with scanning interface with single-switch / multiple-switch was proposed by [Biswas 2011], for computer interface applications. The technique is less strenuous than the only eye gaze-based interfaces because users can switch back and forth between eye gaze tracking and scanning which gives rest to the eye muscles. This technique does not depend on the accuracy of the eye tracker, as eye tracking is only used to bring the cursor near the target (as opposed to on the target), so it can be used with low cost and low accuracy webcam-based eye trackers. One disadvantage of this technique is that it seems slower than only eye gaze-based interface as users need to switch back to the slower scanning technique for each pointing task.

3.2.3 Brain-computer interface

The use of BCI to control a RW is very challenging because BCI yields low transfer rates, and the decoded brain-commands have an associated uncertainty. The low Signal-to-Noise Ratio (SNR) and the non-stationarity of the Electroencephalography (EEG) signal make EEG classification a challenging task. The deployment of efficient signal processing and machine learning techniques to classify the brain patterns are a key issue to decrease the

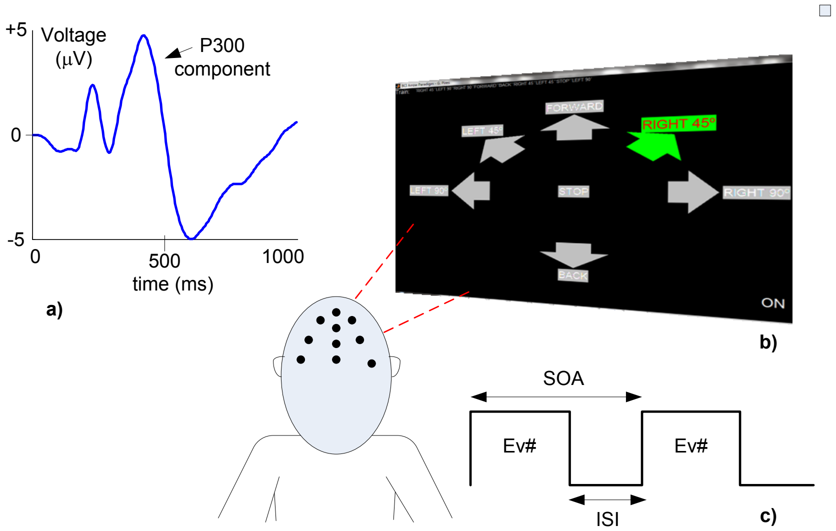


Figure 3.3: Overview of the BCI system: a) P300 component of the ERP: positive deflection occurring approximately at 300 ms after the onset of the relevant stimulus; b) Visual arrow-paradigm encoding 7 symbols for steering control; c) Temporal diagram of the events ($Ev\# \in \{1 \cdots 7\}$ represents the code of the event).

uncertainty. In the other hand, increasing the amount of EEG data used for classification also increases the accuracy, but requires more time and thereby reduces the transfer rate. The BCI must be designed as a compromise between the transfer rate and the accuracy, trying to maximize the transfer rate while keeping the accuracy above a reasonable value. The BCI system developed in [Pires 2012a] is used in the proposed ANS. It is based on the detection of a brain pattern called P300 component, which is an Event Related Potential (ERP), elicited by a relevant event within an oddball task, that occurs typically around 300 ms after the relevant event occurs [Farwell 1988] (see Fig. 3.3a)). The BCI, depicted in Fig. 3.3b), has been designed as a visual oddball paradigm, where the symbols flash randomly. At a given moment, the relevant (target) event is the symbol mentally selected by the user, which corresponds to the direction he/she wants to follow, and all other flashing symbols are the standard (non-relevant) events. The BCI commands are not issued in a self-paced manner by the user, which would allow the user to issue commands only when he/she desires. Additionally, the BCI system is still not able to identify when the user is in an idle state (not focusing any symbol flash), and in those situations it acts like if the user was selecting an available command.

3.2.3.1 P300-based BCI paradigm

The paradigm comprises seven steering commands, encoded by the following symbols: FORWARD, RIGHT45, RIGHT90, BACKWARD, LEFT45, LEFT90 and STOP (Fig. 3.3b)). These symbols flash randomly with an Inter-Symbol Interval (ISI) of 75 ms, and a flash duration of 100 ms, i.e., the Stimuli Onset Asynchrony (SOA) is 175 ms (Fig. 3.3c)). Because of the low SNR of P300 ERPs, several P300 responses have to be collected before machine learning algorithms can identify the mentally selected symbol. Therefore, the overall time needed for symbol detection (TT - trial time) depends on the number of event repetitions (N_{rep}), yielding

$$TT = N_{rep} \times N_s \times SOA + 1 \tag{3.1}$$

where N_s is the number of symbols ($N_s = 7$), and the value 1 is the time required to record the EEG associated with the last event of a trial (e.g., for a user requiring 3 repetitions, the TT will be $7 \times 0.175 \times N_{rep} + 1 = 4.675$ s). The number of repetitions is adjusted for each participant according to his/her offline accuracy, obtained during the BCI calibration. A threshold value of 90% accuracy was settle to select the number of repetitions. Comparing with the arrow paradigm approach proposed in [Lopes 2011, Pires 2012a], the number of symbols was further reduced from 11 to 7 in order to decrease the TT.

3.2.3.2 EEG Signal Acquisition and Classification

The EEG was recorded by a biosignal amplifier (gUSBamp, g.tec Inc.), from 12 passive electrodes at positions Fz, Cz, C3, C4, CPz, Pz, P3, P4, PO7, PO8, POz and Oz according to the extended international 10-20 standard system. The EEG was notch-filtered at 50 Hz and bandpass filtered between 0.5 and 30 Hz, and then sampled at a 256 Hz rate.

The detection of the P300 patterns uses the algorithms described in [Pires 2011b]. This classification methodology was already thoroughly tested and validated with success in experiments made by able-bodied and motor disabled participants [Pires 2012b]. Feature extraction is applied to segments (epochs) of EEG data associated with each event. A segment has $T = 256$ time samples corresponding to one second. Features are extracted by applying an optimal statistical spatial filter to the original dimensional space (12 $channel \times T$), resulting in two high SNR projections, which are then concatenated into a feature vector. A binary Bayesian classifier separates the feature vector into target and non-target events, and associates a posterior probability to each event classification. In the next step, the classifier selects the event with the highest probability as being the

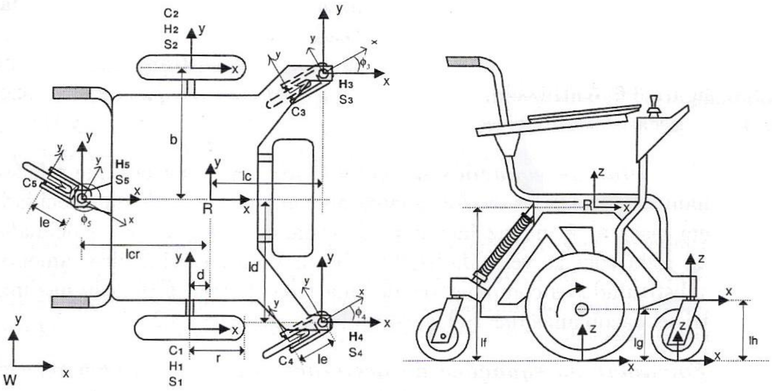


Figure 3.4: RobChair: mechanical structure and system coordinates.

most likely symbol mentally selected by the user (see details in [Pires 2011b]). Signal processing and classification models are fitted to each participant based on the EEG data collected during the calibration session that occurs before the online operation. This calibration takes approximately 3 minutes.

3.3 RobChair setup

3.3.1 Equipment architecture

RobChair is composed by two motorized rear wheels with two casters in front. There is also a fifth rear wheel connected to the back of the wheelchair with a damper used for stability. Figure 3.4 shows the RobChair mechanical structure and the associated system coordinates. During the lifetime of the project, several electronic components have been attached to the base platform in order to improve environment perception and actuation capabilities.

Figure 3.5 presents a block diagram of the actual hardware control architecture. The wheelchair is powered by two 12 V batteries feeding two permanent magnet DC motor with 24 V input voltage and 1000 RPMS. These motors are coupled to two gearboxes with factor 1:10 (one complete wheel revolution corresponds to 10 complete motor revolutions). With the aid of these gearboxes, each wheel may have a nominal torque of 29,3 Nm. There are two power drivers that guarantee the independent and direct control of the motors. Two encoders have been coupled to motor axis, providing the velocity feedback of each motor. The encoders are also used to obtain dead-reckoning data for integration into localization methods.

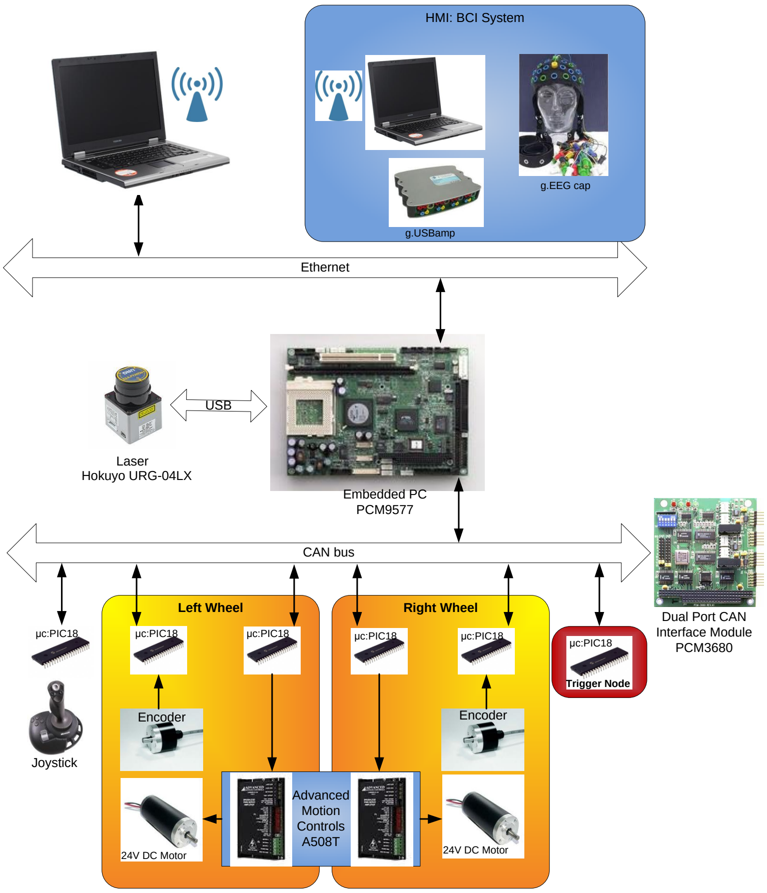


Figure 3.5: RobChair: system setup.

Other types of user interfaces, such as a joystick, are also installed. One Hokuyo URG-04LX laser range finder with the capability to scan 240 degrees, is also integrated on the platform. A real-time software architecture was developed taking advantage of the above described features in order to effectively implement the navigation architecture designed for the wheelchair.

An Embedded-PC is responsible for giving some degree of intelligence to the robot. This computer is connected to distributed devices through fieldbuses. The platform has also the capability to connect to external devices, such as the BCI system, through a wireless link, allowing networked robotic research, such as multi-robot cooperation, and its integration in intelligent environments.

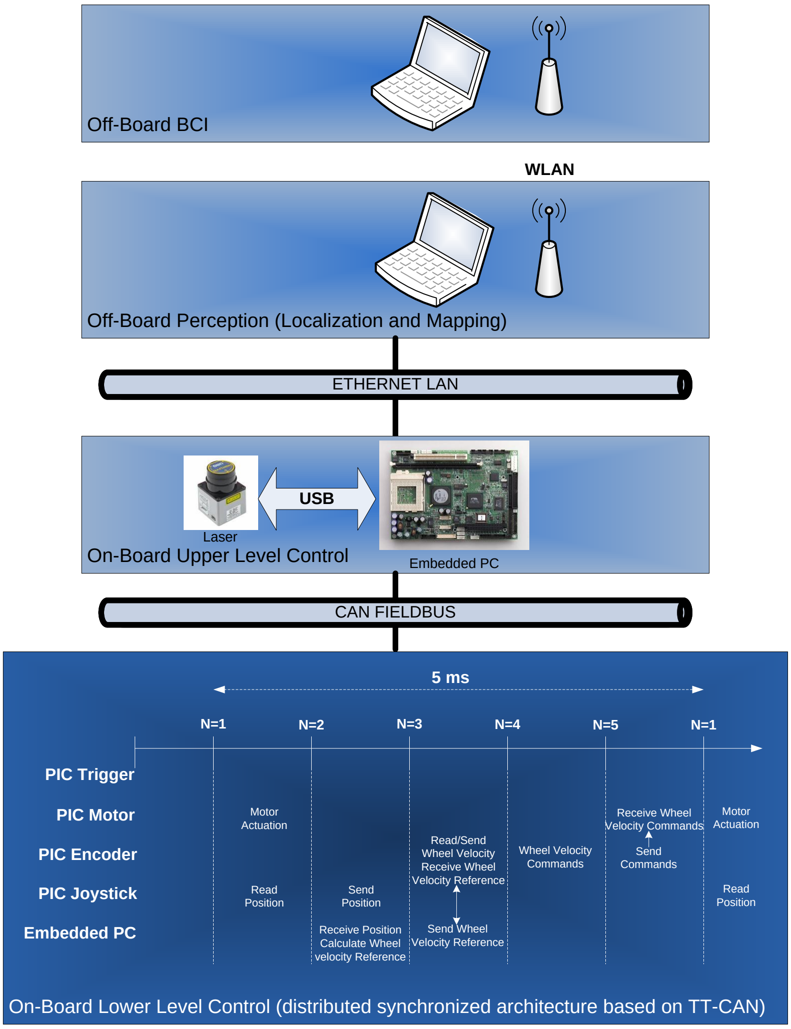

Off-Board BCI
WLAN
Off-Board Perception (Localization and Mapping)
ETHERNET LAN
USB
Laser
On-Board Upper Level Control
Embedded PC
CAN FIELDBUS
5 ms
N=1
N=2
N=3
N=4
N=5
N=1
PIC Trigger
PIC Motor
PIC Encoder
PIC Joystick
Embedded PC
Motor Actuation
Read Position
Send Position
Receive Position Calculate Wheel velocity Reference
Read/Send Wheel Velocity Receive Wheel Velocity Reference
Send Wheel Velocity Reference
Wheel Velocity Commands
Receive Wheel Velocity Commands
Send Commands
Motor Actuation
Read Position
On-Board Lower Level Control (distributed synchronized architecture based on TT-CAN)

Figure 3.6: RobChair: communication system.

3.3.2 Communication system

The overall RobChair communication system is constituted by two communication modules based on TCP/IP communication protocol, and other communication modules based on fieldbuses, namely Controller Area Network (CAN) and Universal Serial Bus (USB), as presented in Fig. 3.6. To enable message exchange between RobChair and BCI, a communication system was developed to mainly allow the reception of BCI commands (user intent) by the RobChair. The platform connects to the BCI system, through a wireless LAN, as shown in Fig. 3.6. The BCI commands are not issued in a self-paced manner by the user, which would allow the user to issue commands only when he/she desired. At the present, the BCI system acts like a passive system, delivering a command to the RW, periodically, even when the user is in an idle state (not focusing any symbol flash). User commands are sent in time intervals that depend on the TT determined for each user, according to (3.1), during the BCI training phase.

As depicted in Fig. 3.5 and Fig. 3.6, the on-board devices distributed through the RW platform are connected through fieldbuses, namely:

- CAN;
- USB.

CAN is used for data transfer of small critical messages between devices, while USB is mainly used for devices, which send or receive large amounts of data. A custom communication protocol, based on the Time-Triggered CAN (TT-CAN) protocol paradigm, was designed and implemented [Nunes 2003, Silva 2005]. All events are synchronized by a message sent from a trigger node based on a Microchip PIC18F258 microcontroller that synchronizes all other microcontroller units dedicated to sensory and actuation systems that do not require large processing power, designated as on-board lower level control. A synchronization message is sent every 1 ms by the trigger node, instructing all low level sensory and actuation units to perform data acquisition or actuation. Tasks requiring more processing power like path-following, path-planning, obstacle-avoidance, or laser-based perception, designated as on-board upper level control tasks, are implemented in a embedded PC. The latter communicates with the on-board lower level control through the CAN bus, since the implemented time-triggered protocol is able to accommodate synchronous and asynchronous traffic in the same fieldbus.

Other tasks requiring even greater amounts of processing power, such as localization and mapping are implemented in an external laptop that communicates with the embedded PC through an Ethernet LAN. The RobChair data flow diagram is depicted in Fig. 3.7.

3.3.3 ANS simulator

A ANS simulator was developed for player/stage simulation environment [Gerkey 2003], in order to more easily test the algorithms and other types of functionalities encompassed in the ANS architecture presented in section 3.1. Figure 3.8 shows the simulator environment, and Fig. 3.9 shows the graphical scanning interface used to select a small set of navigation commands by the user. The graphical scanning paradigm comprises eight steering commands, encoded by the following symbols: FORWARD, RIGHT45, RIGHT90, BACKWARD, LEFT45, LEFT90, BACKWARD and STOP. These symbols flash sequentially with a flash duration of 800 ms.

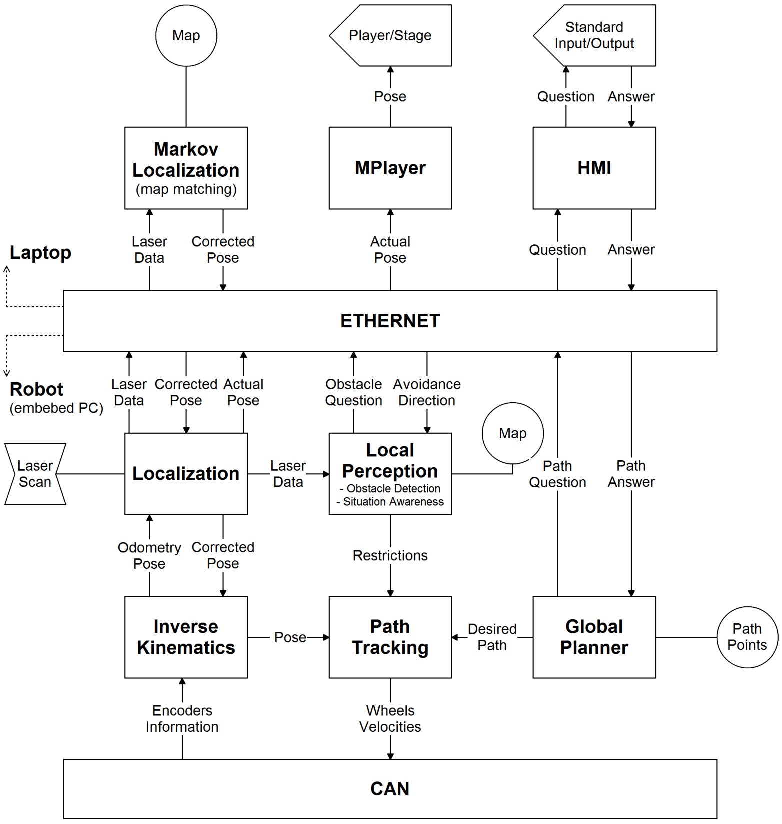


Figure 3.7: Robchair data flow diagram.

The current version of the simulator includes six main functionalities:

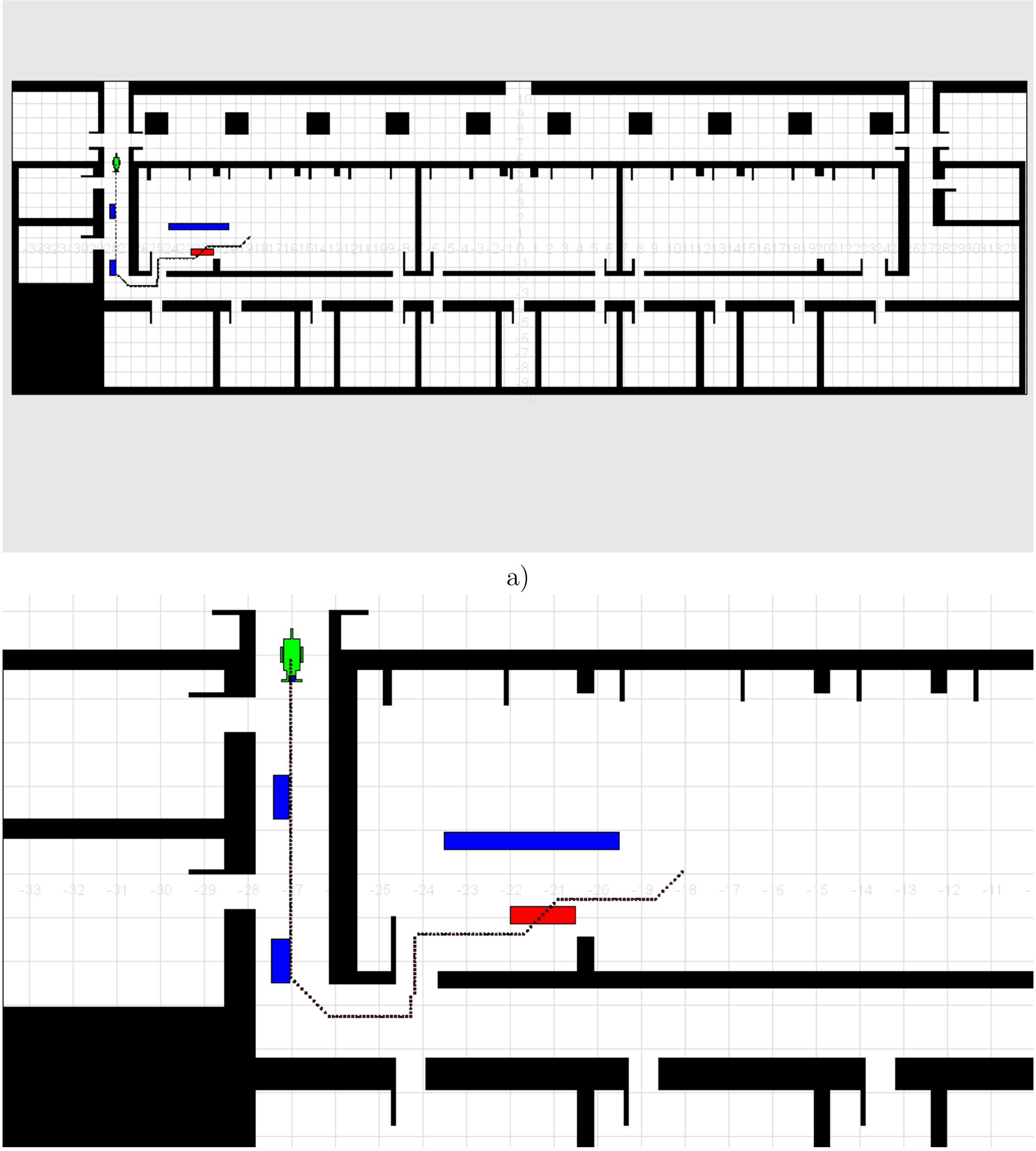
HMI: the user is able to select from a small set of navigation commands using the graphical scanning interface depicted in Fig. 3.9. The selection of a new direction can be performed with the mouse or with a switch device. Each symbol flashes during 800 ms, however this may be adapted to each user performance.

Path planner: the A* algorithm is used for path planning when the user selects a new goal (x,y)-coordinates in the graphical scanning interface.

Obstacle detection: this functionality is assured by a map matching algorithm based on laser scanner.

Local planner: a modified version of the VFH+ algorithm [Ulrich 2000] is used for local planning.

Path following: a sliding-mode algorithm developed by [Solea 2009] is used for path following.

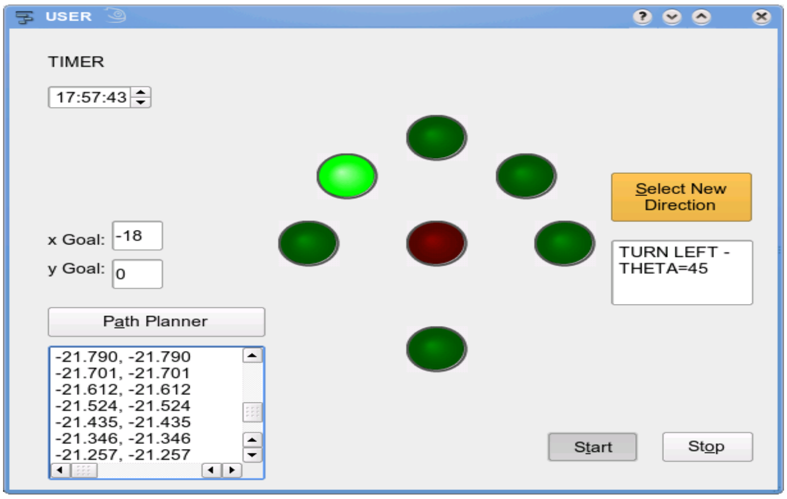


a)

b)

Figure 3.8: ANS simulator: a) screenshot of the ANS global environment; b) screenshot of a detailed part of the ANS environment.

Collaborative control: a two-layer collaborative controller is used to manage the information provided by the two system agents: human and machine, being the core of the ANS.



TIMER

17:57:43

x Goal: -18

y Goal: 0

```
-21.790, -21.790
-21.701, -21.701
-21.612, -21.612
-21.524, -21.524
-21.435, -21.435
-21.346, -21.346
-21.257, -21.257
```

TURN LEFT - THETA=45

Figure 3.9: Graphical scanning interface.

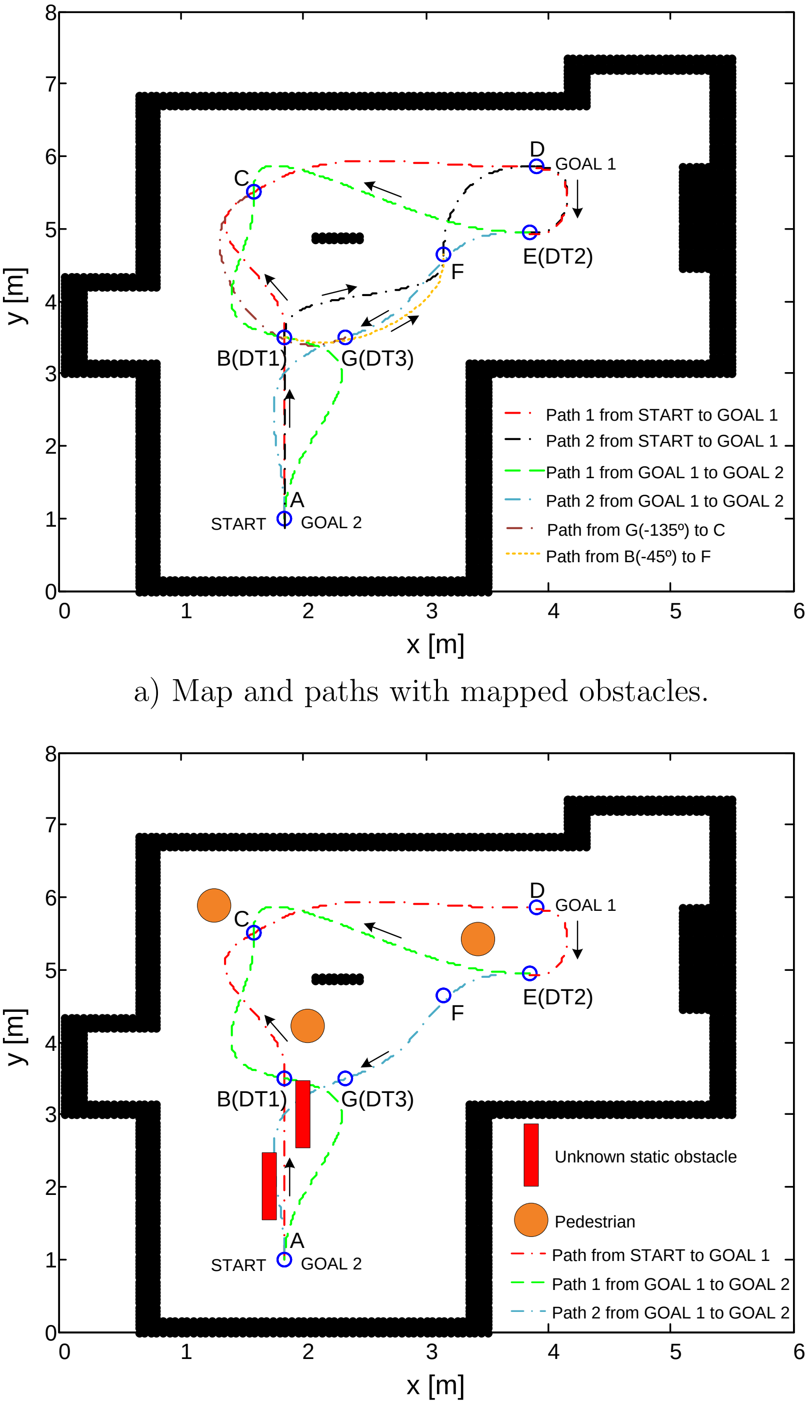
3.4 Experimental setup

3.4.1 Description of scenarios

Two types of environments were used in the experiments reported in this thesis. The first type consisted in a structured known environment that only included static mapped obstacles (see Fig. 3.10 a)); the second type consisted in a structured semi-unknown environment, which included new obstacles in the environment of two types: static, and some moving obstacles, such as pedestrians walking in the set (see Fig. 3.10 b)). The office-like grid map used as a priori occupancy grid map for our experiments has a dimension of $(5.6 \times 7.4)\ m^2$. Each grid cell has a dimension of $(5 \times 5)\ cm^2$. Figure 3.11 shows a scene of the real test environment.

3.4.2 Participants

Ten able-bodied subjects and one disabled subject with cerebral palsy participated in the experiments described in this thesis. All participants gave informed consent, and in the case of the CP participant it was approved by the Cerebral Palsy Association of Coimbra (APCC) to participate in the study. Participants without disabilities included: aged between 25 and 40 years old; approximate number of participants from both genders (six male and four female); higher education; all right-handed; without any relevant history of psychiatric or neurological disorders, all of them being researchers or graduate students at University of Coimbra. The female CP participant (APPC patient) was 20 years old



a) Map and paths with mapped obstacles.

b) Map and paths with unknown static and dynamic obstacles.

Figure 3.10: a) Map and paths with only mapped obstacles; b) Map and paths, including unknown static and dynamic obstacles.

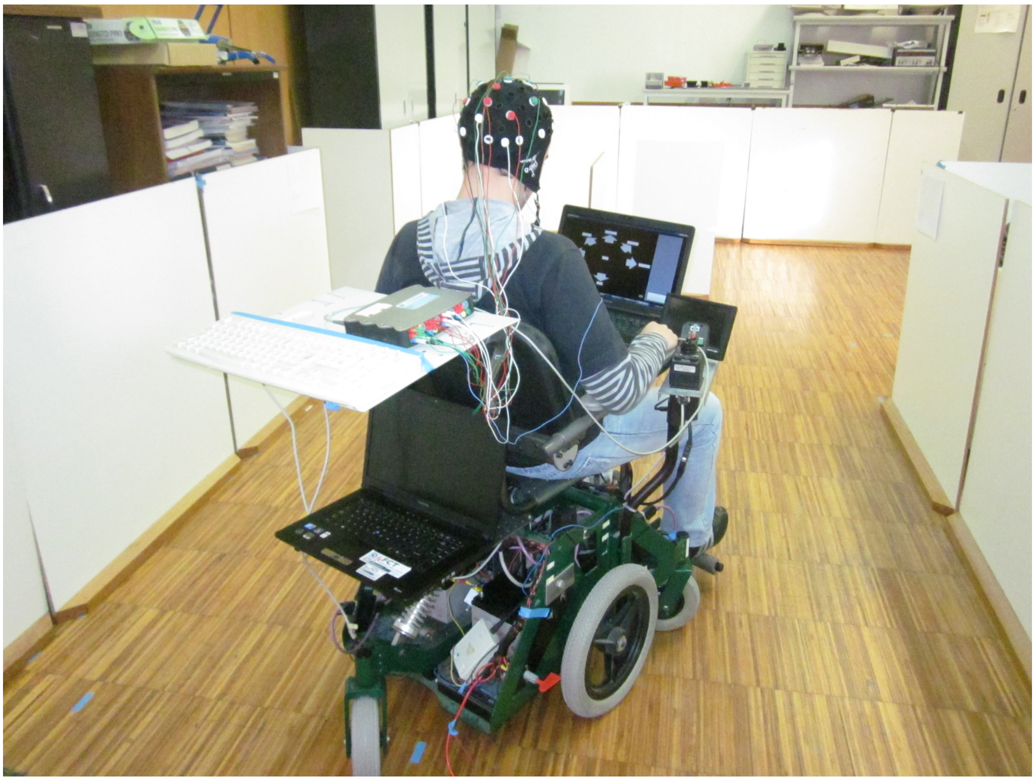


Figure 3.11: Robchair in test scenario.

by the time of the experiments.

3.4.2.1 Cerebral palsy

Cerebral palsy is a non-progressive neurological disorder resulting from a brain injury that occurs before cerebral development is complete [Kriegger 2006]. The etiology can be prenatal, perinatal, or postnatal. It is generically characterized by abnormal movements and posture usually accompanied by dysarthria. The Surveillance of Cerebral Palsy in Europe (SCPE) divides CP into three groups based on the predominant neuromotor abnormality: spastic, dyskinetic, or ataxic, with dyskinesia further differentiated into dystonia and choreoathetosis [Bax 2005]. CP is also classified by topographical pattern of limb involvement, such as diplegia, hemiplegia, or quadriplegia [Jones 2007], however a terminology indicating explicitly the number of limbs affected is now preferred. The CP subtypes are briefly defined: spastic (hypertonia in the muscles that result in stiffness); dyskinetic (involuntary movements, variable muscular tonus); dystonia (hypertonia and involuntary movements); choreoathetosis, combination of chorea (irregular migrating contractions) and athetosis (twisting and writhing); ataxic (poor balance and equilibrium and uncoordinated voluntary movement). Frequently, subjects have associated cognitive impairments. The symptoms and functional levels vary significantly among subjects. In extreme cases, subjects have neither motor control nor communication due to cognitive impairment. Many others are wheelchair bounded but retaining cognitive capabilities sufficient to control adapted interfaces for communication. Other subjects have an almost normal life, despite some motor limitations.

3.4.3 Metrics to assess the ANS performance

In recent years some research effort was devoted in defining common metrics to assess human-robot interaction systems, [Steinfeld 2006, Iturrate 2009, Montesano 2010]. The primary difficulty in defining common metrics is the incredibly diverse range of human-robot applications. Thus, although metrics from other fields (HCI, human factors, etc.) can be applied to satisfy specific needs, identifying metrics that can accommodate the entire application space may not be feasible. The set of metrics to assess HRI systems proposed in [Steinfeld 2006], is an attempt to identify common metrics that can be used for evaluations across a wide range of tasks and HRI systems. This work is focused on task-oriented mobile robots, which is suitable to assess the ANS performance of a RW. The metrics are presented in terms of five task categories that can either be performed with a high-level of human intervention (pure teleoperation), or with a high-level of robot independence (full autonomy), or at any point on the interaction spectrum. The five tasks categories are: navigation, perception, management, manipulation, and social. The latter two task categories are not applicable to our ANS and therefore will no longer be considered. Additionally, an assessment of the BCI system accuracy as proposed in [Iturrate 2009, Montesano 2010] is also considered.

3.4.3.1 Assessment of navigation performance

Navigation is a fundamental task for any mobile robot. It encompasses determining where the robot is (starting pose), where it needs to be (goal pose), how it should get there (path, resource usage, etc.), and how to deal with environmental factors and contingencies (obstacles, hazards, etc.) encountered on the way. A set of effectiveness, efficiency, and workload measures were defined to assess the navigation performance in terms of three key factors: global navigation, local navigation and obstacle detection.

Effectiveness measures how well the task is completed. The following metrics were selected to assess navigation effectiveness of the ANS proposed in this thesis:

- Task success: degree of accomplishment of the navigation task $[0 - 1]$;
- Number of reached subgoals to accomplish the task.
- Path length: distance in meters traveled to accomplish the task;
- Path length optimality ratio: ratio of the path length to the optimal path (gives us a measure of deviation from planned route);
- Avoided obstacles: number of obstacles that were successfully avoided;
- Collisions: number of collisions that could be avoided;

Efficiency measures the time needed to complete the task. Efficiency measures include:

- Time: time taken in seconds to accomplish the task;
- Time optimality ratio: ratio between the time taken to accomplish the task and the optimal time (includes HRI overhead).

Finally, the workload measures include:

- BCI accuracy: accuracy of the pattern-recognition strategy (relation between correct commands and total commands);
- Number of operator interventions: number of unplanned interactions;
- Time needed for intervention;
- Ratio of operator time to robot time (for example, if the operator spends 5 minutes to input a navigation plan that allows the robot to successfully navigate for an hour, we have a 1:12 ratio [Yanco 2004]).

3.4.3.2 Assessment of perception performance

Perception is the process of making inferences about distal stimuli (objects in the environment) based on proximal stimuli (energy detected by sensors). In HRI, perceptual inference can be performed by the robot (localization, obstacle detection, etc.), by the human (e.g. certain obstacles in video images), or both, such as a robot that directs its operator's attention to a certain area of interest, but leaves inference making to the human. There are two basic tasks involved in perception: interpreting sensed data, and seeking new sensor data, therefore HRI metrics for perception are divided in two categories, particularly passive and active perception. Passive perception has to deal with the interpretation of received sensor data whereas active perception addresses the case of multiple sensor data acquisition to disambiguate or increase confidence for perceptual inference [Steinfeld 2006]. The latter category is not applicable to the ANS proposed in this thesis, and therefore will no longer be considered.

Passive perception involves interpreting sensor data: identification, judgment of extent, and judgment of motion. Identification measures detection and recognition accuracy for task objects within sensor range. This type of metrics include:

- Number of multiple-direction requests due to new obstacles detected in the environment;
- Number of avoided pedestrians;
- Local planner interventions due to localization uncertainties.

Judgment of extent measures the accuracy of quantitative judgments about the environment. Measures of this type include:

- Collisions: number of collisions;
- Obstacle clearance: minimum and mean distance to obstacles.

Judgment of motion measures the accuracy with which egomotion or movement of objects in the environment is judged. Measures of this type include:

- Speed (m/s): minimum and mean robot speed;
- Time in motion (s).

3.4.3.3 Assessment of management performance

The purpose of management performance is to coordinate and manage the actions of humans and robots, acting independently or in groups. Of primary concern is allocating and deploying resources to guarantee appropriate coverage (i.e. having the “right” agent at the “right” place at the “right” time). Performing this task requires assessing availability, understanding capabilities, team coordination, monitoring, recognizing problems, and intervention [Steinfeld 2006]. Three types of measures can be evaluated under this task category, in particular: fan out, intervention response time, and level of autonomy discrepancies. Fan out is a measure of how many similar robots can be effectively controlled by a human [Goodrich 2003]. For situations in which a robot can be operated by various humans, this measure is also an indicator of robot hands-off between operators and the upper limit of the workload for operators, as in the case of air traffic control [Rantanen 2003]. Since the ANS proposed in this thesis is a single robot, single operator, fan out metrics are not considered. Measures of intervention response time are a key metric in a HRI system. Whenever an operator does not devote total attention to a robot, there will be a delay between the instant the robot finds a situation requiring human attention, and when the human operator intervenes. Intervention response measures include:

- Time to deliver the request from the robot;
- Time for the operator to notice the request;
- Time for intervention.

The time for the operator to notice the request was not assessed in the experiments presented in this thesis. The BCI system provides a low-level command in fixed time intervals, even when the user is in an idle state (not focusing any command). Due to that reason it is not possible to evaluate when the operator has noticed the request.

Measures of level of autonomy discrepancies encompasses a metric that can evaluate the ability of the human to accurately and rapidly identify the appropriate level of autonomy. This metric encompasses several factors, such as situation awareness, trust, etc. Yet, the ANS proposed in this thesis does not allow users to select the level of autonomy, and therefore this type of metric will no longer be considered.

3.4.3.4 Assessment of online BCI experiments

To assess the performance of a BCI system, several evaluation criteria have already been proposed (see a survey in [Schlogl 2007]). A BCI system encompasses many parameters related to its underlying paradigm, which includes the amount of encoded information, the required time for selecting a symbol and its detection accuracy. The ultimate goal of a BCI is to provide an effective communication channel. Thus, the following metrics were considered to assess BCI performance:

- Number of repetitions required to attain 90% pattern recognition accuracy, during training phase;

- Trial Time, as defined by (3.1);
- BCI accuracy for combined non-decisive and decision commands: ratio of BCI correct selections to total number of selections, considering all commands selected by the user (in non-decisive and decision target points);
- BCI accuracy for decision commands: ratio of BCI correct selections to total number of selections, considering only commands selected by the user in decision target points;
- Reached waypoints: number of attained waypoints to accomplish one mission.

Chapter 4

Localization

Contents

4.1 Introduction

The localization problem is a key problem in mobile robotics. It plays a pivotal role in various successful mobile robot systems [Thrun 2001, Jensfelt 2002]. In [Cox 1991], the localization problem was described as “using sensory information to locate the robot in its environment” and it was considered as “the most fundamental problem to providing a mobile robot with autonomous capabilities.“ If a mobile robot does not know where it is, it will be difficult to decide what to do next [Rahok 2009]. Most of deliberative tasks in mobile robotics are based in the assumption that the robot is able to answer to three fundamental questions, in particular: “Where am I?”, “Where am I going?”, and ”How should I get there?” The first two questions are directly related in solving the localization problem, whereas the latter is related with planning. Briefly, mobile robot localization is the problem of estimating a robot's pose (location, orientation) relative to its environment [Thrun 2001].

The mobile robot localization problem can be addressed and classified in many different ways. In [Thrun 2001], localization methods are classified in two major groups, according to the localization problem being solved, namely: position tracking, and global localization. Position tracking is the most simple localization problem. In this case the robot knows its initial pose and only has to compensate small odometric errors occurring during robot navigation. Typically, position tracking methods are not able to recover, if they lose track of the robot's pose. The other group of methods, designated as global localization, is designed to estimate the robot's pose even under global uncertainty. In this case the robot does not know its initial pose, and a much more difficult localization problem is addressed, since robot's pose must be estimated from scratch. Techniques of this type solve the so-called wake-up robot problem, in that they can localize a robot without any prior knowledge about its pose. They furthermore can handle the kidnapped robot problem, in which a robot is carried to an arbitrary location during its operation. Global localization techniques are more powerful than position tracking. They typically can cope with situations in which the robot is likely to experience serious positioning errors [Fox 1999].

According to [Gutmann 1998, Gutmann 2002], localization techniques are classified in three basic categories: behavior-based approaches; landmarks; and dense sensor matching. Behavioral approaches rely on the interaction of robot actions with the environment in which they navigate. For example, the work of [Sagrebin 2008] describes a behavior or motion based localization algorithm that tracks all moving objects in a scene, and then selects those which are performing a previously specified motion. Another example of this type of localization is that of supplying robots. Differently looking robots which deliver packages to an office building do know to which office they have to go, but don't know how. A behavior localization system is able to localize these different looking robots in the entrance hall and navigate them to the target office. Landmark methods, on the other hand, rely on the recognition of landmarks to keep the robot localized geometrically. Landmarks may be given a priori (for example, the satellites in GPS) or learned by the robot as it maps the environment [Leonard 1992]. While landmark methods can achieve impressive geometric localization, they require either engineering the environment to provide a set of adequate landmarks, or efficient recognition of features to use as landmarks. In contrast, dense sensor methods [Diosi 2005, Lu 1994, Zhang 1992] attempt to use whatever sensor information is available to update the robot's pose. They do this by matching dense sensor scans with a surface map of the environment, without extracting landmark features. Thus, dense sensor matching can take advantage of whatever surface features are present, without having to explicitly decide what constitutes a landmark.

Simultaneous Localization and Mapping (SLAM) also known as Concurrent Mapping and Localization (CML) is the process by which a mobile robot can build a map of the environment and at the same time use this map to compute its location [Durrant-Whyte 2006]. Initially, both the map and the robot pose are not known. The robot has a known kinematic model and it moves through the unknown environment, which is populated with artificial or natural landmarks. A simultaneous estimate of both robot and landmark locations is required. The SLAM problem involves finding appropriate representation for both the measurement and the motion models, which is generally performed by computing their prior and posterior distributions using probabilistic algorithms, such as: Kalman Filters (KF) [Davison 2002, Jensfelt 2006, Se 2002], Particle Filters (PF) [Montemerlo 2002, Montemerlo 2003], or Expectation Maximization (EM) [Burgard 1999]. These techniques are mathematical derivations of the recursive Bayes rule. The main reason for this probabilistic techniques popularity is the fact that robot mapping is characterized by uncertainty and sensor noise, and probabilistic algorithms tackle the problem by explicitly modeling different sources of noise and their effects on the measurements [Thrun 2002].

4.1.1 The Bayes filter

A key concept in probabilistic localization is that of a belief. A belief reflects the robot's internal knowledge about the state of the environment. The state (pose) of the robot cannot be measured directly. Instead, the robot must infer its pose from data [Thrun 2006]. Beliefs are represented through conditional probability distributions. A belief distribution assigns a probability (or density value) to each possible hypothesis with regards to the true state. Belief distributions are posterior probabilities over state variables conditioned on the available data. A belief over a state variable x_t is denoted by $bel(x_t)$, which is an abbreviation for the posterior probability

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{4.1}$$

This posterior is the probability distribution over the state x_t at time t, conditioned on all past measurements $z_{1:t}$ and all past commands $u_{1:t}$.

The most general algorithm for calculating beliefs is given by the Bayes filter algorithm.

This algorithm calculates the belief distribution $bel(x_t)$ from measurement and command data. Algorithm 1 depicts the basic Bayes filter in pseudo-algorithmic form. The Bayes filter is recursive, that is, the belief $bel(x_t)$ at time t is calculated from the belief $bel(x_{t-1})$ at time $t-1$. Its input is the belief $bel(x_{t-1})$, along with the most recent command u_t and the most recent measurement z_t. Its output is the belief $bel(x_t)$ at time t. The Bayes filter algorithm possesses two essential steps. the first step is denoted by motion update. It consists in calculating a belief over the state x_t based on the prior belief over state x_{t-1} and the command u_t. The second step of the Bayes filter is called the measurement update. To do this, the Bayes filter algorithm multiplies the belief $\overline{bel}(x_t)$ by the probability that the measurement z_t may has been observed. The probability distribution $p(z_t|x_t)$ is the sensor measurement model, denoting the probability distribution over image z_t at x_t, and η is a normalization factor ensuring that total probability mass over x_t is 1. The Bayes filter makes a Markov assumption that specifies that the state is a complete summary of the past. Under this assumption the belief can be considered sufficient to represent the past history of the robot.

Algorithm 1 BayesFilter($bel(x_{t-1}), u_t, z_t$)

```
for all $x_t$ do
    //Motion update
    $\overline{bel}(x_t) \leftarrow \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx$
    //Measurement update
    $bel(x_t) \leftarrow \eta p(z_t|x_t)\overline{bel}(x_t)$
end for
return $bel(x_t)$
```

4.1.1.1 Nonparametric filters

An alternative to Gaussian techniques are nonparametric filters. Nonparametric filters do not rely on a fixed functional form of the posterior, such as Gaussians. Instead, they approximate posteriors by a finite number of values, each roughly corresponding to a region in state space. Some nonparametric Bayes filters rely on a decomposition of the state space, in which each value corresponds to the cumulative probability of the posterior density in a compact subregion of the state space, this is the case of the histogram filters.

$$range(X_t) = x_{1,t} \cup x_{2,t} \cup ... \cup x_{K,t} \tag{4.2}$$

Here X_t is the familiar random variable describing the state of the robot at time t. The function $range(X_t)$ denotes the state space, that is, the universe of possible values that X_t might assume. Each $x_{k,t}$ describes a convex region. These regions together form a partitioning of the state space, that is, for each $i \neq k$ we have $x_{i,t} \cap x_{k,t} = \emptyset$ and $\bigcup_k x_{k,t} = range(X_t)$.

Others nonparametric filters approximate the state space by random samples drawn from the posterior distribution, which is the case of particle filters. In all cases, the number of parameters used to approximate the posterior can be varied. The quality of the approximation depends on the number of parameters used to represent the posterior [Thrun 2006]. Algorithm 2 is derived from the general Bayes filter presented in Algorithm 1, by replacing the integration with a finite sum. In the motion update, the belief is calculated for the new state based on the command, u_t, alone. This state estimation is then updated taking measurement z_t into account. The discrete Bayes filter algorithm is popular in many areas of signal processing, where it is often referred to as the forward pass of a hidden Markov model.

Algorithm 2 DiscreteBayesFilter($p_{k,t-1}, u_t, z_t$)

```
for all $k$ do
    //Motion update
    $\overline{p}_{k,t} \leftarrow \sum_i p(X_t = x_k|u_t, X_{t-1} = x_i)p_{i,t-1}$
    //Measurement update
    $p_{k,t} \leftarrow \eta p(z_t|X_t = x_k)\overline{p}_{k,t})$
end for
return $p_{k,t}$
```

The particle filter is an alternative nonparametric implementation of the Bayes filter. Particle filters also approximate the posterior by a finite number of parameters. In this case, the posterior $bel(x_t)$ is represented by a set of random state samples drawn from this posterior. In particle filters, the samples of a posterior distribution are called particles and are denoted by

$$\chi_t := x_t^{[1]}, x_t^{[2]}, ..., x_t^{[M]} \tag{4.3}$$

Each particle $x_t^{[m]}$ (with $1 \leq m \leq M$) is a concrete instantiation of the state at time t, that is, an hypothesis of how the true state may be at time t. Here M denotes the number of particles of the particle set χ_t. Algorithm 3 shows the basic form of the particle filter algorithm. For a particle filter, beliefs are represented by sets of particles and χ_t is recursively constructed from χ_{t-1}. The input of algorithm 3 is the particle set χ_{t-1}, along

Algorithm 3 ParticleFilter(χ_{t-1}, u_t, z_t)

```
$\overline{\chi}_t \leftarrow \chi_t = \emptyset$
**for all** $m = 1 : M$ **do**
    //Sampling
    *sample* $x_t^{[m]} \approx p(x_t|u_t, x_{t-1}^{[m]})$
    //Importance factor
    $\omega_t^{[m]} \leftarrow p(z_t|x_t^{[m]})$
    $\overline{\chi}_t \leftarrow \overline{\chi}_t + \langle x_t^{[m]}, \omega_t^{[m]} \rangle$
**end for**
**for all** $m$ **do**
    //Resampling
    *draw m with probability* $\propto \omega_t^{[m]}$
    *add* $x_t^{[m]}$ *to* $\chi_t$
**end for**
return $\chi_t$
```

with the most recent command u_t and the most recent measurement z_t. The algorithm first constructs a temporary particle set χ_t which is reminiscent (but not equivalent) to the belief $bel(x_t)$. It does this by systematically processing each particle $x_t^{[m]}$ in the input particle set χ_{t-1}. This is carried out in three main steps:

Sampling: a hypothetical state $x_t^{[m]}$ for time t based on the particle $x_{t-1}^{[m]}$ and the command u_t, is generated. This step involves sampling from the next state distribution $p(x_t|u_t, x_{t-1}^{[m]})$.

Importance factor: for each particle $x_t^{[m]}$ the so-called importance factor, denoted $\omega_t^{[m]}$ is derived. Importance factors are used to incorporate the measurement z_t into the particle set.

Resampling: resampling or importance resampling consists in the replacement of M particles from the temporary set $\overline{\chi}_t$. The probability of drawing each particle is given by its importance weight. Resampling transforms a particle set of M particles into another particle set of the same size. By incorporating the importance weights into the resampling process, the distribution of the particles change: whereas before the resampling step, they were distributed according to $\overline{bel}(x_t)$, after the resampling they are distributed (approximately) according to the posterior $bel(x_t) = \eta p(z_t|x_t^{[m]})\overline{bel}(x_t)$. In fact, the resulting sample set usually possesses many duplicates, since particles are drawn with replacement. Particles that are not contained in χ_t tend to be particles with lower importance weight.

4.1.2 Probabilistic localization methods

Bayesian filters are a general concept for recursive state estimation. The belief distribution (4.1) is the key part of the update equation in a Bayes filter, and varies depending on the used approach. Distributions can be continuous (Kalman filter and Multi-Hypothesis Tracking (MHT)) or discrete (Grid Markov Localization (GML) and Monte Carlo Localization (MCL)). Table 4.1 presents a comparison of the probabilistic localization methods.

4.1.2.1 The Kalman filter

Kalman filters are one of the most widely used variants of Bayesian filters, such as in solving the pose tracking problem. The belief distribution is approximated only by the first and the second moment, what makes the Gaussian distribution an optimal choice (since all moments higher than two are equal 0). Though Kalman filters are very efficient, they can represent only unimodal beliefs with limited initial uncertainty. However, in practice they work fine even in highly nonlinear systems, despite underlying linear assumptions about system's dynamics.

4.1.2.2 Multi-hypothesis tracking

In multi-hypothesis tracking the belief is represented by a linear combination of unimodal Gaussian distributions. MHT tracks each single Gaussian hypothesis about possible robot poses (represented by weights), overcoming in this way the main limitation of the Kalman filter. Unfortunately, this algorithm needs sophisticated techniques and heuristics to solve sensor data association problem and to decide when to add or remove hypothesis.

4.1.2.3 Grid-based approaches - Markov localization

In grid-based approaches the state space is divided into regular network of adjacent cells, and the belief is represented by a value assigned to each cell. This allows modeling arbitrary distributions, and as a result, solving the global localization problem. Grid approaches are also relatively accurate and very robust against sensor noise. The obvious disadvantage of the grid approaches is the space and computational complexity which grows exponentially with the number of space dimensions, what may prevent from whichever applications of the algorithm in higher dimensional spaces.

GML approximates the posterior using a discrete Bayes filter over a grid decomposition of the pose space. The discrete Bayes filter maintains as the posterior distribution, $p_{k,t} =$

$bel(x_t)$ a collection of discrete probability values

$$p_{k,t} = \sum_i p(X_t = x_k | u_t, X_{t-1} = x_i) p_{i,t-1} \tag{4.4}$$

where each probability $p_{k,t}$ is defined over a grid cell x_k. The variables x_i and x_k denote individual states, of which there can only be a finite number, and $p_{i,t-1}$ is the prior probability. The set of all grid cells forms a fixed partition of the space of all legitimate poses 4.2.

4.1.2.4 Topological-based approaches

The problem with the exponentially growing complexity in the grid-based approaches can be avoided by using non-metric representation of space, through the use of graphs. Each node of the graph represents a certain location in the environment, while the edges are physical connections between them. Whereas the topological approaches can be quite efficient in terms of computational requirements, they are not very precise about space locations, as the number of state spaces is limited. A technique called Fast Appearance Based Mapping, also designated as FAB-MAP [Cummins 2008, Paul 2010] can be referred as an example of a topological approach. This technique does not require on keeping track of the robot in a metric coordinate system, and is applicable for sparse and discrete observations. This method may be used to determine the localization of new robots that are being integrated in a certain world of which they do not have any prior information.

4.1.2.5 Monte Carlo localization

The MCL algorithms represent a robot's belief by a set of weighted hypotheses (samples),

$$bel(x_t) = \{x_t^{[m]}, \omega_t^{[m]}\}, m = 1, ..., M \tag{4.5}$$

where $x_t^{[m]}$ is the pose, and $\omega_t^{[m]}$ is the weight, or importance factor, associated to each particle m that specifies the quality of that specific particle. The basis of the MCL is to construct a sample-based representation of the Probability Distribution Function (PDF). The MCL is recursive in nature and operates in two stages: motion update and measurement update. In the motion update stage, each particle is modified according to the existing motion model. In the second stage, each particle weight is re-evaluated according to the latest sensory information available (measurement update). Then, the particles with smaller weights are eliminated in a process called resampling. The MCL

Table 4.1: Comparison of different localization approaches.

Approach	Main features	Related works
Kalman filter	Unimodal; Gaussian distribution ; first and second moment of PDF; (non)linear dynamics and observations; polynomial complexity in state dimension; position tracking problem; good accuracy and efficiency.	[Durrant-Whyte 1997, Dissanayake 2002]
MHT	Multimodal; Gaussian distribution; (non)linear dynamics and observations; polynomial complexity in state dimension; global localization problem; good accuracy and robustness; difficult to implement, and computationally expensive.	[Reid 1979, Cox 1996]
Grid-based (GML)	Discrete; piecewise constant approximation; non-linear dynamics and observations exponential complexity in state dimension; global localization problem; good accuracy and robustness to sensor noise.	[Fox 1999]
Particle filter (MCL)	Discrete; sample-based approximation; non-linear dynamics and observations; exponential complexity in state dimension; global localization problem; good accuracy and robustness to sensor noise.	[Thrun 2001]
Topological	Discrete; abstract state-space; abstract dynamics and observations; one-dimensional graph; global localization problem; poor accuracy.	[Cummins 2008]

algorithm poses several advantages. It is able to represent multi-modal distributions, and thus localize the robot globally. It is relatively easy to implement, and it has the potential to considerably reduce the amount of memory required, which enables it to integrate measurements at higher rates. The accuracy-computational costs trade-off is achieved through the size of the particle set.

4.2 GML-based proposed approach

The localization approach proposed in this section is performed using odometry for rough positioning, and laser data for polar scan matching. Similar approaches can be found in [Fox 1999, Thrun 2006], but in those cases different map-matching approaches were applied. Grid Markov localization maintains as posterior a set of discrete probability values $p_{ijk,t}$ that are defined over each grid cell x_{ij}. In the implementation proposed in this paper, the partitioning of the space of all poses is time-invariant, for which each grid cell is of the same size (each grid cell has a fixed dimension of $(5 \times 5)\ cm^2$). The localization approach is presented in Algorithm 4. It is composed by two stages: motion model update and measurement update. The motion model update uses the differential motion model to compute $\bar{x}_t$ from x_{t-1} and command signal u_t. In the measurement update, a polar scan matching algorithm (see Algorithm 6 in Subsection 4.2.3) is used to weight each state $\bar{x}_{ijk,t}$ with the likelihood of observing the current scan measurement r_t, taking the map $\mathscr{M}$ into account. The function $DefineLocalGrid(\bar{x}_t)$ constructs a local grid of (i, j) cells centered on the pose $\bar{x}_t$. Associated to each grid cell, there is a set of poses $\bar{x}_{ijk,t}$ with position $x_{ij,t}$ defined by the center of the cell, and 21 orientations defined by the set $\{-10°, -9°...0°... + 9°, +10°\}$.

Algorithm 4 MarkovLocalization($x_{t-1}, u_t, \mathscr{M}, r_t$)

```
//Motion model update
$\bar{x}_t \leftarrow DifferentialMotionModel(x_{t-1}, u_t)$
$DefineLocalGrid(\bar{x}_t)$
//Measurement update
//for all local grid cell centers $x_{ij,t}$
for all $i, j$ do
  //for the 21 orientations $x_{ijk,t}$
  for all $k$ do
    $p_{ijk,t} \leftarrow \eta PolarScanMatching(\bar{x}_{ijk,t}, r_t, \mathscr{M})$
  end for
end for
return($x_t$ with $max(p_{ijk,t})$)
```

4.2.1 Scan preprocessing

To improve the performance of the Markov localization system, two filters were applied before scan matching: a distance filter, $Dist(r_t)$, and a dynamic object filter, $DynamicObj(r_t)$. The former reduces the maximum range to a predefined threshold, and the latter removes range data resulting from dynamic objects, which are not represented in the a priori map. With the dynamic object filter, range readings are segmented. Small segments are then ignored because they are likely to belong to dynamic objects in the environment such as people, table and chair legs. Segmentation was based on a very simple rule, for which a range reading is in the same segment as its previous neighbor, if they are closer than a threshold [Diosi 2005].

4.2.2 Virtual scan

An important step of scan matching is to determine a virtual scan, $\hat{r}_t$, to be compared to the current scan, r_t. Taking the map $\mathscr{M}$ as an input, we need to find out how the scan would look like, if it was taken from a pose estimation $\bar{x}_{ijk,t}$. Algorithm 5 shows the $VirtualScan(\bar{x}_{ijk,t}, \mathscr{M})$ algorithm used to determine the virtual scan from pose estimation $\bar{x}_{ijk,t}$. The constant N represents the number of scan points, $MAXRANGE$ is the maximum range distance, and Δ is a constant value to guarantee that all cells of the grid map in the direction of $\theta_{bg}(n) + \bar{\theta}_{ijk,t}$ ($n = \{1, ..., N\}$ represents the index of the scan sector), are evaluated in terms of occupation. If a cell with position (X_r, Y_r) is occupied, i.e. $m(X_r, Y_r) = 1$, then the range of the virtual scan in the direction $\theta_{bg}(n) + \bar{\theta}_{ijk,t}$ has been found. Figure 4.1 illustrates how each virtual scan sector is constructed. To obtain a more reliable virtual scan, we should consider the effects of small measurement noise, errors due to unexpected objects, errors due to failures to detect objects, and random unexplained noise. In [Thrun 2006], the measurement is modeled as a mixture of densities, each of which corresponds to a particular type of the mentioned errors. However, since we carry out the preprocessing of the current scan prior to map matching, the errors due to unexpected objects, failures, and random noise are neglected. Therefore, we have only considered the effects of small measurement noise in Algorithm 5. Considering $\hat{r}^*_t(n)$ as the range reading for an ideal sensor, the noise was modeled as a normal distribution with mean $0.98\hat{r}^*_t(n)$, which reflects a measurement accuracy of 2%, and standard deviation $\sigma = 2.9\ mm$, as proposed in [Kneip 2009] for white surfaces.

Algorithm 5 $VirtualScan(\bar{x}_{ijk,t}, \mathscr{M})$

```
for all $n = 1 : N$ do
  for all $l = \Delta : -1 : 0$ do
    $r_{nl} \leftarrow MAXRANGE - l * (MAXRANGE/\Delta)$
    $X_r \leftarrow \bar{X}_{ijk,t} + (r_{nl}) * cos(\theta_{bg}(n) + \bar{\theta}_{ijk,t})$
    $Y_r \leftarrow \bar{Y}_{ijk,t} + (r_{nl}) * sin(\theta_{bg}(n) + \bar{\theta}_{ijk,t})$
    if $m(X_r, Y_r)$ is occupied then
      $\hat{r}^*_t(n) \leftarrow r_{nl}$
      $//Randn(1)$ generates random numbers in the interval $[-1, 1]$
      $\hat{r}_t(n) \leftarrow (0.98\hat{r}^*_t(n)) + \sigma * Randn(1)$
      break;
    end if
  end for
end for
return($\hat{r}_t$)
```

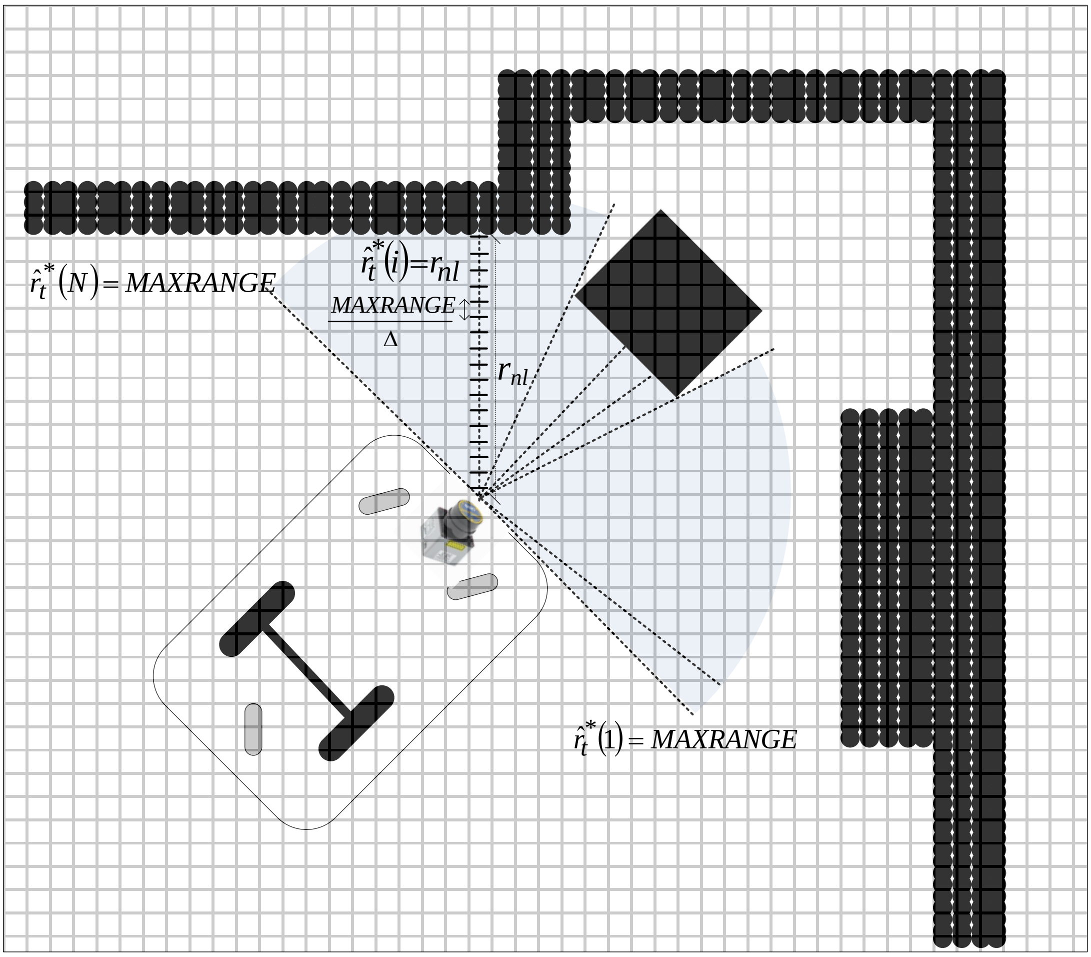

$\hat{r}_t^*(N) = MAXRANGE$
$\hat{r}_t^*(i) = r_{nl}$
$\frac{MAXRANGE}{\Delta}$
r_{nl}
$\hat{r}_t^*(1) = MAXRANGE$

Figure 4.1: Ilustration of virtual scan construction.

4.2.3 Polar scan matching

The laser scan matching method compares the current scan r_t, given by a laser scanner, with the virtual scan $\hat{r}_t$. The latter is determined based on the grid map $\mathscr{M}$, such that the more similar r_t and $\hat{r}_t$ are, the larger $p(r_t|x_{ijk,t}, \mathscr{M})$ is. We used the sample Pearson correlation coefficient $r_{r_t,\hat{r}_t}$ to evaluate the similarity of both current and virtual scans. We obtained $r_{r_t,\hat{r}_t}$ by using sample covariances and variances, as follows:

$$r_{\hat{r}_t,r_t} = \frac{\sum_{i=1}^{N}(\hat{r}_t(i) - \bar{r})(r_t(i) - \bar{r})}{\sqrt{\sum_{i=1}^{N}(\hat{r}_t(i) - \bar{r})^2 \sum_{i=1}^{N}(r_t(i) - \bar{r})^2}} \tag{4.6}$$

where N is the number of scan points, and $\bar{r}$ is an average range value, given by

$$\bar{r} = \frac{1}{2N} \sum_{i=1}^{N} (\hat{r}_t(i) + r_t(i)) \tag{4.7}$$

Laser scan matching considers

$$p(r_t|\bar{x}_{ijk,t}, \mathscr{M}) = max(r_{\hat{r}_t, r_t}, 0) \tag{4.8}$$

as the probability of the current scan conditioned by the map $\mathscr{M}$ and position $\bar{x}_{ijk,t}$. Algorithm 6 presents the polar scan matching algorithm, where r'_t represents the current scan after applying a distance filter, and r''_t represents the current scan after applying a distance filter and dynamic object filter. The constant N'' denotes the number of points in r''_t.

Algorithm 6 PolarScanMatching($\bar{x}_{ijk,t}, r_t, \mathscr{M}$)

```
//Scan preprocessing
$r'_t \leftarrow Dist(r_t)$
$r''_t \leftarrow DynamicObj(r'_t)$
//Virtual Scan and preprocessing
$\hat{r}_t \leftarrow VirtualScan(\bar{x}_{ijk,t}, \mathscr{M})$
$\hat{r}'_t \leftarrow Dist(\hat{r}_t)$
//Polar scan matching
for all points $\in r''_t$ do
    $\bar{r} = \frac{1}{2N''} \sum_{i=1}^{N''} (\hat{r}'_t + r''_t)$
end for
for all points $\in r''_t$ do
    $r_{\hat{r}'_t, r''_t} = \frac{\sum_{i=1}^{N''} (\hat{r}'_t(i) - \bar{r})(r''_t(i) - \bar{r})}{\sqrt{\sum_{i=1}^{N''} (\hat{r}'_t(i) - \bar{r})^2 \sum_{i=1}^{N''} (r''_t(i) - \bar{r})^2}}$
end for
$p(r_t|\bar{x}_{ijk,t}, \mathscr{M}) = max(r_{\hat{r}'_t, r''_t}, 0)$
return($p(r_t|x_{ijk,t}, \mathscr{M})$)
```

4.2.4 Computational complexity

When using a fine-grained grid, the construction of occupancy maps may become a costly operation. Additionally, searching for the robot pose that maximizes the correlation between a measurement (sensor reading) and a map-based virtual measurement may be not possible to perform on real-time, depending on the grid resolution, and map dimensions. Therefore, three techniques to reduce the computational complexity of grid localization, are proposed:

1. Grid sampling, which consists of selecting a local grid, instead of considering the entire grid map to perform measurement and map correlation. The local grid has a dimension of $(30 \times 30)\ cm^2$.

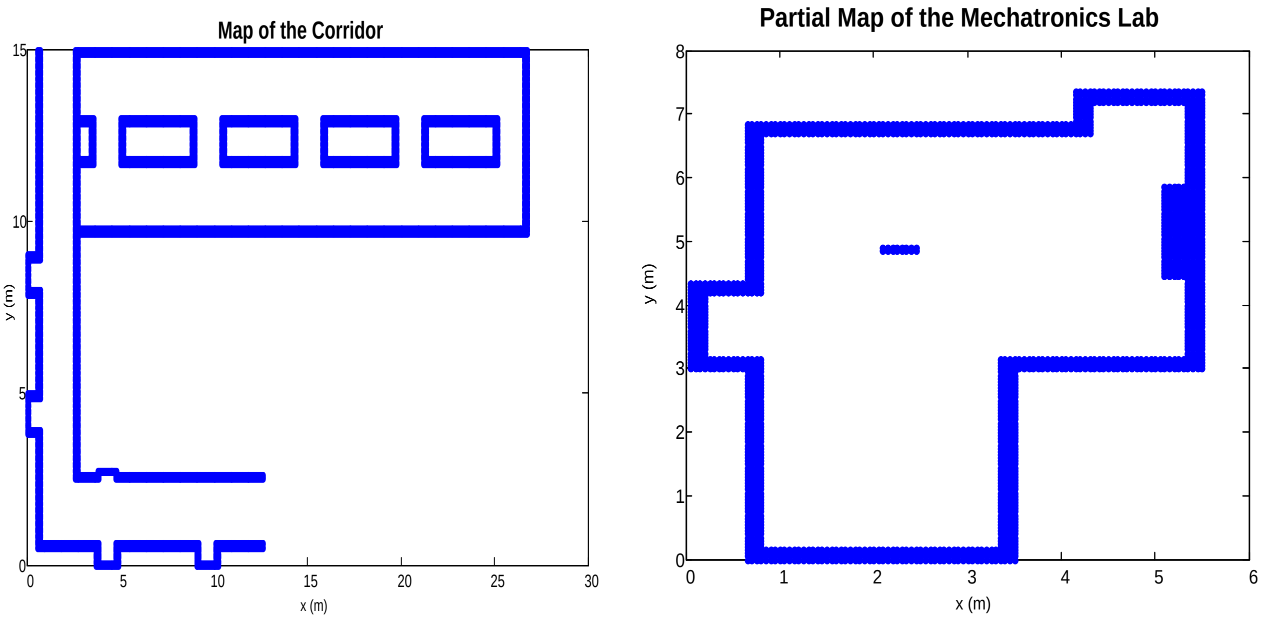
2. Orientation sampling, which was obtained by applying the polar scan matching algorithm to a subset of possible orientations. In our case we have considered that the orientation error estimation was below 0.17 rad (10°). The orientation search resolution was 0.017 rad;

3. The measurement update is applied whenever the RW travels approximately 30 cm, instead of being applied every time a new odometry value is calculated.

The map matching approach proposed in [Thrun 2006] operates in a Cartesian coordinate frame and therefore does not take advantage of the native polar coordinate system of a laser scan. In our approach, we propose to evaluate the correlation between a scan measurement, and a virtual scan, which is pre-calculated, using the predicted scan technique, for the several candidate poses associated to a grid submap obtained using the grid sampling technique. The computation complexity of our approach is mitigated due to two main factors: 1) we do not need to construct an occupancy grid map based on the measurement scan, but instead we simply use rough laser scan data; 2) working with polar coordinates instead of Cartesian coordinates reduces computational complexity from $O(n^2)$ to $O(n)$.

4.2.5 Experimental results

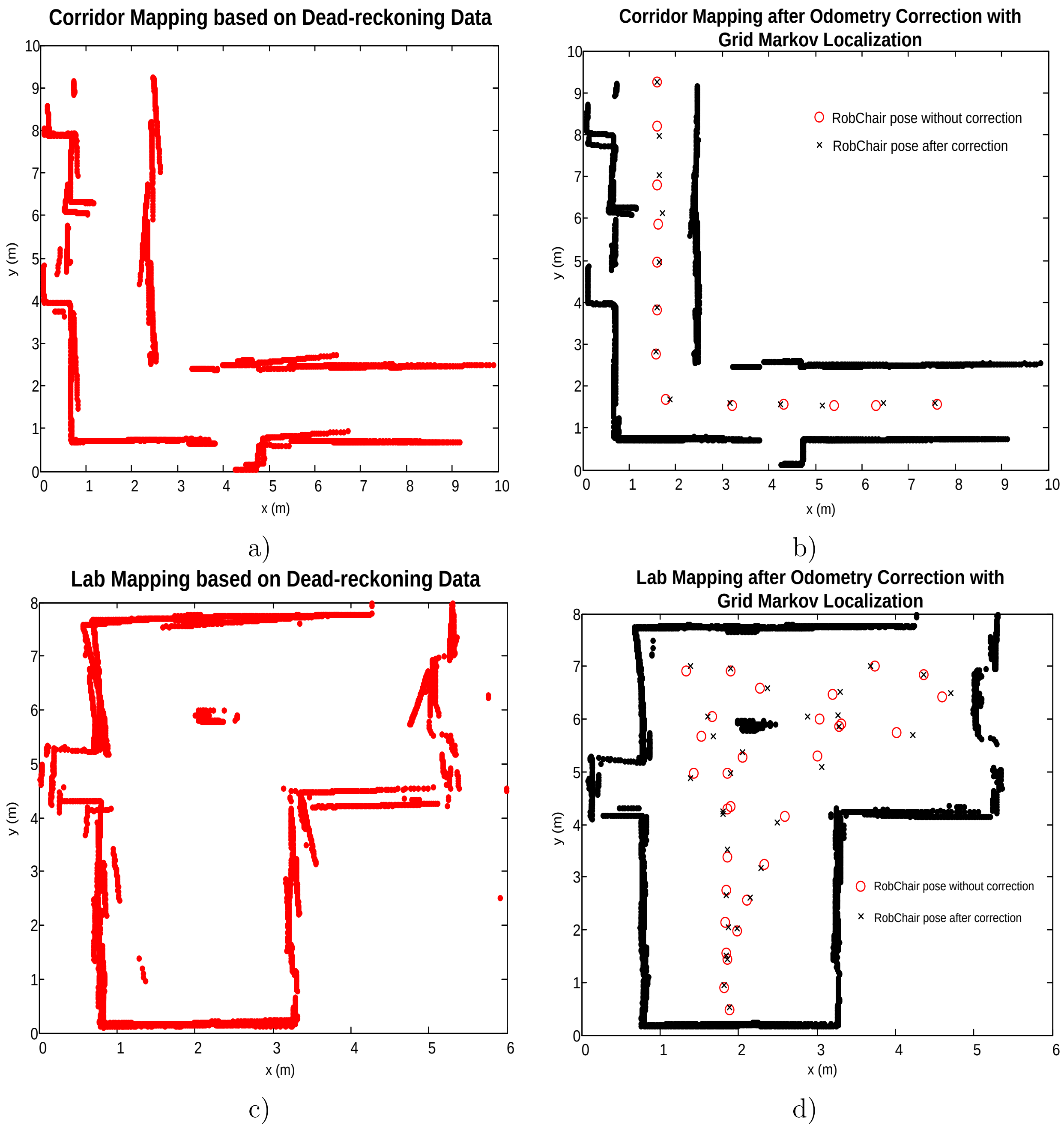
To evaluate the grid Markov localization algorithm, two datasets in two different scenarios were generated. The data was collected with RobChair, and it includes the poses determined with odometry data, and the laser scan readings. Figure 4.2 shows the maps of the two scenarios where data was collected, more precisely, a corridor (see Fig. 4.2 a)) and a partial view of the mechatronics lab at ISR-UC (Fig. 4.2 b)). Figure 4.3 shows the mapping results for both scenarios. Figure 4.3 a) and Fig. 4.3 c) show map reconstruction based on dead-reckoning data, whereas Fig. 4.3 b) and Fig. 4.3 d) show map reconstruction after pose correction using the grid Markov localization algorithm.



a)

b)

Figure 4.2: Maps of two scenarios used to generate datasets for localization assessment: a) map of a corridor at ISR-UC: b) partial map of mechatronics lab at ISR-UC.



a)

b)

c)

d)

Figure 4.3: Map reconstruction for both scenarios: a) corridor mapping before odometry correction: b) corridor mapping after odometry correction; c) lab mapping before odometry correction; d) lab mapping after odometry correction.

Chapter 5

Assistive navigation training framework

Contents

5.1 Introduction

The ANTF was designed to train and characterize users that are severe motor impaired to steer a RW. The use of conventional HMIs to steer a RW, such as a conventional joystick, presents itself as a serious barrier to these users, preventing them to reach a higher mobility level. The choice of a proper HMI, adaptable to users' limited motor capabilities becomes, in this sense, a decisive factor, which, in most cases, consists in HMIs only capable to provide a small set of commands, issued sparsely, such as the P300-based BCI described in section 3.2 that provides a small set of brain-actuated commands. With the ANTF we are pursuing the following goals: to train users to steer an RW autonomously, to train users operating the selected HMI, in this case a P300-based BCI, and to characterize the human user in both, steering the RW and operating the HMI. The user characterization is later used to adapt the RW navigation system to his/her steering capabilities.

The training approach is based on judgment theory, more specifically on Rule-Based Lens (RBL) paradigm, and is able to provide the characterization of individual performance in order to tailor training to the needs of each human user. Individual judgment performance is modeled using a Genetic-Based Policy Capturing (GBPC) technique characterized to infer noncompensatory judgment strategies from human decision data. The ANTF was designed to assist users in steering a RW in an indoor structured environment, and to characterize their performance, choosing an appropriate maneuver from a set of possible maneuvers needed to perform the navigation tasks.

5.2 Background on judgment theory

Social judgment theory describes the implications of fallible judgment for people working together as well as for social policy. Following Brunswik, social judgment theory emphasizes the importance of the task in shaping judgment. The task is both the context for judgment and the context for learning to make judgments. Careful study of the task and understanding how task properties affected judgment is critical for explaining and improving judgmental performance.

Judgment analysis is a method for making a person's judgment strategy explicit. Judgment analysis begins with judgments about each case in a set of cases. Each case is described by the values of several variables, or cues. Cases can be real (e.g., patients judged in a clinical setting) or hypothetical. If the cases are hypothetical, they and the judgment made must be representative. Representative design means that the conditions that the researcher wants to generalize to must be specified, and those conditions must be adequately represented in the experimental task so that the desired generalizations can be supported. A linear multiple regression can provide a reasonably good fit for the judgments of an individual in a variety of contexts. Judgments of an individual are described by identifying the relative weights for the cues, the shape of the function relating each cue and the judgment (e.g., linear or nonlinear), and the principle by which multiple cues are organized into a judgment.

Judgment theory is often used to model the judgment of a human selecting a criterion value based on a set of probabilistic cues. Brunswik's Lens Model (BLM) and its extensions [Brunswik 1955, Hammond 1975] are examples of the application of linear models to describe judgment behavior. The BLM provides dual, symmetric models of both the human judge and the environment. Since those models are based on the same environmental information (the cues), the fit between the model of the human judge and the

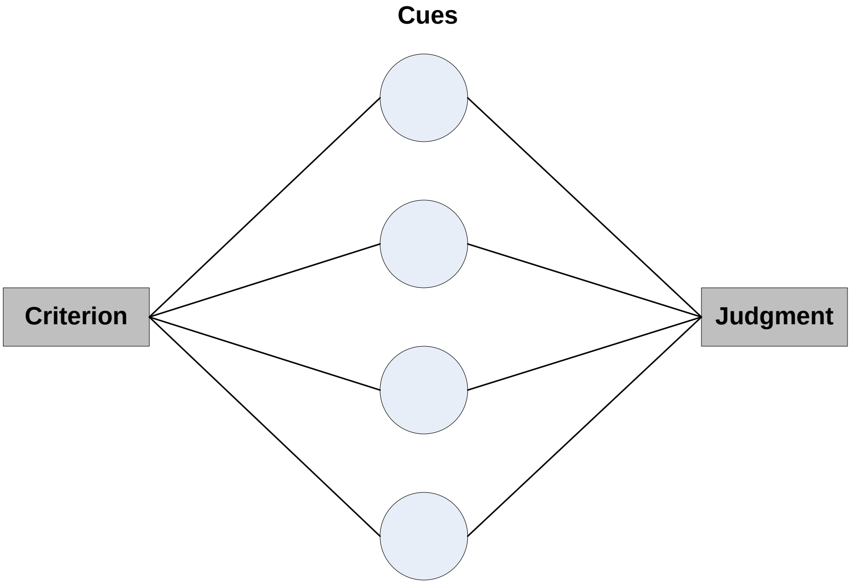


Figure 5.1: Brunswik's lens model.

environmental structure can be formally measured. The BLM [Brunswik 1955] is broadly applicable, however, most of its existing analytical models focus primarily on compensatory judgment behavior. Rothrock and Kirlik [Rothrock 2003] proposed a technique called GBPC to infer noncompensatory judgment strategies from human decision data, which explores rule-based modeling using propositional logic. They focused on the development of an inductive inference technique to capture judgment policies. In [Yin 2006] a rule-based analytical model of the conceptual BLM was developed. The role of the GBPC within the RBL is to generate environment and human models, respectively.

5.2.1 The lens model

The lens model is a method to model the judgment of a human (e.g. a weather forecaster) who selects a criterion value (e.g. if it will rain or not) based on a set of probabilistic cues (e.g. air fronts, dew point temperature and temperature). The lens model, depicted in Fig. 5.1, represents the decision-making system as a symmetrical structure. The task environment, or ecology, is represented in the left half of the figure and the human judge is represented on the right half. The symmetry inherent in this representation allows one to measure the degree of adaptation or "fit" between the judge and the demands of the judgment task.

The conceptual model framed by [Brunswik 1955] is broadly applicable, existing analytical models of the lens model [Hammond 1975] focus primarily on compensatory judgment behavior. Under a compensatory mode of decision making, information is processed exhaustively and trade-offs need to be made between attribute values.

5.2.1.1 Mathematical formulation of the lens model

The standard representation of the lens model is given in Fig. 5.2 where subjects make judgments, denoted as Y_s, on an environmental variable measured by the criterion, denoted as Y_e, by using a set of cues X. The predicted values of the environmental criterion and human judgments, $\hat{Y}_e$ and $\hat{Y}_s$, respectively, are generated by corresponding linear regression equations. Lens model parameters ra, R_e, R_s, G, C are calculated as Pearson's correlation coefficient, r, such that

$$R_e = r(Y_e, \hat{Y}_e) \tag{5.1}$$

$$R_s = r(Y_s, \hat{Y}_s) \tag{5.2}$$

$$G = r(Y_e, Y_s) \tag{5.3}$$

$$C = r[(Y_e - \hat{Y}_e), (Y_s - \hat{Y}_s)] \tag{5.4}$$

$$ra = GR_eR_s\sqrt{1-(R_e)^2}\sqrt{1-(R_s)^2} \tag{5.5}$$

where the achievement, ra, represents how well human decisions adapt to the actual value of the environmental criterion. Parameter R_e represents the environmental predictability and measures how well the environmental model predicts the actual criterion value. Parameter R_s is labeled as human control and indicates how the linear policy model captures the actual human judgments. Linear knowledge, denoted as G, is designed to estimate how well the linear prediction model of the environment maps onto the predicted policy model of the human judgment. Unmodeled knowledge, denoted as C, measures how well the two models (one for the environment and the other for the human judgment) share the common points that are not captured in the corresponding linearly based model.

5.2.1.2 Properties of the compensatory lens model approach

According to [Rothrock 2003], the compensatory lens model approach presents some limitations. The regression assumes that the judge has available a set of cues which he/she is able to measure. This measurement can either be binary (i.e., a cue is either absent or present), or defined in terms of the magnitude of a cue value. In addition, the judge is assumed to use some form of cue weighting policy, which is correspondingly modeled

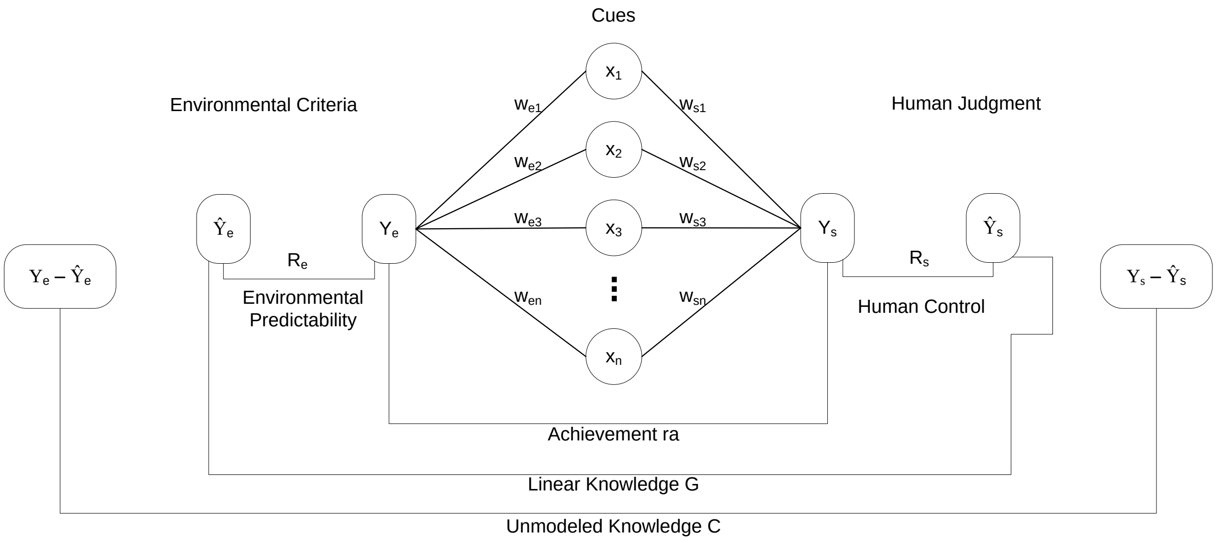


Figure 5.2: Compensatory lens model with associated statistical parameters.

by the set of weights resulting from the regression model that best fits the judge's behavioral data. Moreover, a regression model assumes that the judge integrates (the possibly differently weighted) cue values into a summary judgment. This type of weighting and summing judgment process, as represented by a linear-additive rule, has an important property: it reflects a compensatory strategy for integrating cue information. These strategies are compensatory in the sense that the presence of a cue with a high value, or high positive weighting can compensate for an absence of cues with moderate or low weighting. Similarly, cues with high negative weights compensate for cues with high positive weights, reflecting the manner in which a person might weigh, or trade off, evidence for and against a particular judgment.

5.2.2 The rule-based lens model

The rule-based lens model (RLM) approach is intended to model the human judgment of a probabilistic criterion using rule-based relationships. The works of [Rothrock 2003, Yin 2006] present an extension of the lens model proposed by [Brunswik 1955], so that contingent decision behavior can be incorporated. Research on contingent decision making suggests that decision makers respond adaptively to the variations in the environment through use of different strategies [Yin 2006]. Research has shown that decision makers are sensitive to the increase of problem size and generally shift to noncompensatory evaluation strategies that save effort but are less accurate [Payne 1993]. Time pressure is another reason that the use of a compensatory strategy may be less preferable [Simon 1996].

However, when the task environment does not pose severe constraints in terms of time and task load, people tend to use compensatory strategies so that high decision accuracy can be obtained [Payne 1993].

5.2.2.1 Properties of the noncompensatory lens model approach

A noncompensatory strategy is one in which the "trading off" property is absent. In [Einhorn 1974] two noncompensatory judgment rules are discussed: a conjunctive rule and a disjunctive rule. A conjunctive rule describes a strategy in which every cue considered in the judgment must have a high value (or exceed some threshold) in order for the overall judgment to have a high value. People being evaluated on their job performance often complain when it appears they are being assessed by conjunctive strategy, noting that they will not receive a high evaluation or job promotion unless they perform at a high level on every dimension of the evaluation. Note the noncompensatory nature of this strategy: no cue value, however highly weighted, can compensate for a low value on any one of the other cues.

The second type of noncompensatory strategy discussed by [Einhorn 1974] is a disjunctive rule. A disjunctive strategy is one in which only one cue must have a high value, or exceed some threshold, in order for the overall judgment to have a high value. A good example of a disjunctive strategy might be the evaluation of athletes in a professional (U.S.) football draft: a player might be highly rated if he has high value on any within the set of relevant, evaluative dimensions (e.g., speed, placekicking ability, punting ability, passing ability, etc.). Note that this strategy is noncompensatory, in the sense that a low value on a particular cue or set of cues does not detract from an overall high rating, given the presence of at least one cue with high value [Rothrock 2003]. Other types of noncompensatory strategies may include:

- Elimination by aspects: at each stage of the process, the decision maker selects an attribute and eliminates all alternatives that do not include a specified aspect (i.e., attribute value) until only one alternative remains;

- Take the best: the decision maker uses a sequence of rules to choose an alternative in the face of uncertain knowledge.

5.2.2.2 Genetic-based policy capture

The works of [Rothrock 2003, Yin 2006] proposed a technique called GBPC to infer noncompensatory judgment strategies from human decision data. The role of GBPC within the RLM is to generate Y_e and Y_s. Performers in time-stressed, information-rich environments, develop heuristic task-simplification strategies. Judgment strategies in these environments may have a noncompensatory nature, which may be adaptive to the time stressed nature of these tasks, since such heuristics typically make lower demands for information search and integration than do corresponding, linear-additive, compensatory strategies as proposed by standard representation of the Brunswik Lens Model [Brunswik 1955, Hammond 1975]. As a result, linear regression may be inappropriate for inferring the judgment strategies used by users in time-stressed environments. The GBPC technique infers noncompensatory judgment strategies under the assumption that these strategies can be described as a disjunctive collection of conjunctive rules. The fitness measure embodied in GBPC evaluates candidate rule sets on the three following dimensions [Rothrock 2003]:

1. Completeness: the inferred rule base is consistent with all user judgments;

2. Specificity: the rule base is maximally concrete;

3. Parsimony: the rule base contains no unnecessary rules.

5.2.2.3 Formulation of the RLM

There is a key difference between in building the functional relationship between human judgment and the task environment for compensatory and rule-based strategies, since the prediction mechanisms for compensatory strategies are multiple linear regression models whereas the mechanisms for rule-based strategies are logical propositions. To create the RLM, a framework to compare the results of fitted models of both judgment strategies $(Y_s - \hat{Y}_s)$ and the task ecology $(Y_e - \hat{Y}_e)$ has to be performed. Whereas compensatory model error is measured using the least-squares estimator, the rule-based model error is measured as the number of mismatches between two sets of categorical data. Assuming the ecological criterion to be judged is Y, let $\{Y_1, Y_2, ..., Y_p\}$ be the set of discrete values that Y can take. For instance i, $1 \leq i \leq n$, where n is the number of judgments. Let Y_{ei} represents the actual environmental criterion value; $\hat{Y}_{ei}$ represents the predicted criterion value from the environmental rule-based model; Y_{si} represents the human judgment value and $\hat{Y}_{si}$ represents the predicted judgment value from the judgment rule-based model. The RLM relationships are defined as follows:

$$R_e = \frac{\sum_{i=1}^{n} I_{ei}}{n} \quad , \quad I_{ei} = \left\{ \begin{array}{l} 1 \; if \; Y_{ei} = \hat{Y}_{ei}, \\ 0 \; otherwise \end{array} \right\} \tag{5.6}$$

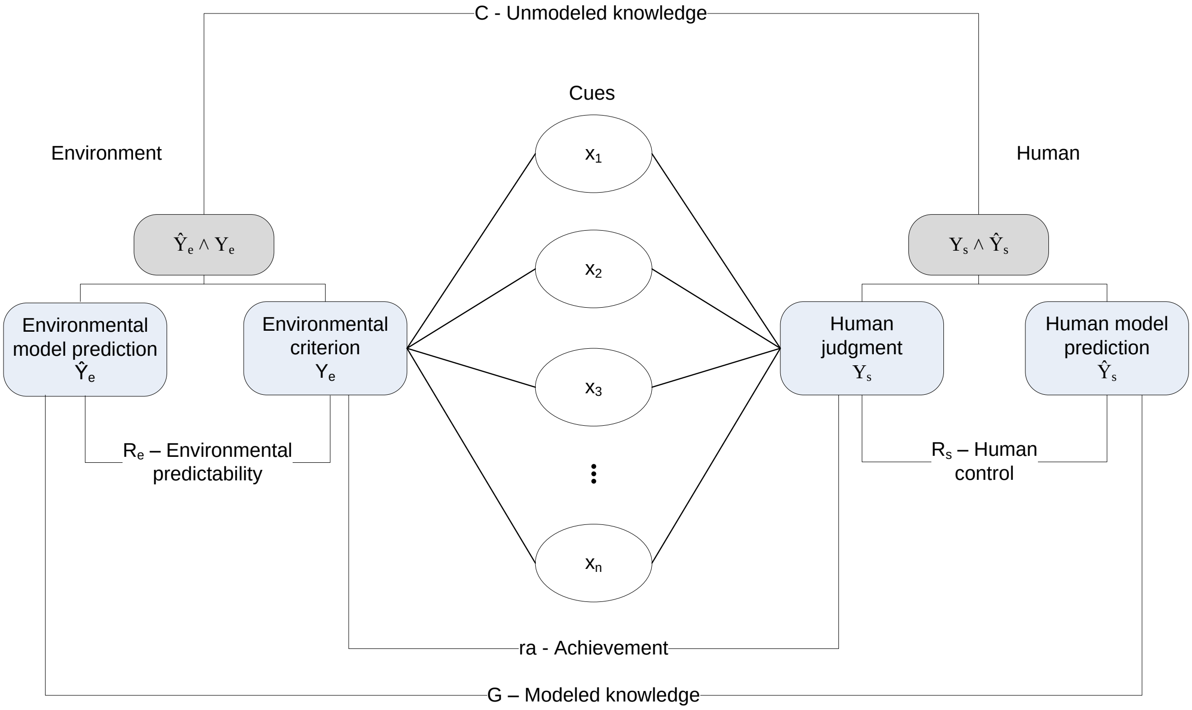


Figure 5.3: Rule-based lens model framework proposed by [Yin 2006].

$$R_s = \frac{\sum_{i=1}^{n} I_{si}}{n} \quad , \quad I_{si} = \left\{ \begin{array}{l} 1\ if\ Y_{si} = \hat{Y}si, \\ 0\ otherwise \end{array} \right\} \tag{5.7}$$

$$ra = \frac{\sum_{i=1}^{n} I_{ri}}{n} \quad , \quad I_{ri} = \left\{ \begin{array}{l} 1\ if\ Y_{ei} = Y_{si}, \\ 0\ otherwise \end{array} \right\} \tag{5.8}$$

$$G = \frac{\sum_{i=1}^{n} IG_i}{n} \quad , \quad IG_i = \left\{ \begin{array}{l} 1\ if\ \hat{Y}_{ei} = \hat{Y}si, \\ 0\ otherwise \end{array} \right\} \tag{5.9}$$

$$C = \frac{\sum_{i=1}^{n} IC_i}{n} \quad , \quad IC_i = \left\{ \begin{array}{l} 1\ if\ I_{ei} = I_{si} = 0, \\ 0\ otherwise \end{array} \right\} \tag{5.10}$$

Fig. 5.3 gives the RLM framework. The interpretation of the lens model parameters for the rule-based case is analogous to the linear case. Achievement, ra, represents the correspondence between human judgments and the actual value of the environmental criterion. Parameter R_e, represents the environmental predictability that measures how well the noncompensatory environmental model can be used to predict the criterion value while R_s, labeled as human control, indicates how well the noncompensatory judgment model captures actual human judgments. Instead of representing linear knowledge in the linear-based lens model, in the RLM, the parameter G represents how well the noncompensatory model of the environment maps onto the noncompensatory model of the human judgments strategy. Now, the C parameter captures systematic regularities between the errors of the noncompensatory ecological and judgment models. In the RLM, the range of parameters ra, R_e, R_s, and G is $[0, 1]$. The closer these values are to 1, the better the achievement, environmental predictability, human control and modeled knowledge, respectively. For C, a high value reveals a high degree of unmodeled knowledge.

5.3 The RBL paradigm proposed for the ANTF

The assisted navigation training framework is based on the RBL paradigm. It uses two main cues to decide the best maneuver to steer the RW, namely: heading error to reach the next subgoal, and obstacle positions. To quantify the progress of user's improving steering skills using the training platform, a criterion definition that is a normative judgment with which the user's judgment is compared is performed. Therefore, a maneuvering set was developed, in which, steering command rules are defined as association rules whose antecedent parts consist of heading errors, and obstacle positions, provided by front and rear laser scan devices installed in the RW, and steering commands as a consequent part. Whenever an human agent makes a maneuvering judgment, the machine agent also outputs a maneuver decision (environmental criterion) based on the cues associated to the heading error and obstacle positions.

5.3.1 GBPC: environment and user models

The machine agent decision rules were developed in order to minimize the turning effort. In this manner system environmental criterion (Y_e) was obtained to be compared with the human judgment (Y_s), allowing to classify the human achievement in steering the RW. To obtain the human and environment models, the cues (inputs) and user judgments (outputs) were encoded in the GBPC as a 14-bit string. Two types of cues were considered: Type I cues related to the heading error; and Type II cues related to obstacle position. The meaning of each string position is shown in Table 5.1.

The environmental criterion was established based on a set of disjunctive and conjunctive rules that are in accordance with the encoded data presented in Table 5.1. An

Table 5.1: Representation and definition of cues (inputs): Type I (related to the heading error), and Type II (related to obstacle position); Representation and definition of judgments (outputs).

Bit	Representation	Definition
Type I cues (Inputs)		
#1	X1	$E_\theta = 0$
#2	X2	$E_\theta = \pi$
#3	X3	$0 < E_\theta \leq \frac{\pi}{2}$
#4	X4	$\frac{\pi}{2} < E_\theta < \pi$
#5	X5	$\pi < E_\theta < \frac{3\pi}{2}$
#6	X6	$\frac{3\pi}{2} \leq E_\theta < 2\pi$
Type II cues (Inputs)		
#7	X7	Front Obstacle
#8	X8	Back Obstacle
#9	X9	Right Obstacle
#10	X10	Left Obstacle
Judgments (Outputs)		
#11	Y1	Go Forward
#12	Y2	Go Backwards
#13	Y3	Rotate Right and Go Forward
#14	Y4	Rotate Left and Go Forward

example of such rules is encoded as follows:

if $(E_\theta = 0) \cap (Go\ Forward) \cup$ is encoded as:

if $(X1) \cap (Y1) \cup$

The 14-bit string representation of the exemplified rule is 10000000001000, as shown in Table 5.2. The exemplar set and the set of rules were established according to the GBPC method described in [Rothrock 2003]. Table 5.2 shows the complete exemplar set defined for the environmental model.

Three human models were defined:

- Beginners: no practice in moving the RW backwards;
- Average: some practice in moving the RW in all directions, but is not skilled to minimize the turning effort;
- Advanced: has practice on moving the RW in all directions, and is skilled to minimize the turning effort;

Table 5.2: Definition of the complete exemplar representation for the environmental model.

Exemplar	Characteristics Represented	14-Bit String
1	$X1 \cap Y1$	10000000001000
2	$X1 \cap X7 \cap Y3$	10000010000010
3	$X1 \cap X7 \cap X9 \cap Y4$	10000010100001
4	$X1 \cap X7 \cap X9 \cap X10 \cap Y2$	10000010110100
5	$X2 \cap X7 = X \cap Y3$	010000X0000010
6	$X2 \cap X7 = X \cap X9 \cap Y4$	010000X0100001
7	$X2 \cap X7 = X \cap X10 \cap Y3$	010000X0010010
8	$X2 \cap X7 = X \cap X9 \cap X10 \cap Y2$	010000X0110100
9	$X3 \cap X7 = X \cap X8 = X \cap Y3$	001000X0000010
10	$X4 \cap X7 = X \cap X8 = X \cap Y3$	000100XX000010
11	$X5 \cap X7 = X \cap X8 = X \cap Y4$	000010XX000001
12	$X6 \cap X7 = X \cap X8 = X \cap Y4$	000001XX000001
13	$X3 \cap X7 = X \cap X8 = X \cap X9 \cap Y4$	001000XX100001
14	$X4 \cap X7 = X \cap X8 = X \cap X9 \cap Y4$	000100XX100001
15	$X5 \cap X7 = X \cap X8 = X \cap X10 \cap Y3$	000010XX010010
16	$X6 \cap X7 = X \cap X8 = X \cap X10 \cap Y3$	000001XX010010
17	$X3 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0010000X111000
18	$X4 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0001000X101000
19	$X5 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000100X011000
20	$X6 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000010X011000
21	$X3 \cap X7 \cap X9 \cap X10 \cap Y2$	00100010110100
22	$X4 \cap X7 \cap X9 \cap X10 \cap Y2$	00010010100100
23	$X5 \cap X7 \cap X9 \cap X10 \cap Y2$	00001010010100
24	$X6 \cap X7 \cap X9 \cap X10 \cap Y2$	00000110010100

Based on the previous assumptions three human user models ($\hat{Y_s}$) were developed also based on a set of disjunctive and conjunctive rules that are also in accordance with the encoded data presented in Table 5.1. Table 5.3 and Table 5.4 show the complete exemplar set defined for the beginner model and average model, respectively. The exemplar set for the advanced user is similar to the one defines for the environmental model (see Fig. 5.2).

Table 5.3: Definition of the complete exemplar representation for the beginner user model.

Ex.	Characteristics Represented	14-Bit String
1	$X1 \cap Y1$	10000000001000
2	$X1 \cap X7 \cap (Y3 \cup Y4)$	10000010000010∪10000010000001
3	$X1 \cap X7 \cap X9 \cap (Y3 \cup Y4)$	10000010100010∪10000010100001
4	$X1 \cap X7 \cap X9 \cap X10 \cap (Y3 \cup Y4)$	10000010110010∪10000010110001
5	$X2 \cap X7 = X \cap (Y3 \cup Y4)$	010000X0000010∪010000X0000001
6	$X2 \cap X7 = X \cap X9 \cap (Y3 \cup Y4)$	010000X0100010∪010000X0100001
7	$X2 \cap X7 = X \cap X10 \cap (Y3 \cup Y4)$	010000X0010010∪010000X0010001
8	$X2 \cap X7 = X \cap X9 \cap X10 \cap (Y3 \cup Y4)$	010000X0110010∪010000X0110001
9	$X3 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	001000X0000010∪001000X0000001
10	$X4 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000100XX000010∪000100XX000001
11	$X5 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000010XX000010∪000010XX000001
12	$X6 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000001XX000010∪000001XX000001
13	$X3 \cap X7 = X \cap X8 = X \cap X9 \cap (Y3 \cup Y4)$	001000XX100010∪001000XX100001
14	$X4 \cap X7 = X \cap X8 = X \cap X9 \cap (Y3 \cup Y4)$	000100XX100010∪000100XX100001
15	$X5 \cap X7 = X \cap X8 = X \cap X10 \cap (Y3 \cup Y4)$	000010XX010010∪000010XX010010
16	$X6 \cap X7 = X \cap X8 = X \cap X10 \cap (Y3 \cup Y4)$	000001XX010010∪000001XX010001
17	$X3 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0010000X111000
18	$X4 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0001000X101000
19	$X5 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000100X011000
20	$X6 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000010X011000
21	$X3 \cap X7 \cap X9 \cap X10 \cap (Y3 \cup Y4)$	00100010110010∪00100010110001
22	$X4 \cap X7 \cap X9 \cap X10 \cap (Y3 \cup Y4)$	00010010100010∪00010010100001
23	$X5 \cap X7 \cap X9 \cap X10 \cap (Y3 \cup Y4)$	00001010010010∪00001010010001
24	$X6 \cap X7 \cap X9 \cap X10 \cap (Y3 \cup Y4)$	00000110010010∪00000110010001

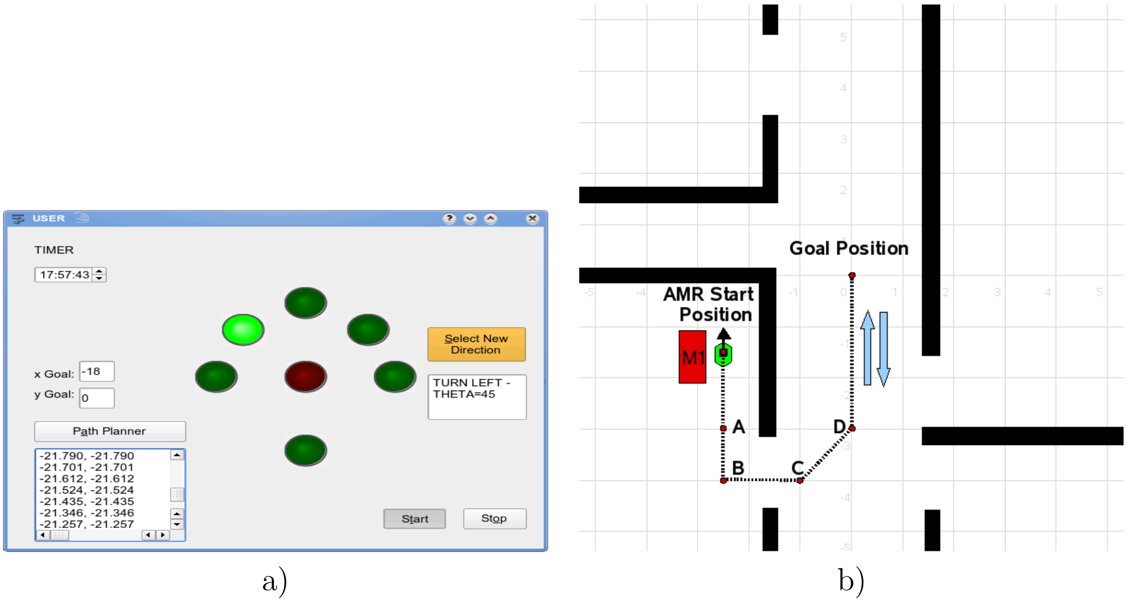
5.4 Experimental results in simulation environment

A training platform simulator using the player/stage [Gerkey 2003] environment was developed. A graphic scanning interface with a switch device is used as the system HMI. Figure 5.4 shows both the graphic scanning interface and the environment chosen to carry out the experiments. The graphic interface only provides sparsely issued commands to the system, namely: go forward, go backwards, go left, and go right, which makes very difficult to steer the RW. To overcome these limitations, the navigation system includes a maneuver executer that receives the desired maneuver, and the next subgoal to be reached, and performs the most appropriate trajectory. Figure 5.5 presents the ANTF

Table 5.4: Definition of the complete exemplar representation for the average user model.

Ex.	Characteristics Represented	14-Bit String
1	$X1 \cap Y1$	10000000001000
2	$X1 \cap X7 \cap (Y3 \cup Y4)$	10000010000010∪10000010000001
3	$X1 \cap X7 \cap X9 \cap (Y3 \cup Y4)$	10000010100010∪10000010100001
4	$X1 \cap X7 \cap X9 \cap X10 \cap Y2$	10000010110100
5	$X2 \cap X7 = X \cap (Y3 \cup Y4)$	010000X0000010∪010000X0000001
6	$X2 \cap X7 = X \cap X9 \cap (Y3 \cup Y4)$	010000X0100010∪010000X0100001
7	$X2 \cap X7 = X \cap X10 \cap (Y3 \cup Y4)$	010000X0010010∪010000X0010001
8	$X2 \cap X7 = X \cap X9 \cap X10 \cap Y2$	010000X0110100
9	$X3 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	001000X0000010∪001000X0000001
10	$X4 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000100XX000010∪000100XX000001
11	$X5 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000010XX000010∪000010XX000001
12	$X6 \cap X7 = X \cap X8 = X \cap (Y3 \cup Y4)$	000001XX000010∪000001XX000001
13	$X3 \cap X7 = X \cap X8 = X \cap X9 \cap (Y3 \cup Y4)$	001000XX100010∪001000XX100001
14	$X4 \cap X7 = X \cap X8 = X \cap X9 \cap (Y3 \cup Y4)$	000100XX100010∪000100XX100001
15	$X5 \cap X7 = X \cap X8 = X \cap X10 \cap (Y3 \cup Y4)$	000010XX010010∪000010XX010010
16	$X6 \cap X7 = X \cap X8 = X \cap X10 \cap (Y3 \cup Y4)$	000001XX010010∪000001XX010001
17	$X3 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0010000X111000
18	$X4 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0001000X101000
19	$X5 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000100X011000
20	$X6 \cap X8 = X \cap X9 \cap X10 \cap Y1$	0000010X011000
21	$X3 \cap X7 \cap X9 \cap X10 \cap Y2$	00100010110100
22	$X4 \cap X7 \cap X9 \cap X10 \cap Y2$	00010010100100
23	$X5 \cap X7 \cap X9 \cap X10 \cap Y2$	00001010010100
24	$X6 \cap X7 \cap X9 \cap X10 \cap Y2$	00000110010100

architecture. Three kind of human behaviors (beginner, average, and advanced) were tested in a virtual environment, in steering the RW in a kind of passing-door maneuver, as depicted in Fig. 5.4 b). Every time a subgoal (denoted by A, B, C, D in Fig. 5.4 b)) is reached the user issue a command to reach the next subgoal. For example, if the user issue a "rotate left" command, and there are no obstacles in the left, then the robot rotates left till the heading error is 0, and then goes forward. The next subgoal and the global path were defined by the system planner (see Fig. 5.4). The user indicates the desired maneuver using the graphic scanning interface (see Fig. 5.4 a)) operated by a switch device. To evaluate the user capability in deciding the best set of maneuvers, the autonomous system (maneuver selector) also decides, in parallel, the best maneuver to reach the desired



TIMER

17:57:43

x Goal: -18

y Goal: 0

-21.790, -21.790
-21.701, -21.701
-21.612, -21.612
-21.524, -21.524
-21.435, -21.435
-21.346, -21.346
-21.257, -21.257

TURN LEFT - THETA=45

Figure 5.4: ANTF simulator: a) scanning interface; b) simulation indoor environment, including a path with one goal position, 4 subgoals (A,B,C,D), and an obstacle M1.

Table 5.5: Example of the RBL model application to an average user.

Subgoal	Y_e	$\hat{Y}_s$	Y_s	I_{si}	I_{ri}	IG_i	IC_i
Start	Y2	Y2	Y2	1	1	1	0
A	Y4	Y4	Y4	1	1	1	0
B	Y4	Y3	Y3	1	0	0	0
C	Y4	Y4	Y3	0	0	1	0
D	Y4	Y3	Y3	1	0	0	0

subgoal, according to the decision rules defined previously. The autonomous system decisions correspond to the environment part of the RBL model (environmental criterion), as shown in Fig. 5.3. The comparison between user (human judge) and autonomous system maneuver decisions, using the RBL model allow us to obtain the user characterization driving the RW.

Table 5.5 shows an example of the application of the RBL model, in a training trial with an average user.

Figure 5.6 shows the simulation results for beginner, average, and advanced users. One can observe that the beginner user had an achievement rate (ra) of 0.37, which is quite low. This result is related to his/her low level of knowledge (G), disabling him/her to perform some types of maneuvers such as moving backwards. The cognitive control

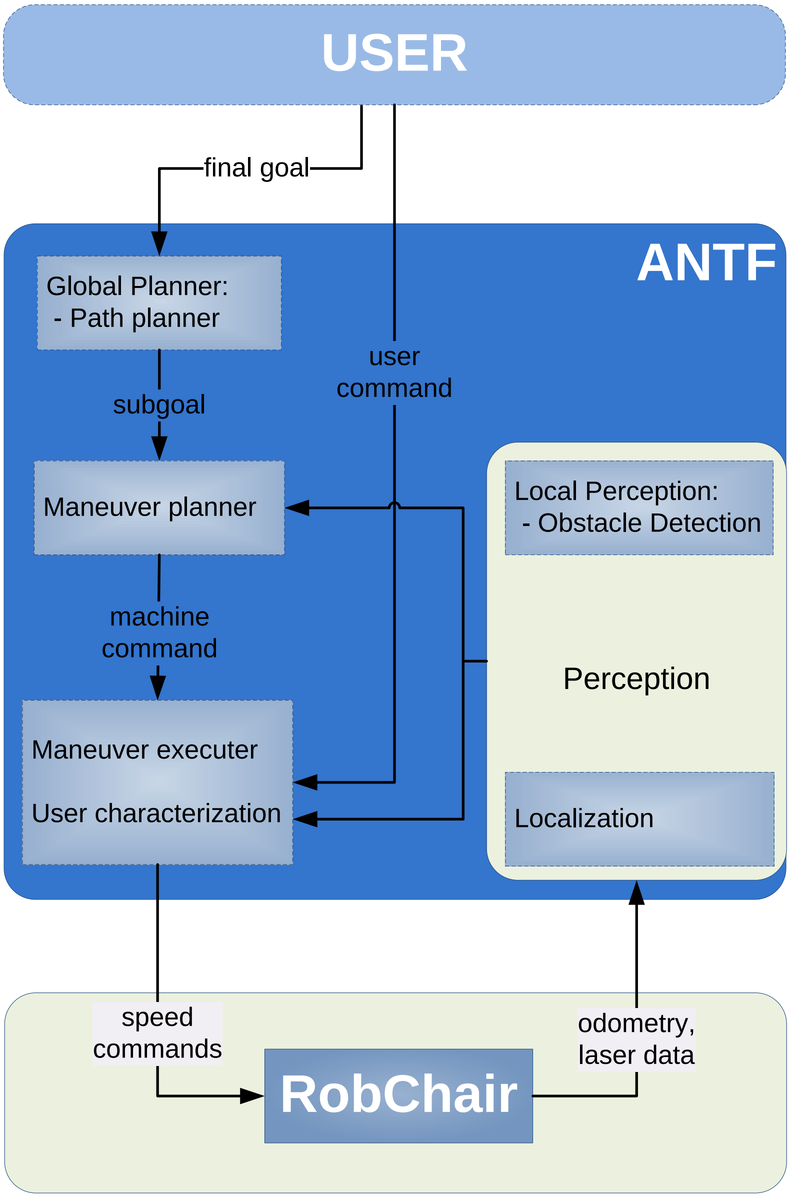


Figure 5.5: Assisted navigation training framework architecture.

(R_s) value shows that some judges did not match the human model for the beginner user. In the case of the advanced user, the achievement rate has a high value, which is related with a high level of knowledge, meaning that the human model for advanced users matches the environmental criterion designed for the autonomous system. The results for cognitive control and achievement rate show that some errors were carried out by this user, which may be related to difficulties in using the HMI device, leading to some mismatches between human judge and human model, and human judge and environmental criterion, respectively. The cognitive control R_s is also a performance indicator of how

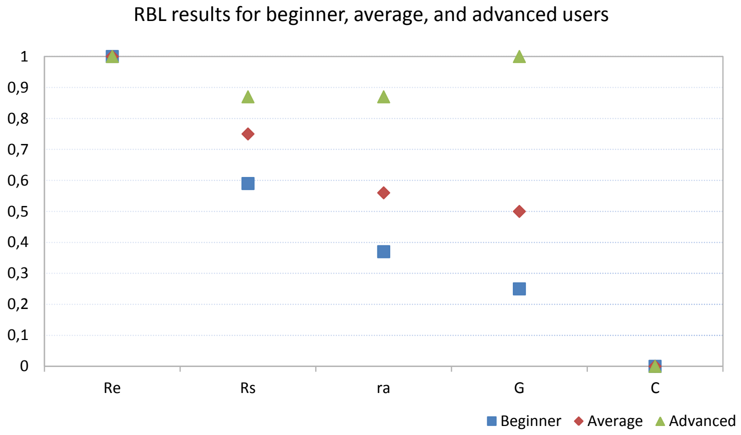


Figure 5.6: Simulation results for beginner, average, and advanced users.

well the human user operates the HMI device. For this case, it was considered that the environmental criterion exactly matched the environmental model. For that reason R_e is equal to one for all cases.

Chapter 6

Collaborative control for an assistive navigation system

Contents

6.1 Introduction

Collaborative control addresses the limitations of other control models (e.g. teleoperation, supervisory, or fully autonomous) through collaboration and dialogue between the agents (e.g. human and robot) integrating the system. Taking the fully autonomous control model as an example, if the robot gets in trouble, it has only two options: to stop or to perform poorly. On the other hand, in a collaborative control model, the robot also has the option of asking the human to assist when needed, by performing perception or cognition, providing advice or information, etc. Thus, more than other system models, collaborative control allows the human to supplement automation and to compensate for inadequacies. Another way in which collaborative control differs is that in which it inherently provides trading/sharing of control and adjustable autonomy. The human is included in the control loop when needed. This is significantly different from other models, such as the supervisory control, which require the user to decide how, when, and where control should be allocated [Fong 2001a].

6.1.1 Design requirements

According to [Fong 2001a], in designing a collaborative controller, four key design issues must be addressed: dialogue, awareness, self-reliance, and adaptiveness. First of all, a collaborative control system must be capable of exchanging information - dialogue. That is, the human and the robot should be able to communicate effectively, convey information, ask questions and judge the quality of received responses [Fong 2001a]. However, when dealing with severely motor impaired humans this question is not so straightforward. Dialogue between humans and robots is carried out through the use of HMIs, which in most cases are unsuited for humans with severe motor limitations (see section 3.2 for details). Thus, the first step is to find the most appropriate HMI to meet user's limitations, and only then is possible to decide a dialogue scheme suited for these users, bearing in mind that they are not able to provide a great deal of information. The dialogue design process should answer the following questions: When should the robot "speak"? Which interface channel does the robot use? and What language features and vocabulary are required to guarantee effective communication?

Awareness, in collaborative control, means that the robot is capable of detecting limitations in what it can do and in what the human can do, determining when, and if, it should ask the human for help, and recognizing when it has to solve problems on its own. A robot operating under collaborative control must be self-reliant, since the robot cannot rely on the human to be always available or to provide accurate information whenever required. This aspect is even more relevant in case of humans with severe motor disabilities with increased difficulties in providing reliable information. A related issue is what to do with non-valid advice. Should the robot ignore the answer or rather reconsider its action?

Adaptiveness to users with variable skills, experience, and training is another fundamental issue in collaborative control. The robot should be able to adapt to different operators. For example, as concerns dialogue, the robot should ask questions based on the operator's ability to answer. Moreover, the robot needs to handle information given by an advanced user differently than that provided by a beginner. Specifically, the robot should evaluate a response based on the extent to which a user can be expected (or trusted) to provide an accurate response. Moreover, because a user's performance (accuracy, reliability, etc.) may vary, the robot may also need to adapt the weight given to user responses over time.

6.1.2 Collaboration in assisted navigation

In mobile robotics, a navigation task comprises everything a robot needs in order to get from a starting point to a goal point, as efficiently as possible, safely avoiding obstacles in the environment. As stated previously, the design of a generic collaborative controller must comply with several requirements of dialogue, awareness, self-reliance, and adaptiveness. When dealing, for example, with the design of an assisted navigation system for users with severe motor disabilities, several problems are raised under those requirements. Concerning dialogue, a question immediately emerges, "Which HMI better complies with the user's motor disabilities?" Another fundamental question concerns the type of information provided by the user, and how the RW deals with that information, which in many cases may be not reliable. To effectively use the HMI, one needs an assistive navigation system that is able to predict and execute user navigation intents with minimum information. The latter problem is directly related with two design requirements: awareness, and self-reliance. This means that the RW must be able to clearly identify situations where aid is required, and ask for assistance in these situations. Since the users are only able to issue sparse information over time, the robot must be self-reliant and able to cope with dynamic changes in the environment, with a minimum assistance from users. The ANS may include an obstacle detection module and a local planner, which allows the RW to deal with changes in the environment, thus giving it some degree of self-reliance. The ANS should also be adaptive to the user's skillfulness in steering the assisted robot. For this purpose, user characterization has to be carried out in advance.

6.2 Motion planning

6.2.1 Related work

The collision-avoidance approaches for mobile robots can be roughly divided into two categories: global and local [Fox 1997]. The global techniques [Latombe 1990, Laumond 1998, LaValle 2006], such as road maps, cell decomposition, generally assume that a complete model of the robot's environment is available. In recent years sampling-based approaches have been successfully proposed to solve challenging motion problems. Examples include Probabilistic Road Maps (PRM) [Kavraki 1996], Randomized Path Planner (RPP) [Barraquand 1991], and Rapidly-exploring Random Trees (RRTs) [LaValle 1999, Cheng 2002]. These practical planners satisfy a weaker form of completeness, i.e., they use randomization to treat the high dimensionality of the configuration space C. Another way to solve the global planning problem is by using a grid map of the environment [Randria 2007]. Thus, each point of the grid becomes a node, and each possible connection between two nodes (which avoids the obstacles and respects physical constraints) becomes an edge. That way a graph is built. There are several algorithms and approaches to find the optimal path in a graph. These methods include the Depth-First Search, the Breadth-First Search, the A* [Hart 1968, Pearl 1982] and Dijkstra [Dijkstra 1959] algorithms. The advantage of global approaches lies in the fact that a complete path or trajectory from the starting point to the target point can be actually computed off-line. However, two important disadvantages can be pointed out: they usually have problems dealing with dynamic changes in the environment (thus avoiding new obstacles in the environment), and problems with robustness when the global world model is inaccurate. For a dynamically changing environment, the global world model must be updated and a new global path has to be re-calculated. The global strategy normally poses problems concerning computational complexity, and slow reaction to changes.

Local or reactive approaches, on the other hand, use only a small fraction of the world model to generate robot control. This comes at the obvious disadvantage that they cannot produce optimal solutions. Local approaches are easily trapped in local minima (such as deadlocks like U-shaped obstacle configurations). There are several local methods proposed in the literature: the potential field methods [Khatib 1985], the Vector Field Histogram (VFH) [Borenstein 1991, Ulrich 2000], or the Dynamic Window Approach (DWA) proposed by [Fox 1997]. These methods are extremely fast, and they typically consider only the small subset of obstacles close to the robot. Most local planning approaches generate motion commands for the RW in two separate stages: in the first stage, a desired motion direction is determined; in the second stage, the steering commands yielding a motion into the desired direction are generated. Local methods quickly adapt to unforeseen changes in the environment. Unlike the potential field method and the VFH, the DWA method incorporates the dynamics of the robot and is particularly suited for robots navigating at high speeds.

6.2.2 Motion planning approach

Our planning approach consists in using the strengths of both local and global planners. When the human provides the robot with a goal destination, the global planner determines the optimal path using a grid-based search algorithm, in our case the A* algorithm (see section 3.1.2 for details). The RW follows the global path, and the local planner only intervenes if and when new obstacles are found close to the RW, which prevents it from following the global path. Our local planner approach is structured in three modules: a steering module where motion direction is determined, using a modified version of the VFH+ algorithm; a local path-planning module, in which a simple path planner was established to determine a reliable path to avoid obstacles close to the robot; and finally, a third module that consists in a blending algorithm that fuses local and global paths. Figure 6.1 presents our planning architecture. In a situation requiring user aid, the collaborative controller receives the candidate directions from the steering module of the local planner. Based on this information, and on the information provided by the user, the collaborative controller determines the final steering command to the local path planner. The overall control system operates at a constant rate of 10 hertz or cycles every 0.1s.

6.2.3 Local planner - Steering

6.2.3.1 Obstacle detection

The perception module is constituted by a localization and an obstacle-detection module, as shown in Fig. 3.1. The latter detects new obstacles in the environment using a

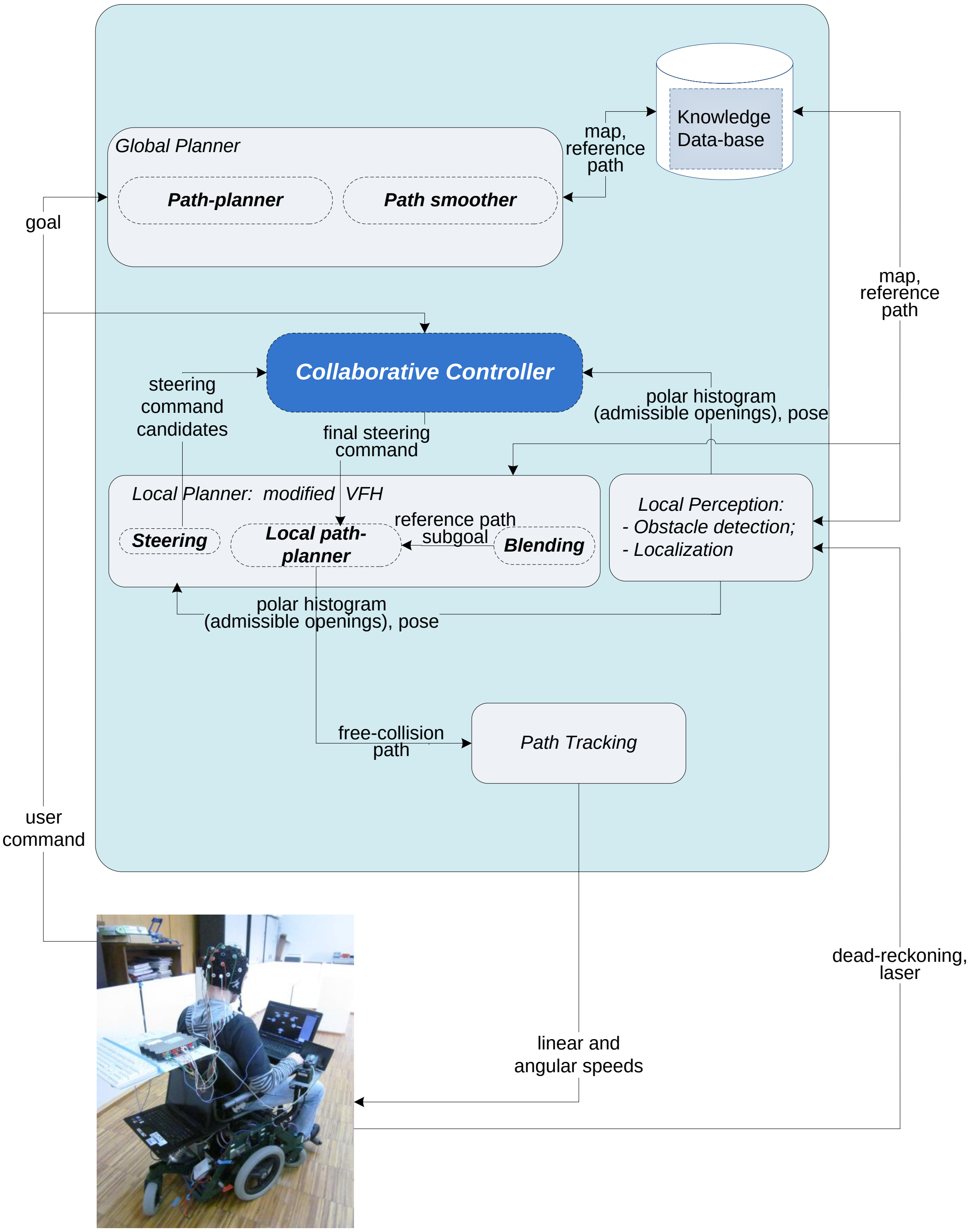


Figure 6.1: Planning strategy consisting in using a global and a local planner. The local planner intervenes every time a new obstacle is detected close to the RW.

matching algorithm, which compares a current laser scan with a predicted laser scan determined from an a priori grid map. Every time an obstacle is closer to the RW than a certain threshold, the obstacle detection module verifies if it is mapped in the a priori grid map, or if it is new to the environment. The obstacle detection approach is presented in Algorithm 7, where $x_{k,t}^{-}$ is the pose estimation, N represents the number of scan points, $\mathscr{M}$ represents the a priori grid map, r_t is the laser scan, and $\theta_{bg}(i)$ is the orientation of the i^{th} scan sector.

Algorithm 7 ObstacleDetection($\mathscr{M}, x_{k,t}^{-}, r_t$)

```
for all $i \in N$ do
  if $r_t(i) < THRESHOLD$ then
    $X_r \leftarrow X_{k,t}^{-} + (r_t(i)) * cos(\theta_{bg}(i) + \theta_{k,t}^{-})$
    $Y_r \leftarrow Y_{k,t}^{-} + (r_t(i)) * sin(\theta_{bg}(i) + \theta_{k,t}^{-})$
    if $\mathscr{M}(X_r, Y_r)$ is occupied then
      $ObstacleDetection \leftarrow true$
      return($ObstacleDetection$)
    end if
  end if
end for
return($ObstacleDetection$)
```

6.2.3.2 Polar histogram

When a new obstacle is detected closer than a certain threshold, avoiding a safe following of the global path, the local planner comes into operation. A polar field histogram is directly determined from the laser scan, assigning to each i^{th} scan sector a value β_i that is equal to the range of that sector. A binary histogram is then determined as follows,

$$b_i = \begin{cases} 1, & \beta_i > \tau_{high} \\ 0, & \beta_i < \tau_{low} \\ b_{i-1}, & otherwise \end{cases} \tag{6.1}$$

Figure 6.2 illustrates the method of finding the polar and binary histograms from a laser scan. To construct the binary polar histogram we used a hysteresis based on two thresholds, as proposed in [Ulrich 2000], namely τ_{low} and τ_{high}, as a way to avoid problems in environments with several narrow openings, as the corresponding opening in the histogram can alternate several times between an open and a blocked state during a few sampling times. Note that b_i is zero, if the sector is considered to be free of obstacles, and one, if it is being occupied.

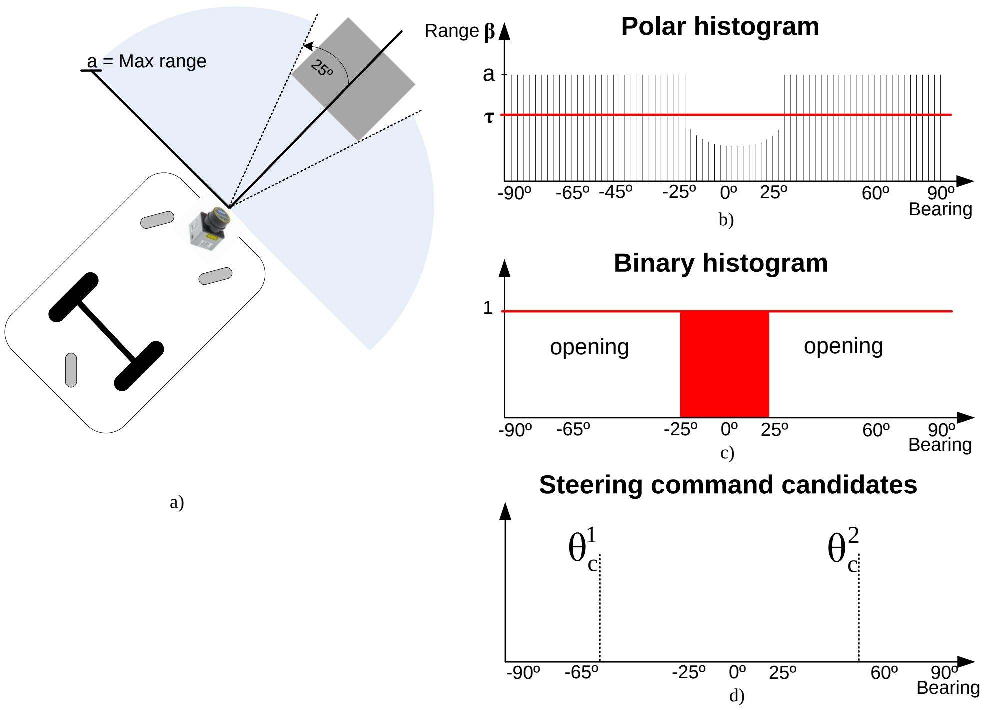


Figure 6.2: Finding the polar histogram and binary histogram from a laser scan.

6.2.3.3 Admissible steering commands

The next step consists in determining the admissible steering commands based on the binary polar histogram. The admissible openings can be defined as having sufficient free space to ensure a safe navigation of the RW. As proposed in the VFH+ method [Ulrich 2000], first all openings in the binary polar histogram must be found, and then a set of possible candidate directions can be determined. In the first step, the right and left borders k_r and k_l of all openings in the binary polar histogram, are calculated. Then, two types of openings are distinguished, namely, wide and narrow ones. An opening is considered wide if the difference between its two borders is larger than a given threshold, s_{max}. The opening is considered narrow if it is larger than s_{min} and smaller than s_{max}. Otherwise, it is excluded as an admissible opening. For a narrow opening, there is only one candidate direction so that the robot steers through the center of the gap between the corresponding obstacles:

$$\theta_n = \frac{k_r + k_l}{2} \tag{6.2}$$

For a wide opening, there are two candidate directions, one to the right and one to the left side of the opening. The target direction is also a candidate direction, if it lies between the two other candidate directions:

$$\begin{aligned}
&\theta_r = k_r + \tfrac{s_{max}}{2} \\
&\theta_l = k_l - \tfrac{s_{max}}{2} \\
&\theta_t = k_t,\ k_t \in [\theta_r, \theta_l]
\end{aligned} \tag{6.3}$$

The candidate directions θ_r and θ_l make the robot follow an obstacle contour at a safe distance, while θ_t leads the robot towards the target direction. Next, we need to define an appropriate cost function, so that the robot selects the most appropriate candidate direction as its new motion direction φ_n. Each candidate direction θ_i is weighted using the cost function,

$$\gamma(\theta_i) = \mu_1 \mathit{\Delta}(\theta_i, \theta_t) + \mu_2 \mathit{\Delta}(\theta_i, \theta) \tag{6.4}$$

where θ_t is the target direction, and θ is the current direction of the RW. The generic term $\mathit{\Delta}(c_1, c_2)$ gives the absolute angle difference between two generic sectors c_1 and c_2. Terms μ_1, and μ_2 are cost parameters.

6.2.4 Local planner - local path-planner

The local path-planner computes a feasible path between any two configurations, considering some intrinsic motion constraints of the mechanical system (it accounts for constraints imposed by joints and non-holonomic constraints but ignores collision-avoidance). This choice may affect the combinatorial complexity of the algorithm. However, in mobile and free-flying systems that are not affected by differential constraints, devising steering methods is generally more straightforward. We propose a simple and effective local path-planner that determines local paths based on linear interpolations, between the current pose of the RW and a new target pose that is located ahead of the RW, in the direction of φ_n, the new direction of motion calculated in the steering module. Figure 6.3 shows an example of how the local planner works to avoid an obstacle placed in the robot path.

6.2.4.1 Path planning method

As depicted in Fig. 6.3, after detecting the obstacle, a new path on the direction φ_n, is planned as a linear interpolation between the current pose of the RW and the target pose $P1$ (a point in the free space between the obstacle and the infra-structure). After reaching $P1$, a new path is calculated in order to reach $P2$. This procedure is repeated until the environment close to the robot is free of new obstacles. A sliding mode path-following controller is used as the system path tracker [Solea 2009].

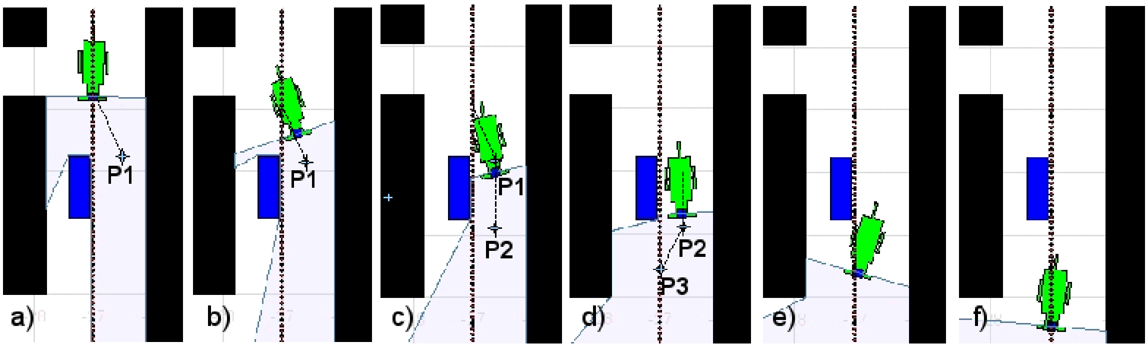


Figure 6.3: Obstacle avoidance method: a) robot detects the obstacle and determines a free path to a target pose $P1$ (center point of the free space between the obstacle and the infrastructure in the direction provided by the collaborative controller); b) robot follows the path; c) robot reaches $P1$, and determines a path to pose $P2$; d) robot reaches $P2$ and determines a path to merge with global path $P3$; e) robot follows the path to reach $P3$; f) robot follows the global path. The dotted line represents the path provided by the global planner.

6.2.5 Local planner - blending

In order to blend the global and local paths, the target direction θ_t is always a subgoal of the global path, i.e. it is the next subgoal provided by the global path. As soon as the RW is free of new obstacles, a final path is planned. This procedure is depicted in Fig. 6.3 d) to f), when a path to reach $P3$ is planned to blend the local path with the global one. This final path is planned as a linear interpolation between the robot's current pose and a subgoal of the global path, which should be sufficiently ahead to comply with the geometric constraints of the RW during obstacle contour. This assumption is guaranteed in all experiments presented in this thesis. However, the blending algorithm requires further developments to ensure generalization.

6.3 Two-layer collaborative controller

We propose a collaborative control approach that depends on the user's ability to steer the RW. Therefore, user characterization is carried out previously with the ANTF platform as proposed in section 5. Figure 6.4 shows how the four design requirements presented so far are integrated in the collaborative controller proposed for the ANS. Dialogue is performed using a discrete HMI, allowing the user agent to issue sparse and discrete commands, including a graphic interface where information is provided to the user. Awareness is achieved via a perception module that is able to determine the correct pose of the robot,

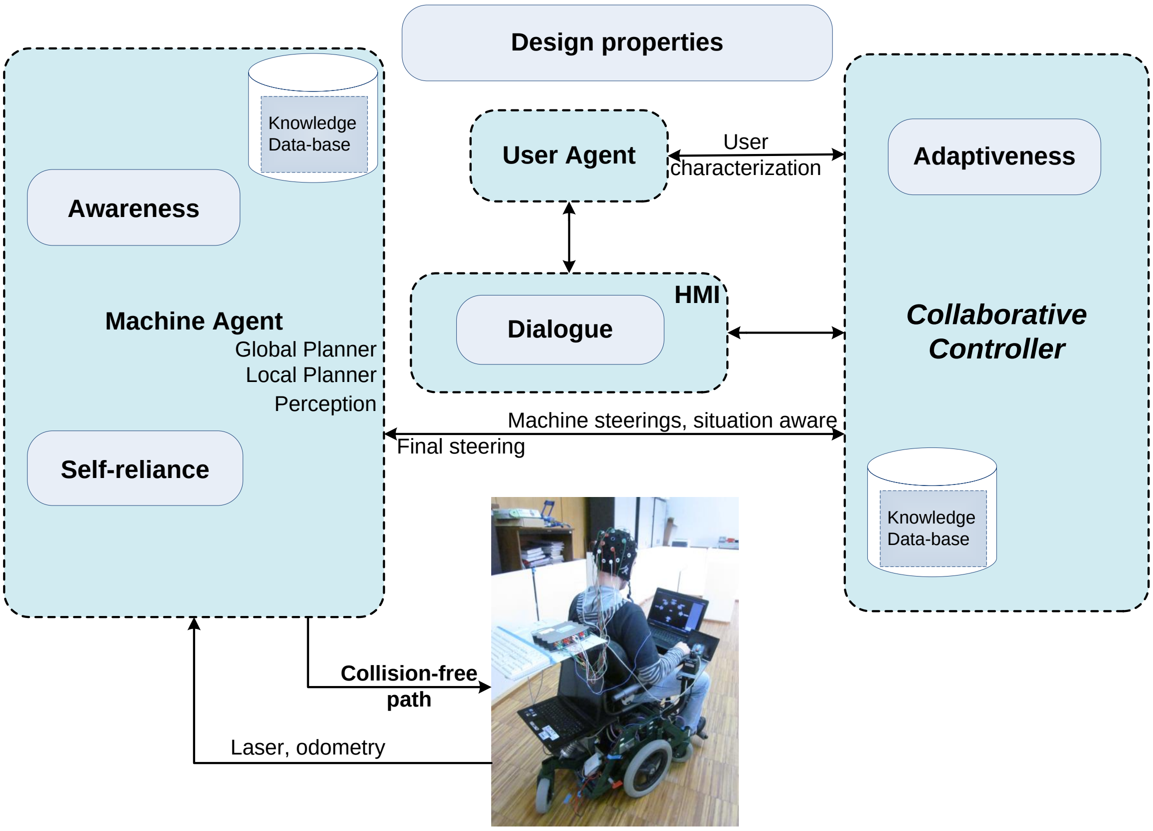


Figure 6.4: Highlight of the collaborative control design requirements: dialogue, awareness, self-reliance, and adaptiveness.

and to perform obstacle detection. The machine agent includes a global and a local planner that provides the RW with some degree of self-reliance. For a set of specific situations in which the machine agent requires user assistance, the collaborative controller goes into action, and determines the final steering command based on the information provided by both user and machine agents. A two-way arrow is used to show the information flow between the user agent and the adaptiveness requirement of the collaborative controller. In fact, user characterization is not static but rather dynamic, as it evolves as user experience in using the ANS increases.

6.3.1 Dialogue

People with severe motor disorders, such those suffering from some types of cerebral palsy disorders, are unable or have great difficulty using conventional HMIs. Non-invasive BCI is thus emerging as a promising HMI alternative for these users. This type of interface offers a communication channel that is independent of muscular activity, and can therefore

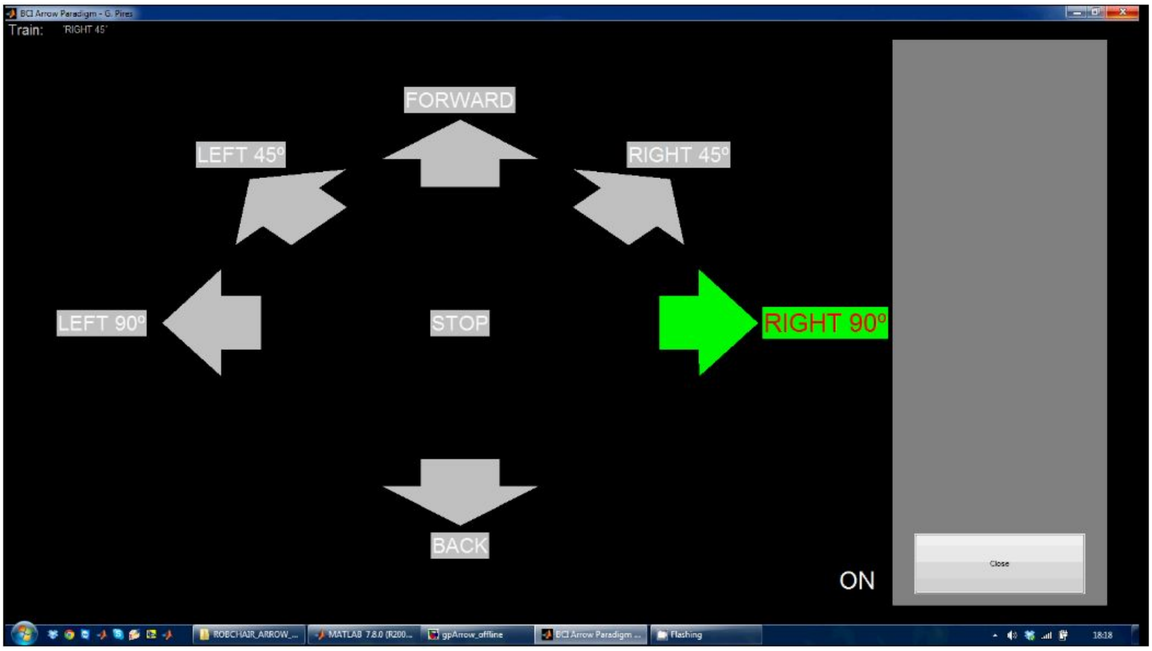


Figure 6.5: Arrow paradigm used for command selection with the BCI system.

be used by people with severely affected motor capability [Pires 2011a]. However, the development of a BCI-guided RW is full of challenges. BCI offers a low information transfer rate associated with a non-negligible error rate. In other words, the user will only be able to provide sparse commands in time and some of these may be unreliable. The BCI, described in Section 3.2.3 (see Fig. 3.3b)), has been designed as a visual oddball paradigm, where symbols flash randomly. At a given moment, the relevant (target) event is the symbol mentally selected by the user, which corresponds to the direction he/she wants to follow, and all other flashing symbols are the standard (non-relevant) events. Figure 6.5 shows the graphic interface, also referred to as the arrow paradigm, used for command selection with the BCI system.

In this thesis, an HMI composed by a switch or multi-switch system and the graphic interface depicted in Fig. 3.9, is also considered. More details on this type of HMI are given in subsection 3.3.3. The information transfer rate of this solution may be significantly higher than the one presented by the BCI system described in this thesis, if the user has the appropriate skills and training. Moreover, the users are able to issue commands in a self-paced manner. The major drawback is related to the fact that this HMI is not independent from user's motor capability, and it may cause fatigue since the user needs to carry out the same movement repeatedly, in order to issue the desired commands.

6.3.2 Awareness

Another challenge raised by the use of the BCI proposed in [Pires 2011a] is related with the fact that commands are not yet issued in a self-paced manner by the user, which would allow the user to issue commands only when desired. Additionally, the BCI system is still not able to identify when the user is in idle state (not focusing on any symbol flash), and therefore the BCI system acts as if the user was actually selecting an available command. The collaborative controller is able to cope with these drawbacks through the use of a traded control layer, here designated by virtual constraint layer. A set of situation-awareness criteria was previously defined, and the virtual constraint layer only allows the user assistance when the criteria are satisfied. The perception module and the local planner are responsible for identifying the ambiguous situations that might require user aid, such as bifurcations, deadlocks, or multiple directions.

6.3.3 Self-reliance

Our ANS includes a perception module and a planning strategy that are able to give the RW some degree of self-reliance. In all situations not considered ambiguous by the RW, the traded controller ignores any user-issued command, and the RW works as if it was autonomous. Again, this layer is fundamental to our collaborative controller, once it filters the BCI commands issued while the user is in "idle state", i.e. the user is not aware in selecting the commands.

6.3.4 Adaptation

User characterization was carried out using the ANTF [Lopes 2009] as proposed in chapter 5. The ANTF deployment had two main goals: to train users to carry out navigation tasks in an automatic manner (not requiring continuous help from therapists), and to characterize user profiles in steering a powered wheelchair. The ANTF platform classifies users into three learning stages, according to their steering capabilities: beginner, average, and advanced user. For each stage of development the user has an efficiency rate r_a that is used in the intent-matching layer of the proposed shared-controller.

6.3.5 Collaborative control architecture

The collaborative control architecture receives commands from two agents: the User Agent (UA), and the Machine Agent (MA). The user issues commands θ_{UA} using the BCI, or

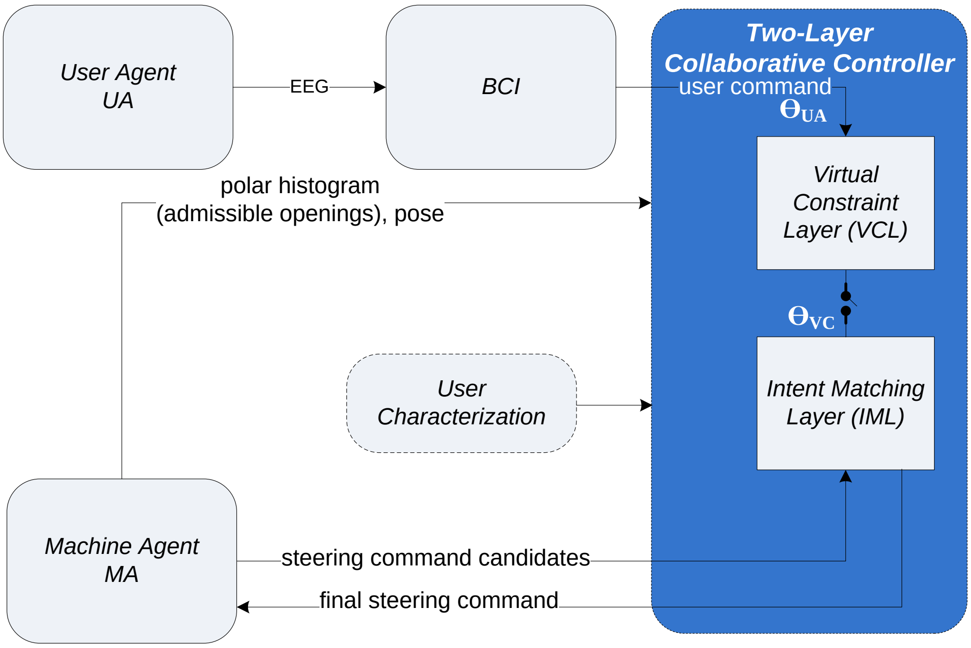


Figure 6.6: Two-layer collaborative control architecture.

other type of discrete and sparse interface (see section 3.2 for details). The proposed collaborative control architecture includes two layers, namely, a virtual-constraint layer and an intent-matching layer. The former is a traded control layer responsible for enabling/disabling user commands, as a function of situation-awareness criteria, and the latter is a shared control layer that determines suitable maneuvers, taking into account user steering competence, as outlined in Fig. 6.6. Two alternative approaches are proposed for the shared control layer, namely an intent matching layer, and a fuzzy layer, to be described later in this chapter.

6.3.6 Virtual-constraint layer for traded control approach

The Virtual-Constraint Layer (VCL) is responsible for enabling/disabling the commands provided by the user, subject to a set of constraints. If using a BCI system, this layer is primarily required to avoid idle-state commands. If using a switch/multi-switch HMI, it prevents erroneous commands issued by the user. When commands are disabled, the system becomes autonomous. User commands are enabled by the VCL according to the following perceived situations:

- $S1$: Multiple possible directions due to a bifurcation;
- $S2$: Multiple possible directions to avoid an obstacle;

- $S3$: Solving a deadlock by moving backwards;
- $S4$: Solving a deadlock with left/right pure rotations.

The VCL includes a situation-awareness module responsible for detecting the occurrence of the referred perceived situations. When one of those situations occurs, the user is asked to select a desired steering command, through the visual arrow paradigm shown in Fig. 6.5. Additionally, the VCL also takes into account constraints associated with user steering competence, as follows:

$$\begin{array}{l}
Basic\ User: \\
\theta_{VC} = \begin{cases} \theta_{UA} & if(S1\ \&\ \theta_{UA} \in \{F,\ L90,\ R90\}) \\ \theta_{UA} & if(S2\ \&\ \theta_{UA} \in \{F,\ L,\ R\}) \\ \theta_{UA} & if(S4\ \&\ \theta_{UA} \in \{L90,\ R90\}) \\ none & otherwise \end{cases} \\
Average\ User: \\
\theta_{VC} = \begin{cases} \theta_{UA} & if(S1\ \&\ \theta_{UA} \in \{F,\ L90,\ R90\}) \\ \theta_{UA} & if(S2\ \&\ \theta_{UA} \in \{F,\ L,\ R\}) \\ \theta_{UA} & if(S3 \mid S4) \\ none & otherwise \end{cases} \\
Advanced\ User: \\
\theta_{VC} = \begin{cases} \theta_{UA} & if(S1\ \&\ \theta_{UA} \in \{F,\ L90,\ R90\}) \\ \theta_{UA} & if(S2 \mid S3 \mid S4) \\ none & otherwise \end{cases}
\end{array} \tag{6.5}$$

where $F \equiv FORWARD$, $L \equiv (LEFT45|LEFT90)$, $R \equiv (RIGHT45|RIGHT90)$, $L90$ denotes a rotation left ($LEFT90$), and $R90$ denotes a rotation right ($RIGHT90$).

6.3.7 Intent-matching layer for shared control approach

The Intent-Matching Layer (IML) layer determines the final steering command to the RW, based on user-command properly modified by the VCL, and taking into account a set of candidate directions proposed by the MA. Errors $e^i_{\theta_{UA}}$, $i = 1, ..., n$ between the user command and the directions provided by the MA are calculated as follows:

$$e^i_{\theta_{UA}} = \theta_{VC} - \theta^i_{MA},\ i = 1, ..., n \tag{6.6}$$

where n is the number of MA candidate directions. Each MA candidate direction has an associated weight η^i_{MA} that is calculated as the inverse of its cost (6.4). A normalization factor is applied so that the sum of all machine weights is equal to one. For each candidate direction a cost function $g(\theta)$ is defined as follows:

$$g(\theta) = min\{\eta^i_{MA} \cdot e^i_{\theta_{MA}} + \eta_{UA} \cdot r_a \cdot e^i_{\theta_{UA}}\},\ i = 1...n \tag{6.7}$$

where n is the number of candidate directions, and $e^i_{\theta_{MA}}$ is the error between the selected direction and each candidate direction,

$$e^i_{\theta_{MA}} = \theta^i_{MA} - \theta_{MA}\ i = 1, ..., n \tag{6.8}$$

The weight η_{UA} is defined according to the user steering competence, and r_a denotes his/her achievement rate, which varies according to $r_a \in [0...1]$. The final direction θ is the one that the results in the minimum value of the cost function $g(\theta)$. For a perceived deadlock situation (S3 and S4), the MA is unable to determine any free direction and, in such a case, the user should command the RW backwards or perform pure left or right rotations (commands BACK, LEFT90, and RIGHT90, respectively) to leave the deadlock. In case of a S1 situation, the RW waits for an admissible command to solve a particular bifurcation. Only the commands FORWARD, LEFT90, and RIGHT90 are allowed. This set of commands may be narrowed down or expanded according to each bifurcation's characteristics, as happens in the experiments described in chapter 7.

6.3.8 Fuzzy layer for shared control approach

The second shared control approach is denoted by Fuzzy Shared Controller (FSC). It has two types of input variables: the user-intent (a sparse issued command) θ_{UA}, and the steering provided by the local planner module θ_{MA}. Unlike the previous approach, which considered that all steering commands proposed by the machine agent were candidate directions to be matched with user-intent, here only the best steering command proposal given by the machine agent is considered, i.e. the one that results in the minimum value of the cost function (6.4). The final steering command is the output variable of the FSC, which is then provided to the local path-planner.

6.3.8.1 Fuzzification Process

The fuzzification module transforms numerical variables in fuzzy sets that can be manipulated by the controller. The FSC uses fuzzy singletons to encode user intent, and consists in a small set of steering commands issued sparsely. The output variable is also modeled as discrete fuzzy set. A triangular membership function is used to encode the target directions provided by the local planner. Figure 6.7 shows the membership functions used to encode inputs and outputs of the FSC module.

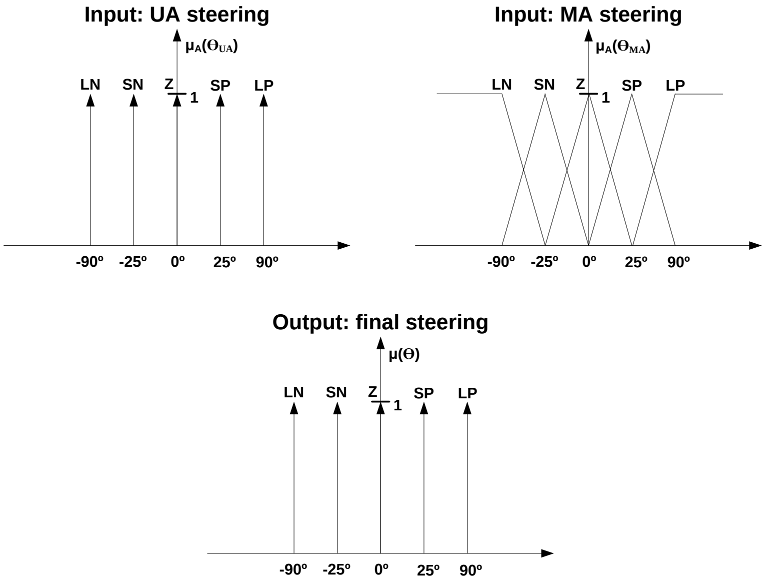


Figure 6.7: Fuzzy set variables: LN - Large Negative (large turn left), SN - Small Negative (small turn left), Z - Zero, SP - Small Positive (small turn right), LP - Large Positive (large turn right)

6.3.8.2 Inference Mechanism

The inference process generally has two steps: first, a matching, that determines which rules are on, and conclusions, that determine which control actions to take. The rule table for final steering command variable is presented in Table 6.1. For the machine-agent steering command variable it is possible to have one or two rules at one time. In this way, it is necessary to determine which conclusions should be taken into account when the rules that are on are applied. The computation of the activation degree α_i, for a rule i is given by

$$\alpha_i = \min(\mu_A(\theta_{MA}), \mu_A(\theta_{UA})) \tag{6.9}$$

Table 6.1: Fuzzy logic matrix for final steering command variable according to user characterization

Beginner User					
	MA steering command				
UA steering	**LN**	**SN**	**Z**	**SP**	**LP**
LN	LN	LN	Z	SP	LP
SN	LN	SN	Z	SP	LP
Z	LN	SN	Z	SP	LP
SP	LN	SN	Z	SP	LP
LP	LN	SN	Z	LP	LP
Average User					
	MA steering command				
UA steering	**LN**	**SN**	**Z**	**SP**	**LP**
LN	LN	LN	Z	SP	LP
SN	SN	SN	Z	SP	LP
Z	LN	Z	Z	Z	LP
SP	LN	SN	Z	SP	SP
LP	LN	SN	Z	LP	LP
Advanced User					
	MA steering command				
UA steering	**LN**	**SN**	**Z**	**SP**	**LP**
LN	LN	LN	LN	LN	LN
SN	SN	SN	SN	Z	SP
Z	SN	Z	Z	Z	SP
SP	SN	Z	SP	SP	SP
LP	LP	LP	LP	LP	LP

where A denotes the input fuzzy sets (defining the linguistic terms), and θ_{MA} and θ_{UA} are the input variables.

6.3.8.3 Defuzzification Process

One fuzzy set results from the inference process, one for linear speed, and the other for angular speed. The defuzzification module outputs, from these sets, numerical values to send to the actuators. We have chosen the Centre of Gravity method (COG), because it takes into account the distribution of the resulting fuzzy set. Applied to fuzzy singletons, the COG method simplifies computation of inference mechanism and reduces processing time, without degradation of the defuzzified value. The defuzzified outputs are given by

$$u = \frac{\sum_i \alpha_i s_i}{\sum_i \alpha_i} \tag{6.10}$$

where α_i (6.9) is the degree of activation of the ith rule, and s_i is the output singleton.

6.4 Experimental results

The navigation system is intended to reduce user effort in steering the RW. In the current experiments the user selects global goals, and the navigation module follows the paths determined by the global planner in order to reach these goals. The navigation module considers the BCI input commands according to user steering competence and situation-awareness, particularly if situations S1-S4 occur. These situations were experimentally tested, and results are presented in the sequel.

6.4.1 Steering command selection to avoid an obstacle (S2)

Figure 6.8 shows results regarding the navigation of a wheelchair in an office-type-building scenario. These results relate to situation S2, described as multiple direction possibilities to avoid new obstacles in the global path. This experiment was simulated in player/stage environment for three types of users, according to their steering competence: basic, average, and advanced. According to Fig. 6.8, when position 1 is reached, the navigation system faces an ambiguous situation, and the user is asked to select the desired direction. The final steering command is then calculated according to (6.7). For these experiments, machine steering weights η^i_{MA} are equal to 1, and user weights are as follows: $\eta^{basic}_{UA} = 2$, $\eta^{average}_{UA} = 3$, and $\eta^{advanced}_{UA} = 4$. An achievement rate $r_a = 1$ was also considered in all experiments. According to user weights, and considering an achievement rate of 1, it is possible to conclude that advanced and average users always have the power to change the direction selected by the MA. Basic users have always less power than the MA. Of course, with lower achievement rates user power can be reduced to nearly zero, and the system may become autonomous for any type of user. The results depicted in Fig. 6.8 are similar for all types of users, with weights and achievement rate described previously. Figure 6.9 shows a S1 experiment using RobChair, the real platform. Robchair detects an obstacle, and the MA proposes two steering command options due to the door opening, leading to a dubious situation. The user chooses the appropriate steering command (LEFT command), the obstacle is avoided, and RobChair is able to reach the final goal.

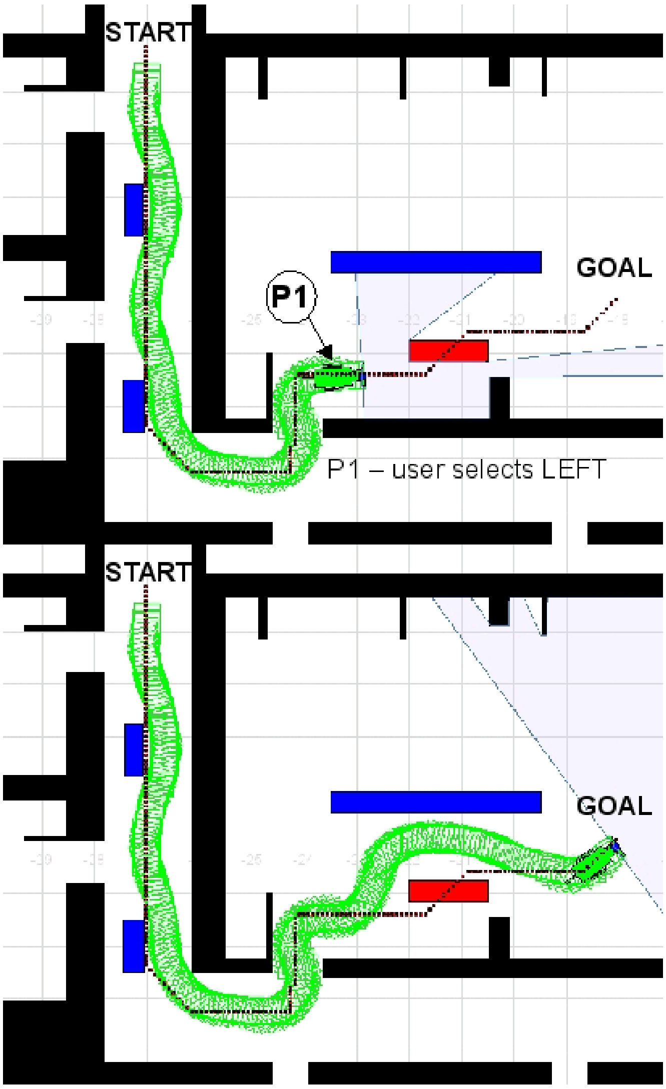


Figure 6.8: Steering command selection to avoid an obstacle. In position P1, the user selects to move left to avoid the obstacle.

6.4.2 The deadlock problem (S3, S4)

Figure 6.10 shows how different users solve the deadlock problem. Basic users are not allowed to move backwards, and therefore we can observe that in position 1 they can only perform pure rotations (LEFT90 or RIGHT90) if they are facing a deadlock situation, in which the MA is not able to determine any free direction. Average and advanced users can solve the deadlock because they are allowed to move backwards when facing a deadlock situation.

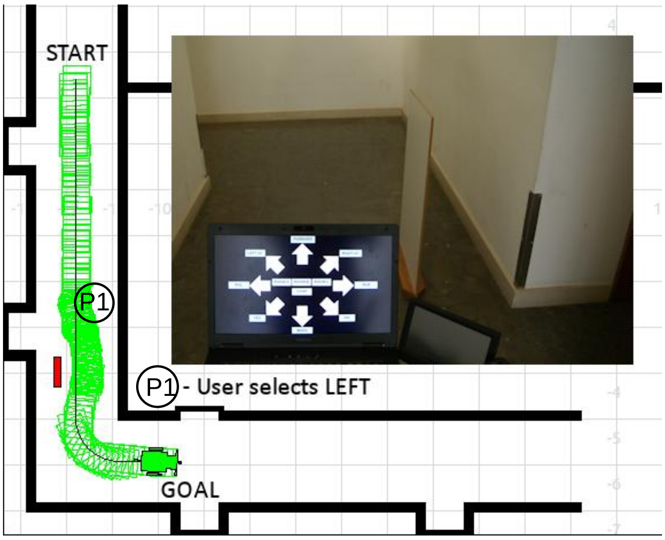


Figure 6.9: Direction selection to avoid an obstacle. In position P1, the user selects to move left to avoid the obstacle. The MA proposes two steering directions due to the door opening.

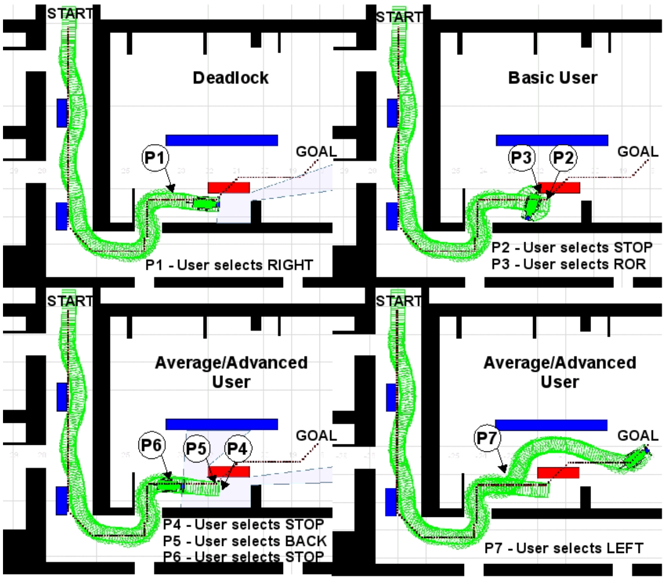


Figure 6.10: Experimental results: each user tries to solve the deadlock problem: In position P1, the user selects RIGHT and enters the deadlock; In P2 and P3, the basic user stops and tries to perform a pure rotation right (ROR) but gets stuck; In P4 and P5, the average/advanced user stops and move backwards (BACK); In P6 and P7, the average/advanced user stops and selects LEFT.

Chapter 7

Experimental results and analysis

Contents

In this chapter we present an assessment of the current stage of RobChair navigation system, in which users are able to steer the wheelchair using a P300-based BCI. Two different navigation tasks were used to assess the performance of the ANS based on BCI. The first navigation task took place in a structured known environment, and the second one in a structured unknown environment with the presence of new static and moving obstacles (e.g. pedestrians). The two navigation task experiments were carried out with ten able-bodied participants and a participant with cerebral palsy and motor impairment. All participants gave informed consent to participate in the study. The navigation module processes the BCI input commands according to user steering competence, and the situation perceived by the machine (e.g. situations S1-S4). S1 and S2 were experimentally tested, and results are presented bellow, in order to assess the performance of BCI online, the assistive navigation system, and users. A video showing parts of the experiments described in this section can be seen in [rob 2012].

7.1 Characterization of participants

Participants without disabilities included: aged between 25 and 40 years old; approximate number of participants from both genders; higher education; all right-handed; without any relevant history of psychiatric or neurological disorders. Ten able-bodied participants, six male and four female, were selected, all of them being researchers or graduate students at university of Coimbra. From this group, two of them had considerable experience using the P300-based BCI system, three of them had used the BCI system once, and five of them used the P300-based BCI system for the first time. Only one user was familiar with the Arrow Paradigm (AP) used in the BCI. Only one female participant with motor disabilities has carried out the proposed experiments. This participant suffers from cerebral palsy disorders and is severe motor impaired. Mainly due to comfort reasons this participant did not sit in RobChair and carried out tests remotely. This participant has high experience in steering a power wheelchair, but using a head-switch interface, and has low experience using the P300-based BCI system, and the AP. Table 7.1 shows the most relevant data concerning participant characterization. All participants used the ANS for the first time.

Table 7.1: Characterization of participants. Degree of motor disability and participant experience are classified as none, low, moderate or high. Gender is classified as M for male and F for female.

				Experience		
Participant	Age	Gender	Degree of Motor Disability	Steering RW	Using BCI	Using AP
1	36	F	None	Low	Low	None
2	25	M	None	Low	High	None
3	40	F	None	None	Low	None
4	39	M	None	Low	High	Moderate
5	37	F	None	None	None	None
6	31	F	None	None	Low	None
7	28	M	None	None	None	None
8	31	M	None	None	None	None
9	36	M	None	None	None	None
10	29	M	None	None	None	None
11	25	F	High	High	Low	Low

7.2 Experimental design and procedures

The experimental tests started with a BCI training/calibration phase where participants selected a set of predefined commands, using the AP. After the training/calibration phase, each participant was required to perform two real-time navigation tasks, in the wheelchair. The first navigation task, denoted by TASK1, took place in a structured known environment that only included static mapped obstacles (see Fig. 7.1 b)); the second task, denoted by TASK2, was carried out in a structured unknown environment, which included new obstacles in the environment of two types: static, and some moving obstacles, such as pedestrians walking in the set (see Figs. 7.1 b), c)). Figure 7.2 shows a scene of the real test environment.

TASK1 consisted in navigating the wheelchair from START to GOAL1, and returning to START again (designated as GOAL2). This navigation task was organized in a sequence of 12 waypoints denoted as sequence A, which is presented in Table 7.2. The Decision Target (DT) points were B, E and G, where the user was able to decide if he/she wants turn right (RIGHT90) or left (LEFT90). To accomplish the navigation task with success, *i.e.* with a minimum number of required waypoints, each user needed to select an appropriate command (RIGHT90 or LEFT90) when a DT point was reached. If the user selected other command besides RIGHT90 or LEFT90, the RW remained stopped till an appropriate command was provided. If the user selected a wrong but admissible command, the RW followed the path selected by the user, and in that case the required waypoint sequence was not followed as requested. Simultaneously, to assess the online BCI performance, it was also asked to each user to select a set of predefined commands between DT points. The commands between DT points were not taken into account by the navigation system, and are designated as non-decisive commands.

In TASK2, users were asked to perform two navigation sequences (sequence B or sequence C as defined in Table 7.2) with the RW, in a structured unknown environment, with the presence of new static obstacles and moving obstacles. The users were allowed to perform one or both sequences defined in Table 7.2 for TASK2.

For all experiments presented in this thesis, η_{UA} and ra parameters were tuned manually, as proposed in [Lopes 2011], because all users experimented the ANS for the first time. A user weight parameter, $\eta_{UA} = 2$, (all participants are beginners), and an achievement rate, $ra = 1$, were also considered in all experiments. To calculate the candidate directions of (6.2) and (6.3), a s_{max} of 120° and a s_{min} of 70° were considered. The cost parameters in (6.4) were as follows: $\mu_1 = 2$ and $\mu_2 = 1$.

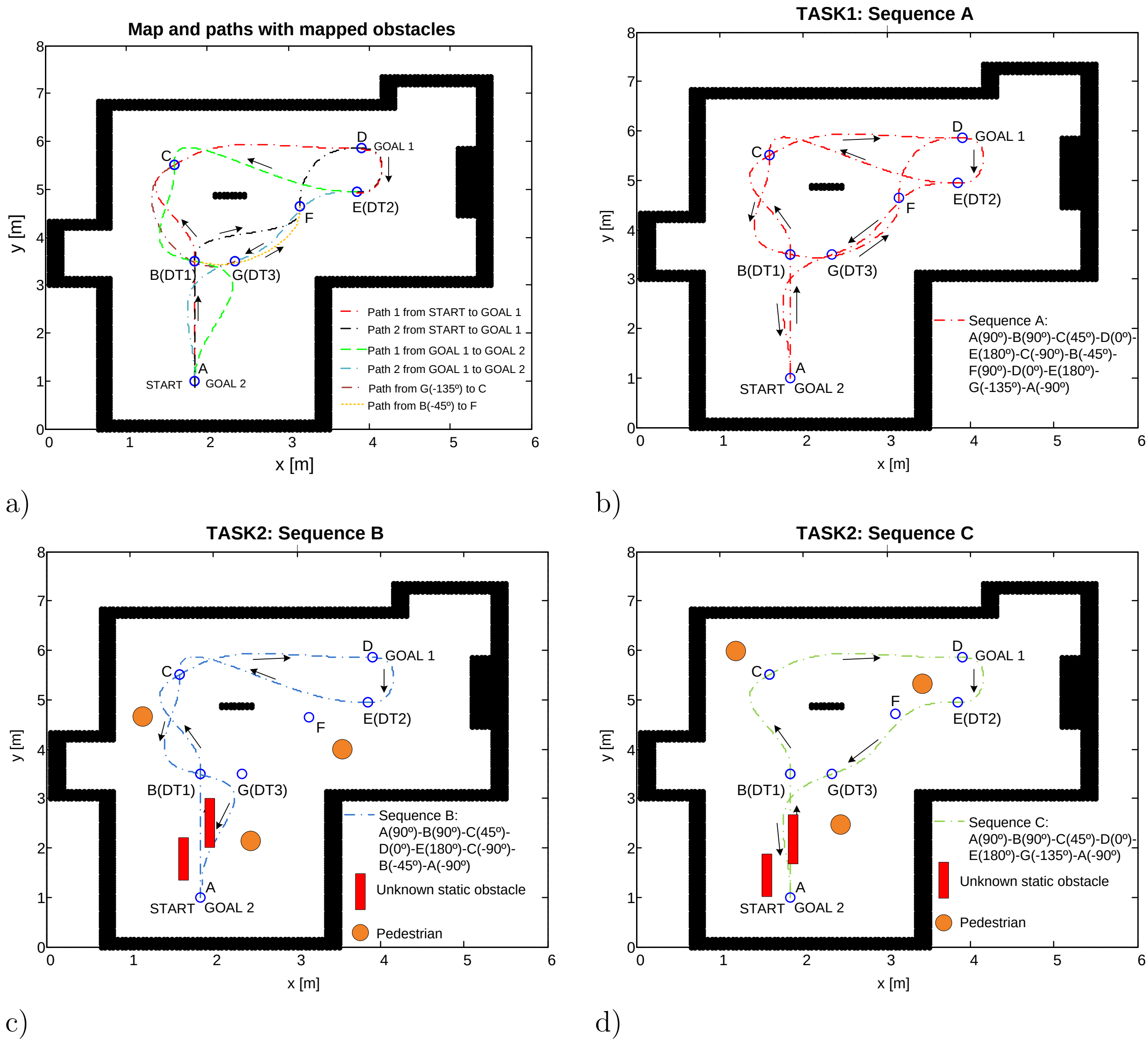

Map and paths with mapped obstacles
y [m]
x [m]
D GOAL 1
C
E(DT2)
F
B(DT1)
G(DT3)
A
START GOAL 2
Path 1 from START to GOAL 1
Path 2 from START to GOAL 1
Path 1 from GOAL 1 to GOAL 2
Path 2 from GOAL 1 to GOAL 2
Path from G(-135º) to C
Path from B(-45º) to F
a)
TASK1: Sequence A
Sequence A:
A(90º)-B(90º)-C(45º)-D(0º)-E(180º)-C(-90º)-B(-45º)-F(90º)-D(0º)-E(180º)-G(-135º)-A(-90º)
b)
TASK2: Sequence B
Sequence B:
A(90º)-B(90º)-C(45º)-D(0º)-E(180º)-C(-90º)-B(-45º)-A(-90º)
Unknown static obstacle
Pedestrian
c)
TASK2: Sequence C
Sequence C:
A(90º)-B(90º)-C(45º)-D(0º)-E(180º)-G(-135º)-A(-90º)
Unknown static obstacle
Pedestrian
d)
0
1
2
3
4
5
6
7
8

Figure 7.1: a) Map and planned paths with only mapped obstacles; b) Sequence A: sequence of waypoints required to perform navigation TASK1; c) Sequence B: first possible sequence of waypoints to perform navigation TASK2; d) Sequence C: second possible sequence of waypoints to perform navigation TASK2.

7.3 Assessment of overall performance

The metrics proposed in [Iturrate 2009], [Montesano 2010] were adopted for the overall performance assessment of the proposed ANS:

- Task success: degree of accomplishment of the navigation task;
- Path length: distance in meters traveled to accomplish the task;

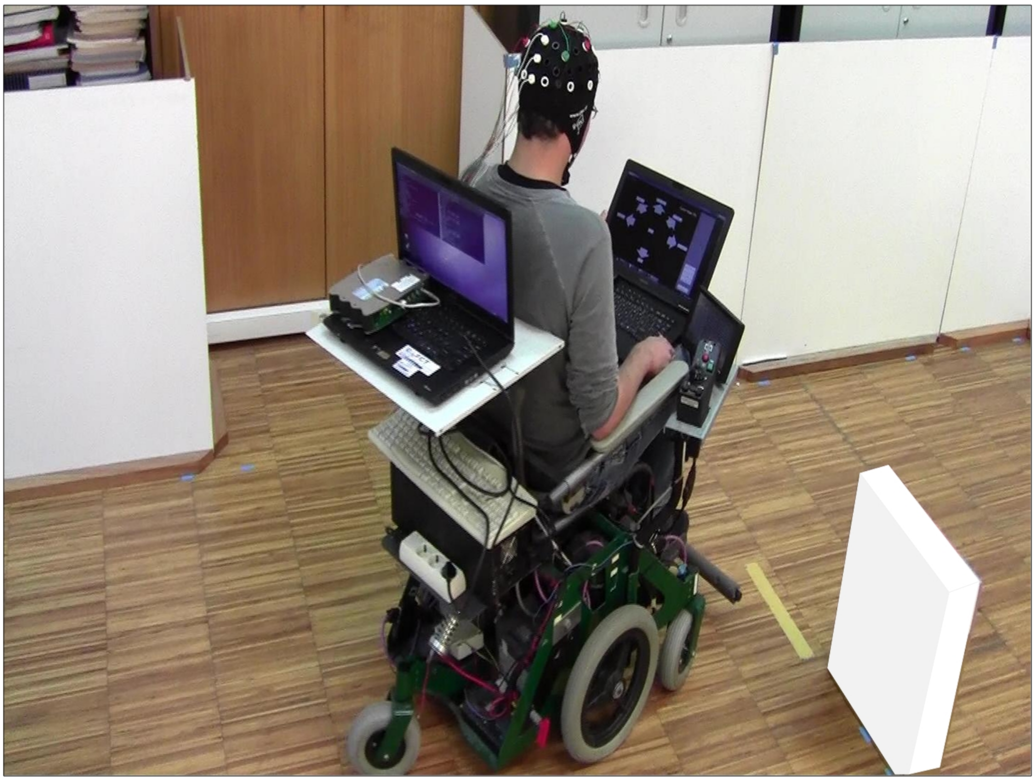


Figure 7.2: Robchair in test scenario.

Table 7.2: Requested waypoint sequences for navigation TASK1 and TASK2 with the predefined orientation for each waypoint. DT stands for Decision Target point.

TASK1		TASK2			
Sequence A	Orientation	Sequence B	Orientation	Sequence C	Orientation
A	↑	A	↑	A	↑
B (DT)	↑	B (DT)	↑	B (DT)	↑
C	↗	C	↗	C	↗
D	→	D	→	D	→
E (DT)	←	E (DT)	←	E (DT)	←
C	↓	C	↓	G (DT)	↙
B (DT)	↘	B (DT)	↘	A	↓
F	↑	A	↓		
D	→				
E (DT)	←				
G (DT)	↙				
A	↓				

Table 7.3: Metrics to evaluate the overall navigation system performance.

	Task 1				Task 2			
Metrics	min	max	mean	std	min	max	mean	std
Task success	1	1	1	0	1	1	1	0
Path length (m)	23.0	31.7	24.6	3.6	21.9	32.3	28.7	3.1
Time (s)	185	333	234	59	324	479	390	46
Path opt. ratio	1.0	1.38	1.07	0.16	1.0	1.65	1.16	0.21
Time opt. ratio	1.61	2.89	2.03	0.51	2.95	4.90	3.25	0.49
Collisions	0	0	0	0	0	0	0	0
BCI accuracy	0.5	0.96	0.77	0.18	0.75	1.0	0.88	0.1

- Time: time taken in seconds to accomplish the task;

- Path length optimality ratio: ratio of the path length to the optimal path (the optimal path length was 23.03 m for TASK1, and 13.24 m for TASK2);

- Time optimality ratio: ratio between the time taken to accomplish the task and the optimal time (the optimal time was calculated assuming an average velocity of 0.2 m/s without stops, resulting in 115 s for TASK1 and 66 s for TASK2);

- Collisions: number of collisions;

- BCI accuracy: accuracy of the pattern-recognition strategy (relation between correct commands and total commands).

According to Table 7.3 all participants were able to accomplish the navigation tasks with success. The path length and the time needed to accomplish both tasks were similar for all participants. The time differences were due to different trial time, (TT), and number of repetitions, N_{rep}, values required by each user to issue BCI commands, and due to wrong command selections by some participants. The path optimality ratio indicates that there was a small difference between the optimal path length and that performed by the participants (1.07 and 1.16 on average for TASK1 and TASK2, respectively, *i.e.* an increase of 10% to 15%). The time optimality results indicate a good overall performance of the navigation and BCI systems. The extra time above the optimal (115 s for TASK1 and 66 s for TASK2 - sequence C) was mainly due to the time required to issue BCI commands. TASK2 has a higher value than TASK1 due to the presence of unknown obstacles, which implies a reduced speed during its contour. Concerning the interaction with the wheelchair using the BCI, results are very satisfactory. A mean BCI accuracy

Table 7.4: Results of online BCI experiments with users moving on the RW, and simultaneously issuing non-decisive and decision commands.

		HR			
Participant	NRep	NDDC	DC	TT (s)	Reached waypoints
1	4	50%	50%	5.9	16
2	4	83%	90%	5.9	12
3	4	92%	90%	5.9	12
4	4	96%	88%	5.9	12
5	4	50%	35%	5.9	12
6	3	85%	100%	4.7	12
7	4	58%	56%	5.9	16
8	5	88%	71%	7.1	12
9	3	92%	100%	4.7	12
10	4	60%	83%	5.9	12
11	4	88%	100%	5.9	12

of 77% was registered for TASK1. To calculate the BCI accuracy in TASK1, a large set of data was gathered (besides the 5 decision commands, a total of 21 non-decisive commands were requested to the users during the RW navigation). For TASK2 a mean BCI accuracy of 88%, was obtained. In this case the participants only had to worry about selecting commands on DT points, which helped significantly the participants to perform the navigation task.

7.4 Assessment of online BCI on road

Results of online BCI performance for TASK1 and TASK2, with known and unknown obstacles, are presented in Tables 7.4 and 7.5, respectively. According to Table 7.4, the majority of the participants, including the motor disabled participant, showed an Hit Rate (HR) higher than 80% for both, combined Non-Decisive and Decision Commands (NDDC), and only Decision Commands (DC). It has to be noticed, that all experiments were performed in a real-world scenario with several types of disturbances, such as: people talking during experiments, mobile phone ringing, dropping objects into the set, etc. The N_{rep} and respective TT to issue a command are also shown in Table 7.4. Most participants were able to issue a command each 5.9 s, including the cerebral palsy participant, but participants 5 and 9 only needed 4.7 s, and participant 8 required 7.1 s. Most participants were able to accomplish TASK1 with the minimum of 12 waypoints, with the exception of participants 1 and 7 requiring 16 waypoints, due to wrong selection of an admissible

Table 7.5: Results of online BCI experiments with users moving on the RW, issuing decision commands for solving bifurcations and multiple directions due to new obstacles in the environment.

Participant	NRep	HR-DC	TT (s)	Reached Waypoints
1	4	78%	5.9	15
2	4	90%	5.9	15
3	4	90%	5.9	15
4	4	100%	5.9	14
5	4	83%	5.9	15
6	3	89%	4.7	15
7	4	75%	5.9	11
8	5	81%	7.1	15
9	2	100%	3.4	15
10	4	80%	5.9	14
11	4	100%	5.9	15

command when reaching a DT point, forcing to the execution of an additional track. The significant performance decay of some participants, when comparing the performance during the calibration and the performance in the wheelchair, may suggest that they have been negatively affected by stressing situations during navigation tasks.

Table 7.5 summarizes TASK2 results of the online BCI experiments. Only decision commands, for solving bifurcations and multiple directions, are considered. The $Nrep$, and the respective TT to issue a command are also presented in Table 7.5. Most participants were able to accomplish the requested navigation task composed by two sequences (4 sequence possibilities: BB, BC, CB, CC), with 15 reached waypoints, with the exception of participants 4 and 10 that only needed 14 waypoints. Participant 7 was only able to accomplish a course (none of the predefined sequences) composed by 11 waypoints, due to wrong selection of an admissible command when reaching a DT point. The motor disabled participant showed an online BCI performance higher than the average online BCI performance of all participants, requiring a number of repetitions similar to able-bodied participants. The preliminary experiments with a cerebral palsy user indicate that the ANS based on BCI has potential to be suited for certain cerebral palsy users.

7.5 Assessment of navigation performance

To evaluate the navigation system performance we used the metrics proposed in [Iturrate 2009, Kuipers 2006, Montesano 2010], namely:

Table 7.6: Metrics to evaluate the navigation system for TASK1.

	Task1			
	min	max	mean	std
Task Success	1	1	1	0
# Waypoints	12	16	12.7	1.6
# Collisions	0	0	0	0
Speed (m/s)	0.17	0.2	0.19	0.01
Time in Motion (s)	115	166	132	18
Clearance min (m)	0.3	0.5	0.4	0.06
Clearance mean (m)	1.99	2.14	2.03	0.05

Table 7.7: Metrics to evaluate the navigation system for TASK2

	Task2			
	min	max	mean	std
Task Success	1	1	1	0
# Waypoints	11	15	15.5	1.6
# Collisions	0	0	0	0
Speed (m/s)	0.10	0.13	0.11	0.01
Time in Motion (s)	207	319	255	34
Clearance min (m)	0.3	0.4	0.34	0.05
Clearance mean (m)	1.89	2.0	1.95	0.04
# Localization uncertainties	0	2	0.73	0.7
# Avoided obstacles	3	6	5.30	0.9
# Multi-direction requests	2	4	2.64	1.1
# Avoided pedestrians	1	2	1.45	0.5

- Task success;
- Collisions;
- Obstacle clearance;
- Number of waypoints.

The results of the assessment of the navigation system performance for TASK1 and TASK2 are shown in Tables 7.6 and 7.7, respectively. The performance of the navigation system was good, since all waypoints were reached without collisions, and all navigation tasks were accomplished in a successful manner. In total, the system reached 299 waypoints and traveled 670 m. There were no collisions during these experiments. The mean of minimum clearance was 0.4 m for TASK1, and 0.34 m for TASK2. This can be considered a good result, since the scenario included very narrow passages, and in case of TASK2, sometimes there were pedestrians quite close to the RW. The mean clearance was around 2 m for both tasks, which indicates that the robot had enough safety margins to carry out obstacle avoidance.

Another indication of safety performance is concerned with adaptability to environments with different constraints [Iturrate 2009]. In TASK1 the average speed was 0.19 m/s, but in TASK2, this value decreased for 0.11 m/s. This result indicates that the ANS was capable to adapt to the environment conditions, reducing in average the speed in TASK2, where maneuverability became more important, for instance during obstacle avoidance. Since polar scan matching is computationally high demanding, it was only applied for odometry correction after the RW traveled a certain distance. This situation could lead to uncertainty in localization, specially when the RW carried out pronounced maneuvers to avoid obstacles in narrow spaces. When a high uncertainty in localization occurs, the RW remains under the control of the local planner, while there is the perception that the RW is out of the reference path. This situation happened sporadically (less than one time per session) for TASK2, and the RW was always capable of recovering its localization with success.

7.6 User assessment

For our collaborative control approach, users are sorted in one of three possible learning stages: beginner, average, and advanced. Since all participants were using the ANS for the first time, all of them were considered as beginners. We used the ANTF approach [Lopes 2009] to evaluate users at the end of the experiments.

7.6.1 Environment and human models

The ANTF with BCI uses two main types of cues for decision-making, as presented in Table 7.8: Type I ($X1$ to $X5$) is related to bifurcations, and Type II ($X6$ to $X8$) is related to obstacle positions (to solve multiple-directions). The system only considers the BCI command in a multiple-direction situation, when more than one steering direction can solve the obstacle situation. In other cases, user aid is not required. A set of rules were established according to the GBPC method described in chapter 5. The models presented in Table 7.9 are simplified, because they only encode possible situations that may occur

Table 7.8: Representation and definition of cues (inputs): Type I (related to bifurcations), and Type II (related to obstacle position), where $A : DT_1 : C$ stands for decision target 1 (DT_1) reached from A and going to C; Representation and definition of judgments (outputs).

Representation	Definition
Type I cues (Inputs)	
X1	$A : DT_1 : C$
X2	$D : DT_2 : C$
X3	$C : DT_1 : A$
X4	$D : DT_2 : G$
X5	$E : DT_3 : A$
Type II cues (Inputs)	
X6	Front Obstacle Ahead
X7	Front Obstacle Right
X8	Front Obstacle Left
Judgments (Outputs)	
Y1	RIGHT90
Y2	RIGHT45
Y3	LEFT90
Y4	LEFT45

in our navigation set. Accordingly, environment and beginner user models ($\hat{Y}e$ and $\hat{Y}s$) were established according to the experiments carried out for a structured unknown environment with new static and moving obstacles in the set. Concerning the environment and user models presented in Table 7.9, there are a few considerations concerned with how the assisted navigation should react to Type II cues (see Table 7.8 and Table 7.9). In case of cue X6 - front obstacle ahead, the system model is established to go around the right with a minimum turning effort. This can be done with a BCI command of RIGHT45. In case of cue X7 - front obstacle in the right side, the navigation system goes around the left with a minimum turning effort (BCI command of LEFT45). For a cue X8 - obstacle in the left side, the navigation system goes around the left with a minimum turning effort (BCI command of RIGHT45). The participants were all beginners, and for that reason, and despite the fact that they were previously instructed on which command they should select for each specific situation, they tend to make choices that are not according to the ideal system model. For instance, a beginner user is normally more concerned with getting away from obstacles than minimizing the turning effort.

According to Table 7.9, $\hat{Y}s$ column represents the model for a beginner user, while

Table 7.9: Environment model $\hat{Y}e$, and beginner user model $\hat{Y}s$ in the experimental scenario.

Cue	Type of Cue	$\hat{Y}e$	$\hat{Y}s$
X6	Type II	$Y2$	$Y1 \cup Y2 \cup Y3 \cup Y4$
X1	Type I	$Y3$	$Y1 \cup Y3$
X2	Type I	$Y1$	$Y1 \cup Y3$
X3	Type I	$Y1$	$Y1 \cup Y3$
X7	Type II	$Y4$	$Y1 \cup Y2 \cup Y3 \cup Y4$
X8	Type II	$Y2$	$Y1 \cup Y2 \cup Y3 \cup Y4$
X1	Type I	$Y3$	$Y1 \cup Y3$
X4	Type I	$Y3$	$Y1 \cup Y3$
X5	Type I	$Y3$	$Y1 \cup Y3$
X7	Type II	$Y4$	$Y1 \cup Y2 \cup Y3 \cup Y4$

$\hat{Y}e$ encodes the most appropriate decision for each cue. We have decided to include all possible options as part of the model for beginner user. This means that we expect that participants will try to select randomly any admissible brain-actuated command for a particular situation. This model reflects the low knowledge level concerning the two main navigation aspects, namely: choosing the best path to accomplish the navigation tasks, low turning effort. Beginner users are usually not concerned with these aspects and their priorities are mostly in avoiding the obstacles as much as far of the obstacle as possible, and in selecting an admissible command.

7.6.2 Assessment of user performance

Table 7.10 shows the results of assisted navigation using BCI in a structured unknown environment. Initially all participants were sorted as beginners. The Rule-based lens model parameters [Yin 2006], environmental predictability, R_e, human control (conformance with human model, see Fig. 7.3), R_s, achievement, ra, modeled knowledge, G, and unmodeled knowledge, C, were computed according to the following expressions (see Fig. 7.3:

$$R_e = \frac{\sum_{i=1}^{n} I_{ei}}{n} \quad , \quad I_{ei} = \left\{ \begin{array}{l} 1 \; if \; Y_{ei} = \hat{Y}_{ei}, \\ 0 \; otherwise \end{array} \right\} \tag{7.1}$$

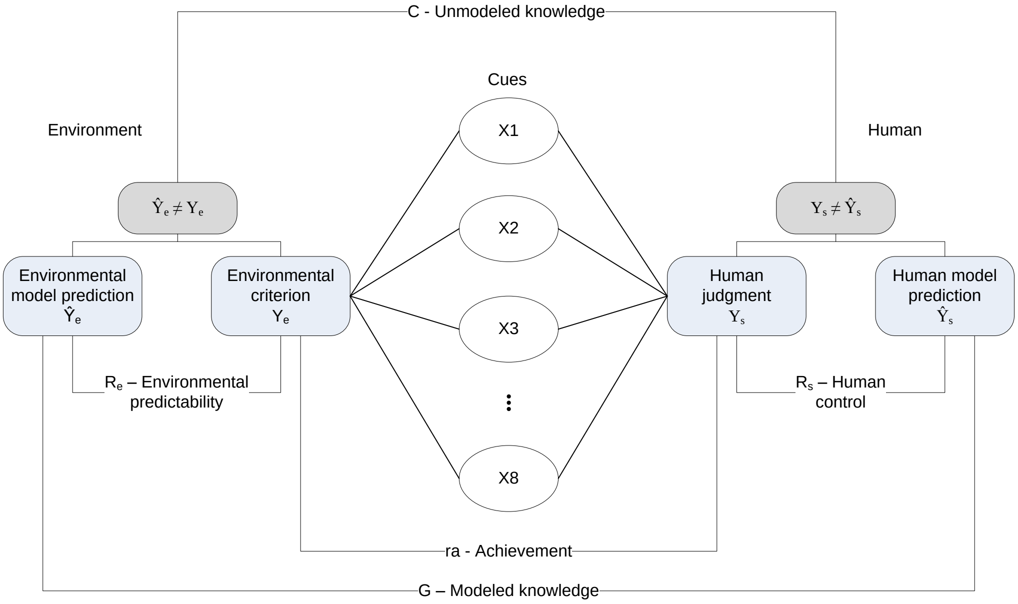


Figure 7.3: Rule-based lens model parameters based on [Yin 2006]: environmental predictability, R_e, human control, R_s, achievement, ra, for achievement, modeled knowledge, G, and unmodeled knowledge, C.

$$R_s = \frac{\sum_{i=1}^{n} I_{si}}{n} \quad , \quad I_{si} = \left\{ \begin{array}{l} 1\ if\ Y_{si} = \hat{Y}_{si}, \\ 0\ otherwise \end{array} \right\} \tag{7.2}$$

$$ra = \frac{\sum_{i=1}^{n} I_{ri}}{n} \quad , \quad I_{ri} = \left\{ \begin{array}{l} 1\ if\ Y_{ei} = Y_{si}, \\ 0\ otherwise \end{array} \right\} \tag{7.3}$$

$$G = \frac{\sum_{i=1}^{n} IG_i}{n} \quad , \quad IG_i = \left\{ \begin{array}{l} 1\ if\ \hat{Y}_{ei} = \hat{Y}_{si}, \\ 0\ otherwise \end{array} \right\} \tag{7.4}$$

$$C = \frac{\sum_{i=1}^{n} IC_i}{n} \quad , \quad IC_i = \left\{ \begin{array}{l} 1\ if\ I_{ei} = I_{si} = 0, \\ 0\ otherwise \end{array} \right\} \tag{7.5}$$

To analyze the results presented in Table 7.10, the following classification grades were taken into consideration for parameters R_e, R_s, ra, and G:

- Excellent for results between 85% and 100%;
- Very Good for results between 75% and 84%;

Table 7.10: Results for assisted navigation using BCI in a structured unknown environment for ten able-bodied participants and one motor disabled. All participants were initially sorted as beginners. R_e stands for environmental predictability, R_s for human control, ra for achievement, G for modeled knowledge, and C for unmodeled knowledge.

Participant	NRep	R_e	R_s	ra	G	C
1	4	100%	78%	62%	40%	0%
2	4	90%	90%	70%	40%	9%
3	4	90%	90%	70%	40%	9%
4	4	80%	100%	100%	40%	20%
5	4	100%	83%	42%	40%	0%
6	3	82%	89%	67%	40%	18%
7	4	89%	75%	62.5%	40%	11%
8	5	100%	81%	45%	40%	0%
9	2	80%	100%	87.5%	40%	20%
10	4	100%	80%	60%	40%	0%
11	4	100%	100%	70%	40%	0%

- Good for results between 65% and 74%;
- Sufficient for results between 50% and 64%;
- Weak for results between 25% and 49%;
- Poor for results under 25%.

According to this classification we can conclude that concerning the human control parameter, all participants show a very good or excellent performance. This parameter, R_s, indicates if the user is acting or not according to the user model. In practice this means that all participants had a good performance in the selection of admissible commands for each specific situation. It is worth to mention that results for parameter R_s are similar to the online BCI results for the parameter DC presented in Table 7.5, which gives the hit rate for the selection of admissible decision commands with BCI. However, if the navigation aspects associated to the environment model are also analyzed, results are not so enthusiastic. Parameter ra gives more than just the selection of a correct BCI command, it also gives an evaluation of how well each participant navigates the wheelchair, since this parameter is the result of comparing user judgment with the environmental criterion that was tailored to minimize turning effort, and optimize the path to reach a predefined goal. Results show that two participants have an excellent achievement rate, four participants have a good achievement rate, three participants have sufficient, and two have a weak achievement rate. The motor disabled participant presented a good result, which is above the average. The environmental predictability R_e should be 100%, if the wheelchair's navigation system worked as expected. According to Table 7.10 this is not always the case. Sometimes, in a multiple-direction situation, the wheelchair did not wait for the user command. Since each user needs a certain amount of time to issue a BCI command (parameter TT), if the multiple-direction situation occurred during this period, and if an admissible, but unintentional, command was accepted, the collaborative controller considered it a valid user's command. This is the reason why parameter C indicates, for some obstacle avoidance cases ($C \neq 0$ in Table 7.10), a certain degree of unmodeled knowledge, since user and environment model outputs did not match in those cases. The ANS did not behave as defined by the environmental model, and the user was not able to provide a command to solve that situation. This problem was already solved with the use of a delay according to the TT of each participant. Parameter G shows the low level of modeled knowledge, which is expected for a beginner user.

In summary, results show that participants are able to move themselves on RobChair using BCI with relative ease, although most of them present some limitations concerning fundamental navigation aspects, namely: choosing the best path to accomplish the navigation tasks, and low turning effort. These results were expected, since all users were experimenting the system for the first time.

Chapter 8

Conclusions and future work

Contents

This thesis presents our research work on human-robot collaboration based on sparse HMIs, such as the P300-based BCI. This work is primarily intended for users with motor disabilities. This chapter summarizes the main contributions and draws some of our lines of future research.

8.1 Conclusion

8.1.1 Main achievements

In this thesis we propose an assistive navigation system based on collaborative control, which mainly uses a P300-based BCI to select steering commands. Since BCI commands are issued sparsely, an assisted navigation architecture based on a two-layer collaborative controller was designed, and implemented in RobChair (ISR-UC wheelchair platform). The collaborative control architecture includes a traded control layer, and a shared control layer that are able to deal with the information provided by the human, which is discrete, sparse, and often unreliable. The proposed collaborative controller complies with four essential design requirements: dialogue, awareness, self-reliance, and adaptiveness.

The focus of this research work was in increasing the mobility of people with severe motor disabilities that have difficulty in operating commercial HMIs, such as a conventional joystick. This type of users is mainly able to operate discrete and sparse HMIs, such as the P300-based BCI described in this thesis. This means that due to the low transfer rates of the P300-based BCI system, the user is only able to issue sparse and discrete commands over time. In this sense, to effectively use such HMI, an ANS able to predict and execute user navigation intents with minimum information, was proposed. This has directly to do with two design requirements: awareness and self-reliance, meaning that the RW must be able to clearly identify situations where aid is required, and ask for help in these situations. Moreover, since the users are only able to issue sparse information over time, the robot must be self-reliant and able to cope with dynamic changes in the environment, without requiring any aid, or if it is really needed, being able to deal with unreliable and delayed information.

Our ANS includes a localization system, an obstacle detection module, and a local planner that were designed to deal with changes in the environment, and give some degree of self-reliance to the RW. We present a new planning strategy that is able to deal with low cluttered semi-structured environments. It also includes a global planner that provides a global path to the ANS. The local planner intervenes if changes in the environment are detected. We proposed a modified VFH that is carried out in three stages: steering, path-planning and blending. Unlike the approaches proposed by [Borenstein 1991, Ulrich 2000], our approach builds a polar histogram directly from laser scan information and does not requiring the construction of the Cartesian local map. Additionally, our approach is also able to blend global and local paths.

We proposed a Markov localization approach that uses dead-reckoning data (odometry) for rough positioning, and laser data for polar scan matching. The Markov localization is carried out in three main stages: scan preprocessing, virtual scanning, and matching. The proposed matching algorithm uses the sample Pearson correlation coefficient to evaluate the similarities between current and virtual scans. The correlation factor is determined on polar coordinate space, leading to a reduction on computational complexity of this algorithm.

Our collaborative controller for assisted navigation is also adaptive, once it depends on the user's skills in steering the assisted robot. For this purpose, user characterization should be carried out in advance with the ANTF system proposed in chapter 5. The ANTF is also intended for training users with disabilities to steer the RW, through a small set of simple commands.

8.1.2 Benefits

Our collaborative control approach helps reduce the impact of operator limitations and variation on system performance. Because it treats the operator as a limited source of planning and information, collaborative control allows the use of human perception and cognition without requiring continuous or time-critical response. If the human cannot respond because he/she is unavailable or busy performing other tasks, the system will still function, without posing the human in a dangerous situation.

The ANS is particularly relevant, in situations in which the robot does not know what to do, or in which it is working poorly. For those cases, a simple human answer is often all that is required to get the robot out of trouble or to perform better. Moreover, this is true regardless of whether the human is a beginner or an advanced user. What is important to understand is that even a beginner can help compensate for insufficient perception or autonomy.

The proposed ANS was successfully tested with eleven participants, 10 able-bodied and one motor disabled that suffers from severe cerebral palsy disorders. All participants were able to accomplish the requested navigation tasks, and the system showed a high level of performance in terms of online BCI accuracy and navigation efficiency. The ANS also gave indications of good adaptability to different navigation scenarios.

8.1.3 Limitations

Although several achievements were attained throughout this thesis, the ANS still poses several limitations:

- BCI offers a low information transfer rate associated with a non-negligible error rate. In other words, the user will be able to provide only sparse commands in time and some of them may be unreliable.

- BCI commands are not issued in a self-paced manner by the user. This topic is being researched to allow the user to issue commands only when he/she desires. This issue is fundamental for more advanced users.

- Currently, the RW stops when new dynamic obstacles (e.g. pedestrians) appear close to it. As a future work we plan to improve the local planner to enable cooperative obstacle avoidance, taking into account the environment perception that includes obstacle positions, humans' intentions, types of obstacles, data from other robots etc.

- Additional sensor data (e.g. laser in the rear part of the wheelchair) is required to improve the RW maneuverability in obstacle avoidance.

- The proposed localization system struggles to keep on tracking in cluttered environments. Additional sensor data is required to obtain a more reliable SLAM system.

- The collaborative controller is limited to a human-robot pair and only takes into account human steering performance for adaptiveness, disregarding other factors such as human mental state or data provided by other robots in the environment.

- Adaptation is not carried out in an automatic manner. This means that at the present stage the human progress steering the RW is not automatically incorporated into the collaborative controller. We plan to automatically tune the efficiency rate ra and user weight parameter η_{UA} by implementing user assessment online.

8.2 Future work

In this thesis we have proposed a two-layer collaborative control approach designed to deal with sparse commands provided by a human agent and adaptable to user skills steering the wheelchair with a specified HMI. This collaborative control approach was tested in a real environment with several participants, with an overall result showing the effectiveness of the system. The current state of the system is the base for improvements and new developments that are described in the sequel.

8.2.1 Extend the ANS to a networked multi-robot system

We propose to research new collaborative control approaches for assistive navigation, to increase autonomy of people with severe motor disabilities, in semi-structured and unstructured dynamically changing environments, with a high level of uncertainty, and being able to interact with human users with different steering skills and mental states. The latest can be evaluated in terms of stress, fatigue, focus, and emotion. Additionally, new local planning techniques able to deal with crowded environments are also to be pursued, and integrated in the new collaborative control approach. A dynamically changing environment includes new static and moving objects, where it is essential to perceive and interact with humans and robots. If all assisted robots of a certain scenario are connected, information can be shared, and used by the collaborative controller of each assisted robot.

In this sense we will research a control scheme that is based not only in the collaboration between an assisted robot and a human, but is also able to collaborate with other human/machine agent pairs, leading to a multi assistive-robot collaborative navigation. Thus, the following goals and methodologies are to be pursued:

- Integration of data to evaluate human user state of mind in terms of stress, fatigue, focus, and emotion into the collaborative control approach.
- Extend the ANS and the ANTF proposed in this thesis, to a networked multi-robot system to allow collaborative navigation among assistive robots and humans. For such purpose new collaborative control algorithms will be adapted for the networked framework. Internal and external agents can provide the following inputs, which are to be considered in the collaborative control approach design:
    - User ability on using the HMI and user skills on steering the wheelchair;
    - User mental state expressed in terms of stress, fatigue, and focus of attention;
    - Automatic adaptation to user variation in terms of steering skills, stress, fatigue, and focus of attention;
    - Situation awareness.
- Research on new local planning approaches able to deal with crowded environments, e.g. environmental scenarios with pedestrians and robots navigating in dynamically changing environment.
- Definition of benchmarks and metrics to effectively evaluate proposed collaborative control methodologies.

8.2.2 Improvements of the localization system

A reliable localization system is paramount for any mobile robot. Therefore, the improvement of the localization system of the proposed ANS is an important goal. Visual SLAM (VSLAM) methods are planned to be developed that will allow determining a robot location from an image or sequence of images, without any prior knowledge of its position. Problems commonly arise when the robot is returning to a previously mapped location after traversing an unmapped area. In such cases the robot's pose estimate can differ markedly from the ground truth. A possible way to solve this problem is to use appearance-based approaches to detect loop closures in a SLAM problem. To accomplish this, the system must be able to deal with the possibility that the current view comes from a previously unvisited place and has no match within the map. Another challenging issue arises from the fact that different parts of the workspace may appear the same to the perception robot's system. This problem is designated as perceptual aliasing. We plan to explore a technique called Fast Appearance Based Mapping, also designated as FAB-MAP [Cummins 2008]. This technique does not require on keeping track of the robot in a metric coordinate system, and is applicable for sparse and discrete observations. This solution seems to be an appropriate method to determine the localization of new robots that are being integrated in a certain world of which they do not have any prior information. In this sense the following goals are to be pursued:

- Research on appearance-based techniques for loop closure detection;
- Development of methods and algorithms to avoid perceptual aliasing;
- Research on techniques to perform localization of new agents in a networked robotic framework.

Appendix A

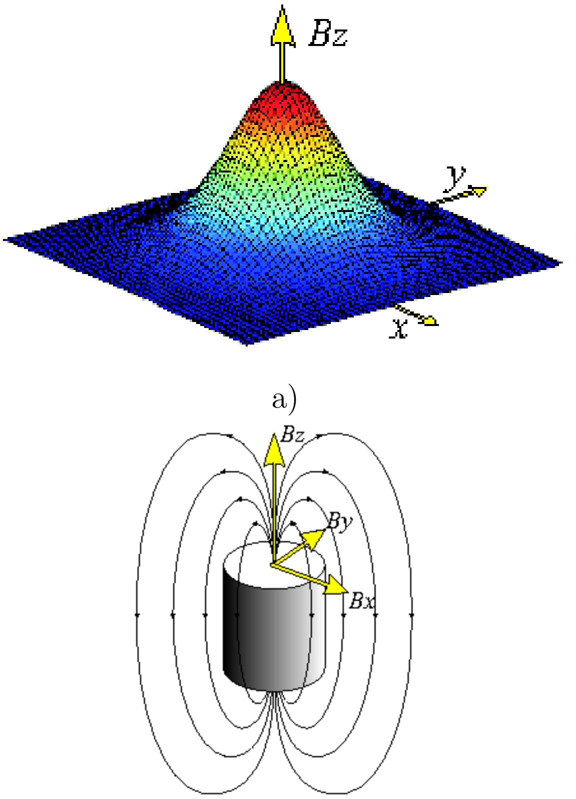
Extended Kalman filter based on the detection of magnetic markers

The ANS also includes a guidepath navigation system that uses permanent magnetic markers embedded in the ground. This methodology can be used for outdoor localization purposes, in situations requiring great precision (e.g. navigating in narrow sidewalk), or areas with poor GPS signal. Additionally, the detection of magnetic markers can also be used for indoor positioning. In this case the magnetic markers work as landmarks for absolute positioning in certain reference sites. The odometric data provided by the wheel encoders is fused with the data from magnetic markers. The system and measurement models involved in fusion process of on-line pose estimation are non-linear, and due to this fact an EKF was adopted [Bento 2005, Nunes 2007]. The RW is equipped with a Magnetic Sensing Ruler (MSR) developed at ISR-UC [Lopes 2007] that is able to perform a robust detection of magnetic markers. The detection is based on a 3-D algorithm that includes Longitudinal Fitting Detection (LFD) and Cross-Fitting Detection (CFD). Both, the LFD and the CFD are based on the Least Squares Fitting (LSF) of the measurement data with the 3-D model of the vertical magnetic field. This MSR system is robust, once it allows the detection of true magnetic markers, and the elimination of noisy magnetic distortions and false detections.

A.1 Detection of magnetic markers

The guidance system is composed by two main subsystems: the magnetic markers (see Fig. A.1) embedded in the ground, defining the center points of the path to be followed; and Hall-effect sensors mounted on the RW for sensing the magnetic field emanating from the markers. The vertical magnetic field was chosen to be measured by the MSR, since it is the strongest component. The 3-D representation of the magnetic field is given by

$$B = \frac{\mu_0 . M}{4\pi r^5} . \left[3xz \overleftarrow{i} + 3yz \overleftarrow{j} + \left(2z^2 - x^2 - y^2 \right) \overleftarrow{k} \right] \tag{A.1}$$



a)

b)

Figure A.1: a) Vertical magnetic field originated by the magnetic marker. b) Magnetic marker and representation of the magnetic field components.

where $r = \sqrt{x^2 + y^2 + z^2}$, μ_0 is the permeability, and M is the magnetic moment of the magnet marker (see Fig. A.1).

The detection of magnetic markers is carried out in three main steps applied to the vertical magnetic field, as shown in Fig. A.2: Kalman-based filtering; time-to-space transformation; and detection. To improve magnetic marker detection, a chained algorithm is implemented. The magnet peak detection is carried out using a 3-D detection strategy that includes: threshold detection, derivative peak finder (DPF), LFD, and CFD.

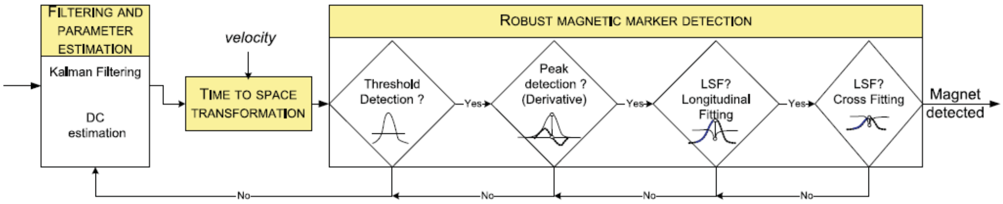

FILTERING AND PARAMETER ESTIMATION
Kalman Filtering
DC estimation
velocity
TIME TO SPACE TRANSFORMATION
ROBUST MAGNETIC MARKER DETECTION
Threshold Detection ?
Yes
Peak detection ? (Derivative)
Yes
LSF? Longitudinal Fitting
Yes
LSF? Cross Fitting
Magnet detected
No
No
No
No

Figure A.2: Algorithm for magnetic markers detection, using velocity information [Lopes 2007].

A.2 Kalman-based filtering

In order to reduce the signal noise, a Kalman filter is applied to the sum of the N values of the vertical magnetic field sensed by each magnetic sensor i, B_z^i:

$$B_z = \sum_{i=1}^{N} B_z^i \tag{A.2}$$

The Kalman filter is composed by the prediction stage,

$$\begin{array}{rcl} \widehat{B}_z^-(k) & = & \widehat{B}_z(k-1) \\ P^-(k) & = & P(k-1)+Q \end{array} \tag{A.3}$$

and correction stage,

$$\begin{array}{rcl} K(k) & = & P^-(k).\left(P^-(k)+R\right)^{-1} \\ \widehat{B}_z(k) & = & \widehat{B}_z^-(k)+K(k).\left(B_z(k)-\widehat{B}_z^-(k)\right) \\ P(k) & = & \left(1-K(k)\right).P^-(k) \end{array} \tag{A.4}$$

where Q represents the process noise variance, and R represents the measurement noise variance. A discrete low-pass filter is applied to the vertical magnetic field estimation, $\widehat{B}_z$, resulting in an approximation to the DC component of the vertical magnetic field value, denoted by $\widehat{B}_z^{DC}$ (see Fig. A.3). This value is used to calculate the threshold levels: $Threshold^+$ and $Threshold^-$.

A.3 Time-to-space transformation

Another important aspect of this algorithm is concerned to the velocity of the vehicle. The spacial sampling along the *x-axis* varies with the RW velocity, i.e. the number of samples of the vertical magnetic field over a magnet decreases with velocity increase. This implies different shapes for the sampled vertical magnetic field curve. To solve this problem, a time-to-space transformation is applied along the *x-axis*, taking into account the vehicle's velocity, in order to adjust the vertical magnetic field shapes.

A.4 Detection

A.4.1 Threshold detection:

Basically, a peak or valley is detected when the vertical magnetic field surpasses the threshold levels, as follows:

$$\begin{array}{ll} \text{if } (\widehat{B}_z(k) > Threshold^{+}) & \Rightarrow \text{Possible Peak} \\ \text{if } (\widehat{B}_z(k) < Threshold^{-}) & \Rightarrow \text{Possible Valley} \end{array} \tag{A.5}$$

A.4.2 Derivative peak finder

After the detection of a possible peak/valley, peak detection is applied using the discrete derivative method. A peak or valley is detected when the derivative of the vertical magnetic field crosses zero, at position denoted $x(J)$, as illustrated in Fig. A.3. A peak is detected when the following condition is validated:

$$\left| K_{dpf}\widehat{B}_z(k) - \sum_{i=1}^{i=K_{dpf}} \widehat{B}_z(k-i) \right| < \delta \tag{A.6}$$

where δ is a small threshold value, and K_{dpf} is a constant value. Peak detection is used to reduce the processing requirements of the LFD algorithm. As shown in Fig. A.2, the LFD algorithm is only applied if a peak detection has occurred.

A.4.3 Longitudinal fitting detection:

The least square fit or least squares estimate is a parameter vector that minimizes the least squared error norm [van der Heijden 2004]. The LSF algorithm is applied to the

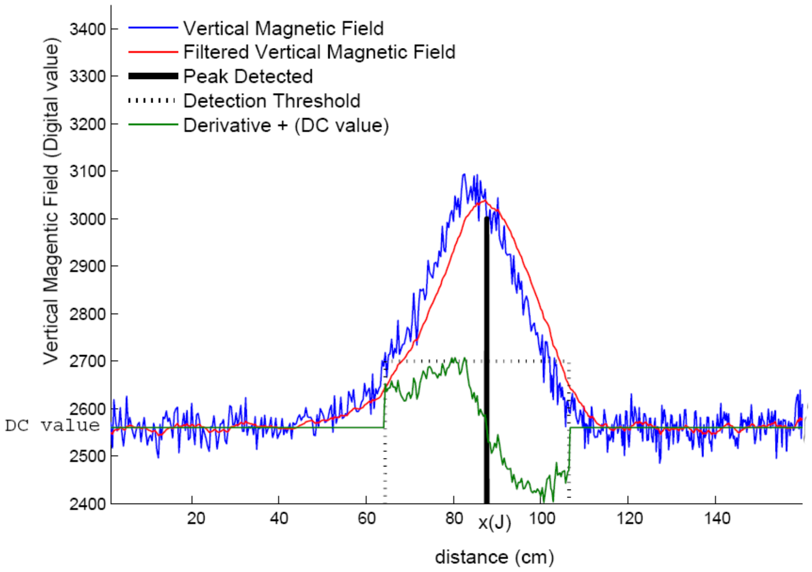


Figure A.3: Results of Kalman-based filtering, threshold detection, and derivative peak finder algorithms.

sum of the 6 higher measurements and considering (A.7), resulting in:

$$ {}_{lg}\widehat{B}_z^m(x) \quad = \quad \textstyle\sum_{i=0}^{5} \frac{\mu_0 M\left(2Z^2 - x^2 - (Y.(i-offset))^2\right)}{4\pi(Z^2+x^2+(Y.(i-offset))^2)^{\frac{5}{2}}} \tag{A.7} $$

where μ_0 is the permeability of the open space, M is the magnetic moment of the magnetic marker, Z and Y are constant values. The *offset* is used to set the origin at the middle of the six Hall-effect sensors. In the presence of a peak or a valley, the vertical field reaches its maximum and minimum value, respectively.

Whenever a peak is detected by the derivative peak finder, point $x(J)$ in Fig. A.3, the LFD algorithm is applied to the backward curve defined by J points $\left\{\widehat{B}_z(0), ..., \widehat{B}_z(J)\right\}$. From the curves defined by the real data and by the model, the LS error norm is calculated as follows:

$$ \|\varepsilon\|^2 = \sum_{j=0}^{J} \varepsilon_j^2 = \sum_{j=0}^{J} \left(\widehat{B}_z(j) -_{lg} \widehat{B}_z^m(j)\right)^2 \tag{A.8} $$

where J depends on the velocity of the RW, since the LFD algorithm is applied to a fixed distance of x. The least square fit is the parameter vector that minimizes the least square error norm (A.8):

$$ \Lambda = argmin_x \|\varepsilon\|^2 \tag{A.9} $$

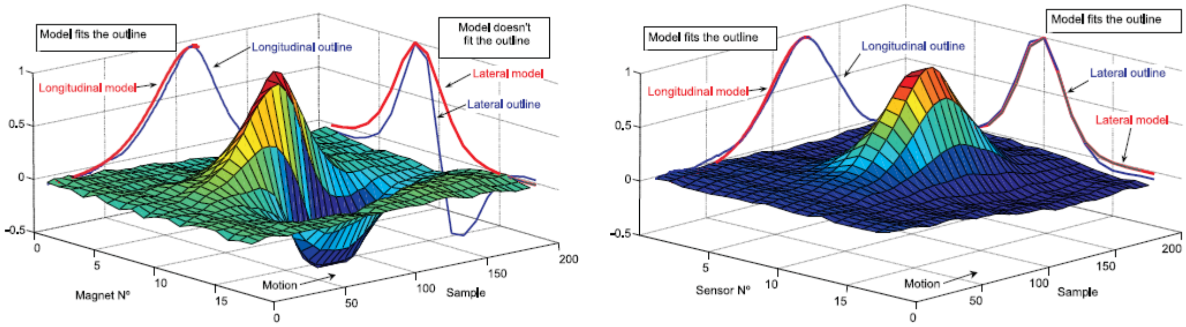


Figure A.4: Implementation of the LFD and CFD algorithms. Left side: detection of a false magnetic marker; Right side. detection of a true magnetic marker.

A.4.4 Cross-fitting detection

The lateral detection or cross-fitting detection algorithm is also based on the fitting of the vertical magnetic field lateral model ${}_{lt}\widehat{B}_z^m$ to the vertical magnetic field $\widehat{B}_z$, which varies along the *y-axis* coordinate. The *x-axis* coordinate is fixed for the peak or valley detected by the LFD algorithm. However, in order to improve the robustness of the lateral algorithm, a sliding window is applied to a few fixed values of x around the peak or valley. In order to apply a LSF to $\widehat{B}_z$, an estimation of lateral vertical magnetic field model is designed, as follows:

$$ {}_{lt}\widehat{B}_z^m(y) = \tfrac{\mu_0 M\left(2Z^2 - X^2 - y^2\right)}{4\pi(Z^2+X^2+y^2)^{\frac{5}{2}}}, \; y = -l\tfrac{N}{2}, ..., l(\tfrac{N}{2}-1) \qquad \text{(A.10)} $$

where Z and X are constant values, and l is the distance between Hall-effect sensors. The CFD algorithm is easier to apply than the LFD algorithm because it is velocity independent. Since the lateral curve has a number of points (N) equal to the number of sensors of the MSR, a normalized function of $\widehat{B}_z$ can be used. The normalized model of ${}_{lt}\widehat{B}_z^m$ has a peak value of 1, for positive magnetic markers, and a valley value of -1 for negative markers. Before the application of the LSF, the N points of the lateral vertical magnetic field $\widehat{B}_z$ must be divided by the peak or valley value, in order to obtain the respective normalized values.

Figure A.4 shows how the implementation of the LFD and CFD algorithms is carried out. A false magnet is detected when the outlines of the longitudinal or the lateral curve of the vertical magnetic field does not fit the respective model (see Fig. A.4). The detection of a true magnetic marker occurs when the outline of both, the longitudinal and lateral curves, fit the respective models.

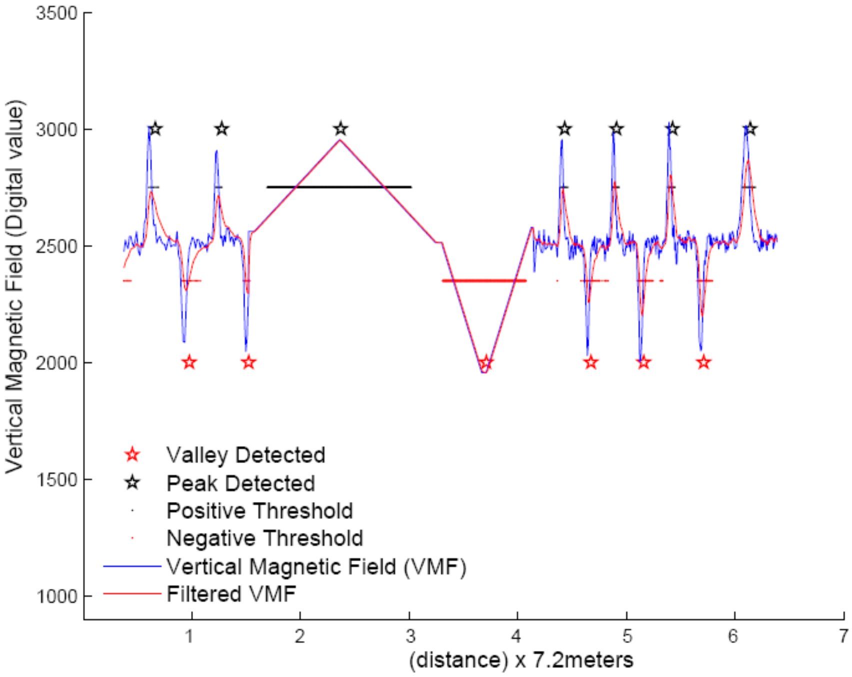


Figure A.5: Results of application of the threshold detection, and DPF algorithms. The false markers at about $\mathbf{x = 2.5}$ and $\mathbf{x = 3.75}$ were not discarded.

A.4.5 Off-line experiments with the MSR

Magnet detection is carried out based on the application of the 3-D algorithm described above. Figure A.5 shows the results of the application of Kalman-based filtering, time-to-space transformation, threshold detection and derivative peak finder algorithms. A value of $K_{dpf} = 5$ was considered for the DPF algorithm. The vertical magnetic field values were obtained for different speeds. To analyze the robustness of the 3-D algorithm, a ramp-like disturbance to simulate false markers was added to the real data.

According to Fig. A.5, by itself, the threshold detection algorithm is not able to detect if the vertical magnetic field curve is originated by a magnetic marker. To eliminate undesired detections of false markers, the LFD and the CFD algorithms are applied, every time a peak, or valley, is found using the DPF method. The results of application of the 3-D algorithm are depicted in Figs. A.6, and A.7.

To estimate the longitudinal vertical magnetic field (A.7), the following values were considered: $Z = 150\ mm$ (height between the magnetic marker and the MSR), $Y = 30\ mm$, and $offset = 2.5$. For the lateral vertical magnetic field (A.10), the following values were considered: $Z = 150\ mm$, $X = 0\ mm$, and $l = 30\ mm$. After the application

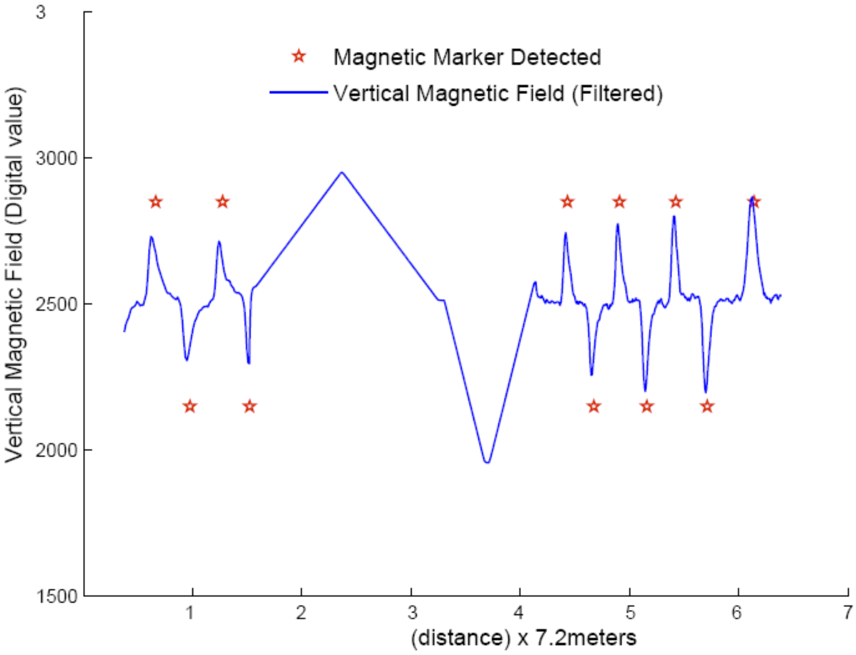


Figure A.6: Results of application of the 3-D detection algorithm. False markers were discarded.

of the LFD and CFD algorithms, the false markers were rejected, and only the appropriate vertical magnetic fields were detected (see Figs. A.6, and A.7). The application of the 3-D detection algorithm has shown good results, as depicted in Fig. A.7 that shows an extended view of the results presented in Fig. A.6.

A.5 Fusion of odometry data with magnetic marker data using an EKF

This section summarizes the method used to fuse odometric data and magnetic marker detection [Bento 2005, Nunes 2007]. In the experiments, magnetic markers were used as beacons, however this methodology is valid for other types of beacons, giving range-bearing data.

A.5.1 Odometry model

The wheelchair is a differential-drive wheeled mobile robot (WMR) with a distance L between the wheel-axis and the front magnetic ruler, and a wheel radius r. The config-

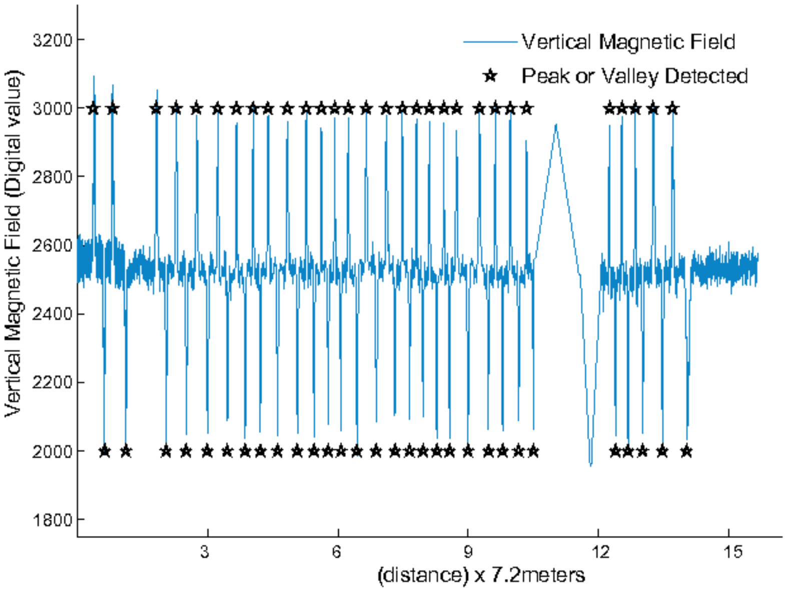


Figure A.7: Extended view of the results of application of the 3-D detection algorithm.

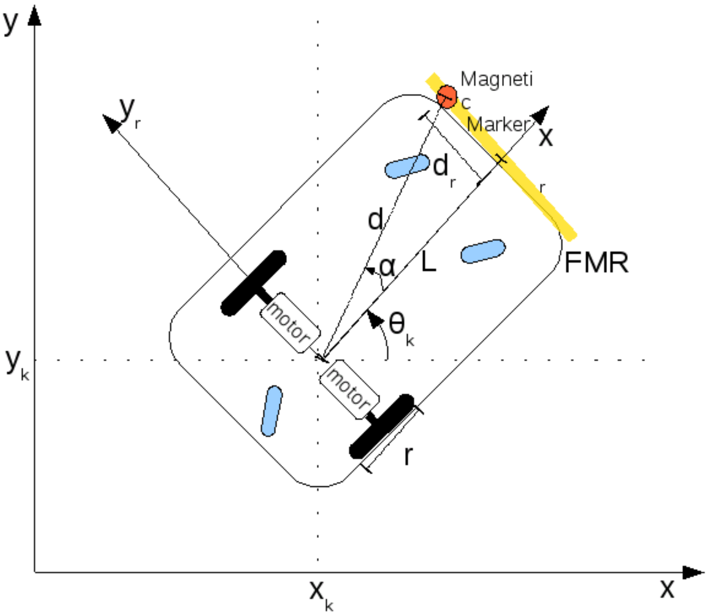


Figure A.8: Wheelchair configuration and measurement models. The Wheelchair is equipped with a FMR.

uration and measurement models are depicted in Fig. A.8. To describe the wheelchair odometric model lets consider its position represented by the central point (x_k, y_k) at the time instant t_k. Additionally the local system coordinate $\{Ox_ry_r\}$ is attached to the

WMR. The linear velocity of the robot is in the direction of the x_r-axis, and θ_k represents the heading direction at instant t_k. The pose evolution of the WMR is given by:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta\cos(\theta_k + \frac{\omega}{2}) \\ y_k + \Delta\sin(\theta_k + \frac{\omega}{2}) \\ \theta_k + \omega \end{bmatrix} \tag{A.11}$$

where Δ is calculated, assuming Δ_r and Δ_l as the result from the right and left encoder measurements, respectively:

$$\Delta = \frac{\Delta_r + \Delta_l}{2} \tag{A.12}$$

and ω represents the elementary rotation from state k to state $k+1$

$$\omega = \frac{\Delta_r - \Delta_l}{2R} \tag{A.13}$$

where R is half of the distance between wheels.

A.5.2 System and measurement models

A.5.2.1 System model

The WMR configuration is defined by $[x\ y\ \theta]^T$, which are also the state variables of the system model. Δ, and ω constitute the inputs of the EKF data fusion process. The system model is defined by the kinematic non-linear equations of (A.11), with state vector $\mathbf{x}_k = [x_k\ y_k\ \theta_k]^T$, and input $\mathbf{u}_k = [\Delta\ \omega]^T$. The system model can be written in the compact form by

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \tag{A.14}$$

where $\mathbf{v_k}$ denotes the system noise, with associated covariance matrix $\mathbf{Q}$ that models the uncertainties of the system model.

A.5.2.2 Measurement model

Let(d, α) be the range-bearing pair, associated to a detected landmark, defined in the local robot coordinate system (see Fig. A.8). Thus the following equations yield

$$\begin{aligned} d &= \sqrt{(y_m - y_k)^2 + (x_m - x_k)^2} \\ \alpha &= \arctan(\tfrac{y_m - y_k}{x_m - x_k}) - \theta_k \end{aligned} \tag{A.15}$$

where (x_m, y_m) represents the Cartesian position of the magnetic marker. Equation (A.15) represents the nonlinear measurement model, which can be written in the compact form, as follows

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \sigma_k \tag{A.16}$$

where $\mathbf{h}(\mathbf{x_k})$ is given by

$$\mathbf{h}(\mathbf{x_k}) \;=\; \begin{bmatrix} \sqrt{(y_m - y_k)^2 + (x_m - x_k)^2} \\ \arctan(\frac{y_m - y_k}{x_m - x_k}) - \theta_k \end{bmatrix} \tag{A.17}$$

and σ_k is considered zero-mean white noise with covariance matrix $\mathbf{R}$. The range-bearing data (d, α) are the observation values entering the EKF, $z = [d\alpha]^T$, which are calculated from sensor measures as follows

$$\begin{aligned} d &= \sqrt{d_m^2 + L^2} \\ \alpha &= \arctan(\tfrac{d_m}{L}) \end{aligned} \tag{A.18}$$

where d_m is the magnetic ruler measure which corresponds to the distance between the marker with known position (x_m, y_m) and the MSR central point, and L is the distance between the front magnetic ruler and the vehicle rear axis (we are assuming that the ruler is perfectly parallel with the y-axis of robot coordinate system).

Equation (A.18) represents the nonlinear measurement model, which can be written in the compact form, as follows:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \sigma_k \tag{A.19}$$

where σ_k is considered zero-mean white noise with covariance matrix $\mathbf{R}$.

A.5.3 EKF Fusion

The EKF algorithm is composed by two main stages: the prediction stage, and the correction stage.

The prediction stage gives an estimate $\widehat{\mathbf{x}}_k^-$ before making any measurement observation:

$$\begin{aligned} \widehat{\mathbf{x}}_k^- &= \mathbf{f}(\widehat{\mathbf{x}}_{k-1}^-, \mathbf{u}_k) \\ \mathbf{P}_k^- &= \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q} \end{aligned} \tag{A.20}$$

where matrix $\mathbf{A}_k$ is calculated as the following Jacobian of the system function $\mathbf{f}(.)$:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0 & -\Delta \sin(\theta_k + \frac{\omega}{2}) \\ 0 & 1 & -\Delta \cos(\theta_k + \frac{\omega}{2}) \\ 0 & 0 & 1 \end{bmatrix} \tag{A.21}$$

Every time a magnet i with known position $\mathbf{z}_k = [x_m^i \; y_m^i]^T$ is detected, the measure d_m is provided to the fusion process to be used in the correction stage, as follows:

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R} \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k)) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k \end{aligned} \tag{A.22}$$

where $\mathbf{I}$ is the identity matrix and the measurement matrix $\mathbf{H}$ is calculated as the following Jacobian of the measurement function $\mathbf{h}(.)$:

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & -d \sin(\theta_k + \alpha) \\ 0 & 1 & d \cos(\theta_k + \alpha) \end{bmatrix} \tag{A.23}$$

A.5.4 Data association

The data association process is based on the innovation sequence and its predicted covariance. The innovation sequence υ_k relates observations $\mathbf{z}_k$ to the predicted observations:

$$\upsilon_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \tag{A.24}$$

The normalized innovation distance can be defined as:

$$\overline{d}_k = \upsilon_k^T \mathbf{S}_k^{-1} \upsilon_k \tag{A.25}$$

where $\mathbf{S}_k$ is the innovation covariance matrix defined in (A.22). If the innovation υ_k has a Gaussian distribution, then $\overline{d}$ is a random variable following the χ^2 distribution. The innovation sequence is the basis of the gate validation technique that accepts observations inside a fixed region of a χ^2 distribution, and rejects the observations making the innovation fall outside these bounds [Bento 2005].

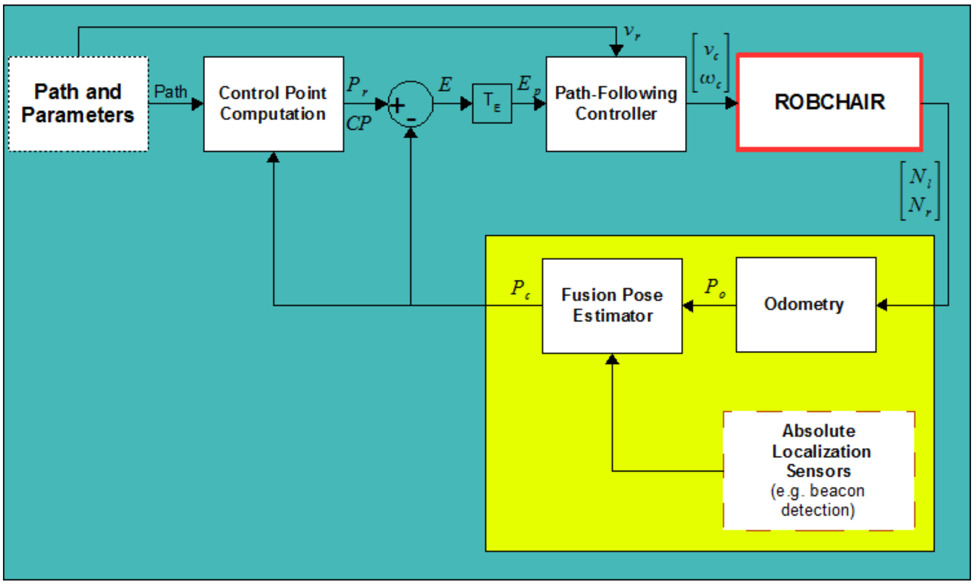

Path and Parameters
Path
Control Point Computation
P_r
CP
+
-
E
T_E
E_p
v_r
Path-Following Controller
$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix}$
ROBCHAIR
$\begin{bmatrix} N_l \\ N_r \end{bmatrix}$
P_c
Fusion Pose Estimator
P_o
Odometry
Absolute Localization Sensors (e.g. beacon detection)

Figure A.9: Path-following architecture with data-fusion based pose estimation module.

A.6 Experimental results

To evaluate the fusion pose estimator, a sliding-mode based path-following controller [Solea 2009, Lopes 2007] was implemented in the Robchair, using the following parameters: $Q = 0.5$, $P = 1.5$, $k_0 = 1$, and $k_1 = 5$. The path-following architecture with a fusion pose estimation module is depicted in Fig. A.9.

To test the fusion pose estimator, an MSR was installed in the RobChair. Every time the MSR detects a magnet, a CAN-message with the magnetic polarity, the distance from the center of the ruler to the magnet, and the magnet number is sent to the PC through CAN-bus. This information is used during the correction stage of the fusion process to correct the cumulative errors originated by the odometry. Figure A.10 shows the path-following results for a line segment path of 5 m. A constant linear speed of 0.6 m/s was used in the experiments. Figure A.10 clearly shows that without the fusion pose estimator the wheelchair deviates significantly from its goal path. Without correction, the lateral error is about 20 cm when the wheelchair reaches the last magnet. With the EKF pose estimator, the lateral error is very small along the path. Figure A.11 shows the path-following results for an ellipsoidal-like path, with around 27 m, using the sliding-mode controller and EKF correction. The wheelchair is able to follow prescribed paths within a given range of velocities, with reduced lateral errors along the path.

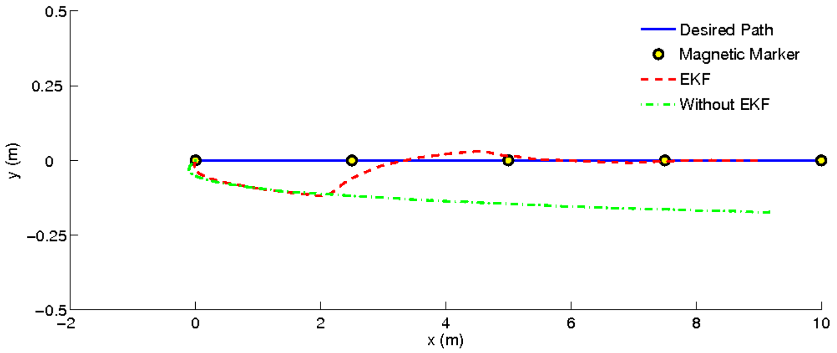


Figure A.10: Path-following results for a line segment path of **5m**.

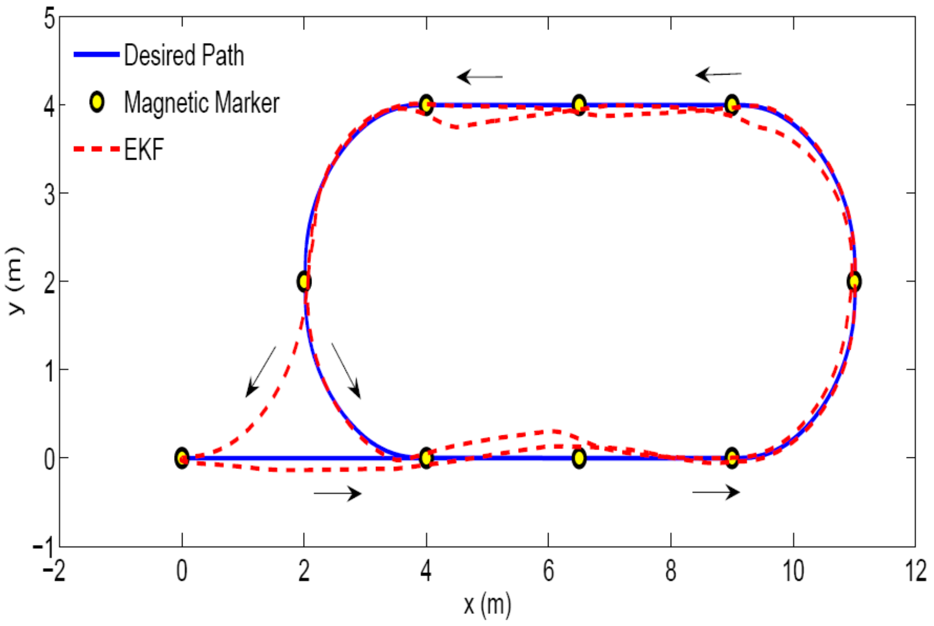


Figure A.11: Path-following results for an ellipsoidal-like path.

References

[A. Huntemann 2008] M. Nuttin A. Huntemann E. Demeester and H. V. Brussel. *Online user modeling with Gaussian processes for Bayesian plan recognition during power-wheelchair steering.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'08), Nice, France, Sep 2008.

[Aigner 1997] P. Aigner and B. McCarragher. *Contrasting Potential Fields and Constraints in a Shared-Control Task.* In 1997 International Conference on Intelligent Robots and Systems (IROS'97), 1997.

[Aigner 1999] P. Aigner and B. McCarragher. *Shared control framework applied to a robotic aid for the blind.* IEEE Control Systems Magazine, vol. 19, no. 2, pages 40–46, 1999.

[Arkin 1991] R. C. Arkin. *Reactive control as a substrate for telerobotic systems.* IEEE AES Systems Magazine, vol. 6, no. 6, pages 24–31, June 1991.

[Barraquand 1991] J. Barraquand and J.C. Latombe. *Robot motion planning: a distributed representation approach.* Int. J. Rob. Res., vol. 10, no. 6, pages 628–649, December 1991.

[Bax 2005] M. Bax, M. Goldstein, P. Rosenbaum, A. Leviton, N. Paneth, B. Dan, B. Jacobsson and D. Damiano. *Proposed definition and classification of cerebral palsy.* Developmental Medicine Child Neurology, vol. 47, no. 8, pages 571–576, 2005.

[Bento 2005] L.C. Bento, U.Nunes, F. Moita and A. Surrécio. *Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments.* In In Proc. of IEEE Intelligent Transportation Systems, pages 245 – 250, September 2005.

[Billings 1997] Charles E. Billings. Aviation automation: the search for human-centered approach. Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 1997.

[Biswas 2011] P. Biswas and P. Langdon. *A new input system for disabled users involving eye gaze tracker and scanning interface.* Journal of Assistive Technologies, vol. 5, no. 2, pages 58–66, 2011.

[Borenstein 1991] J. Borenstein and Y. Koren. *The vector field histogram-fast obstacle avoidance for mobile robots.* IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pages 278 –288, June 1991.

[Boy 2002] E. Boy, C. Teo and E. Burdet. *Collaborative wheelchair assistant.* In Proc. of the IEEE/RJS International Conference on Robotics and Intelligent Systems (IROS'02), volume 30, pages 1511–1516, Lausanne, Suisse, 2002.

[Braynov 2003] S. Braynov and H. Hexmoor. Agent autonomy, chapter Quantifying Relative Autonomy in Multiagent Interaction, pages 43–56. Springer, 2003.

[Brunner 1993] B. Brunner, G. Hirzinger, K. Landzettel and J. Heindl. *Multisensory Shared Autonomy and Tele-Sensor Programming - Key Issues in the Space Robot Technology Experiment ROTEX.* In Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'93), 1993.

[Brunswik 1955] E. Brunswik. *Representative design and probabilistic theory in a functional psychology.* Psychological Review, vol. 62, pages 193–217, 1955.

[Burgard 1999] W. Burgard, D. Fox, H. Jans, C. Matenar and S. Thrun. *Sonar-Based Mapping With Mobile Robots Using EM.* In International Conference on Machine Learning, 1999.

[Burke 2004] J. L. Burke, R. R. Murphy, E. Rogers, V. J. Lumelsky and J. Scholtz. *Human-Robot Interaction: Final Report for DARPA/NSF Study on Human-Robot Interaction.* IEEE Transactions on Systems, Man and Cybernetics, Part C. Applications and Reviews, vol. 34, no. 2, pages 103–112, May 2004.

[C. Lin 2006] W. Chen C. Chiu M. Yeh C. Lin C. Ho. *Powered wheelchair controlled by eye-tracking system.* Optica Applicata, vol. 36, no. 2-3, pages 401–412, 2006.

[Carlson 2011a] T. Carlson, G. Monnard, R. Leeb and J. del R. Millán. *Evaluation of proportional and discrete shared control paradigms for low resolution user inputs.* In IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 1044 –1049, October 2011.

[Carlson 2011b] T.E. Carlson, G. Monnard and J. del R. Millán. *Vision-Based Shared Control for a BCI Wheelchair.* International Journal of Bioelectromagnetism, vol. 13, no. 1, pages 20–21, 2011.

[Carlson 2012] T. Carlson and Y. Demiris. *Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload.* IEEE Transactions on Systems, Man, and Cybernatics - Part B: Cybernetics, vol. 42, no. 3, pages 876–888, 2012.

[Casanova 2008] J. L. Casanova. *Pessoas com deficiência e incapacidades: um inquérito nacional (in Portuguese).* In VI Congresso Português de sociologia, 301, Lisbon, Portugal, June 2008.

[Casper 2003] J. Casper and R. R. Murphy. *Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center.* IEEE Transactions on Systems, Man and Cybernetics, Part B. Cybernetics, vol. 33, no. 3, pages 1–17, June 2003.

[Cen 2002] *Census 2001: Análise da população com deficiência (in Portuguese).* Technical report, Instituto Nacional de Estatística (INE), 2002.

[Chaves 2004] E. S. Chaves, M. L. Boninger, R. Cooper, S. G. Fitzgerald, D. B. Gray and R. A. Cooper. *Assessing the influence of wheelchair technology on perception of participation in spinal cord injury.* Archives of Phisical Medecine and Rehabilitation, vol. 85, pages 1854–1858, 2004.

[Cheng 2002] P. Cheng and S.M. LaValle. *Resolution complete rapidly-exploring random trees.* In Proc. IEEE International Conference on Robotics and Automation (ICRA'02), volume 1, pages 267 – 272, 2002.

[Cox 1991] I.J. Cox. *Blanche-an experiment in guidance and navigation of an autonomous robot vehicle.* IEEE Transactions on Robotics and Automation, vol. 7, no. 2, pages 193–204, April 1991.

[Cox 1996] I. J. Cox and S. L. Hingorani. *An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 2, pages 138 –150, February 1996.

[Crandall 2002] J. W. Crandall and M. A. Goodrich. *Characterizing Efficiency of Human Robot Interaction: A Case Study of Shared-Control Teleoperation.* In 2002 International Conference on Intelligent Robots and Systems (IROS'02), 2002.

[Cummins 2008] M. Cummins and P. Newman. *FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance.* Int. J. Rob. Res., vol. 27, no. 6, pages 647–665, June 2008.

[Curry 1976] R. E. Curry and A. R. Ephrat. Monitoring behaviour and supervisory control, chapter Monitoring and control of unreliable systems, pages 193–206. Plenum press, 1976.

[Davison 2002] A.J. Davison and D.W. Murray. *Simultaneous localization and map-building using active vision.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pages 865–880, July 2002.

[Demeester 2008] E. Demeester, A. Hüntemann, D. Vanhooydonck, G. Vanacker, H. Van Brussel and M. Nuttin. *User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving.* Auton. Robots, vol. 24, no. 2, pages 193–211, February 2008.

[Dijkstra 1959] E. W. Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, vol. 1, pages 269–271, 1959. 10.1007/BF01386390.

[Diosi 2005] A. Diosi and L. Kleeman. *Laser scan matching in polar coordinates with application to SLAM.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'05), Alberta, Canada, 2005.

[Dissanayake 2002] G. Dissanayake, S. B. Williams, H. Durrant-Whyte and T. Bailey. *Map Management for Efficient Simultaneous Localization and Mapping (SLAM).* Auton. Robots, vol. 12, no. 3, pages 267–286, May 2002.

[Donegan 2009] M. Donegan, J. D. Morris, C. Fulvio, I. Signorile, A. Chio, V. Pasian, A. Vignola, M. M. Buchholz and E. Holmqvist. *Understanding users and their needs.* Universal Access in the Information Society - Special Issue: Communication by Gaze Interaction, vol. 8, no. 2, pages 259–275, 2009.

[Duchowski 2007] A. T. Duchowski. Eye tracking methodology, theory and practice. Springer, 2007.

[Durrant-Whyte 1997] H. Durrant-Whyte. *An autonomous guided vehicle for cargo handling applications.* In O. Khatib and J. Salisbury, editors, Experimental Robotics IV, volume 223 of *Lecture Notes in Control and Information Sciences*, pages 372–379. Springer Berlin / Heidelberg, 1997. 10.1007/BFb0035227.

[Durrant-Whyte 2006] H. Durrant-Whyte and T. Bailey. *Simultaneous localization and mapping: part I.* IEEE Robotics Automation Magazine, vol. 13, no. 2, pages 99 –110, June 2006.

[Einhorn 1974] H.J. Einhorn. The use of nonlinear, noncompensatory models in decision making. University Microfilms, 1974.

[Ellis 1996] R. E. Ellis, O. M. Ismaeil and M. G. Lipsett. *Design and Evaluation of a High-Performance Haptic Interface.* Robotica, vol. 4, pages 321–327, 1996.

[Enes 2010] Aaron Enes and Wayne Book. *Blended Shared Control of Zermelo's Navigation Problem.* In Proc. 2010 American Control Conference (ACC2010), Baltimore, MD, USA, June 2010.

[Farwell 1988] L.A. Farwell and E. Donchin. *Talking off the top of your head: toward a mental prosthesis utilizing event related brain potentials.* Electr. and Clin. Neuroph., vol. 70, no. 6, pages 510–523, 1988.

[Fehr 2000] L. Fehr, W. E. Langbein and S. B. Skaar. *Adequacy of power wheelchair control interfaces for persons with severe disabilities : A clinical survey.* Journal of Rehabilitation Research and Development, vol. 37, no. 3, pages 353–360, 2000.

[Feil-Seifer 2005] D. Feil-Seifer and M. J. Matarić. *Defining Socially Assistive Robotics.* In Proc. IEEE 9th Int. Conf. on Rehabilitation Robotics (ICORR'05), pages 465–468, 2005.

[Fejtová 2009] Marcela Fejtová, Luis Figueiredo Petr, Novák, Olga Štěpánková and Ana Gomes. *Hands-free interaction with a computer and other technologies.* Universal Access in the Information Society, vol. 8, pages 277–295, 2009.

[Fong 2001a] T. Fong. *Collaborative control: a robot-centric model for vehicle teleoperation.* Technical report, Carnegie Mellon University, 2001.

[Fong 2001b] T. Fong, C. Thorpe and C. Baur. *Collaboration, Dialogue, and Human-Robot Interaction.* In 10th International Symposium of Robotics Research, Victoria, Australia, November 2001.

[Fong 2002] T. Fong, C. Thorpe and C.Baur. *Collaboration, Dialogue, and Human-Robot Interaction.* In In 10th International Symposium of Robotics Research. Springer-Verlag, 2002.

[Fox 1997] D. Fox, W. Burgard and S. Thrun. *The dynamic window approach to collision avoidance.* IEEE Robotics Automation Magazine, vol. 4, no. 1, pages 23 –33, March 1997.

[Fox 1999] D. Fox, W. Burgard and S. Thrun. *Markov localization for mobile robots in dynamic environments.* Journal of Artificial Intelligence Research, vol. 11, pages 391–427, 1999.

[Galán 2008] F. Galán, M. Nuttin, E. Lew, P.W. Ferrez, G. Vanacker, J. Philips and J. del R. Millán. *A brain-actuated wheelchair: Asynchronous and non-invasive Brain-computer interfaces for continuous control of robots.* Clinical Neurophysiology, vol. 119, no. 9, pages 2159 – 2169, 2008.

[Gao 2011] Y. Gao, M. Sedef, A. Jog, P. Peng, M. Choti, G. Hager, J. Berkley and R. Kumar. *Towards Validation of Robotic Surgery Training Assessment across Training Platforms.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'11), 2011.

[Gerkey 2003] B. Gerkey, R. T. Vaughan and A. Howard. *The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems.* In Proc. IEEE Int. Conf. on Advanced Robotics (ICAR 2003), Coimbra, Portugal, 2003.

[Ghosh 2011] M. Ghosh and F. Tanaka. *The Impact of Different Competence Levels of Care-Receiving Robot on Children.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'11), 2011.

[Giminez 2003] A. Giminez, C. Balaguer, S. M. Sabatini and V. Genovese. *The MATS robotic system to assist disabled people in their home environments.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'03), pages 2612–2617, 2003.

[Giralt 1997] G. Giralt, R. Chatila and R. Alami. *Remote intervention, robot autonomy, and teleprogramming: generic concepts and real world application cases.* In 1997 International Conference on Intelligent Robots and Systems (IROS'97), Yokohama, Japan, July 1997.

[Goodrich 2003] M.A. Goodrich and D.R. Olsen. *Seven principles of efficient human robot interaction.* In IEEE International Conference on Systems, Man and Cybernetics, volume 4, pages 3942–3948, October 2003.

[Grasse 2010] R. Grasse, Y. Morère and A. Pruski. *Assisted navigation for persons with reduced mobility: path recognition through particle filtering (condensation algorithm).* Journal of Intelligent and Robotic Systems, no. 60, pages 19–57, March 2010.

[Grimbergen 2004] C. A. Grimbergen and J. Jaspers. *Robotics in minimally invasive surgery.* In 2004 International Conference on Systems, men and cybernetics, pages 2486–2491, 2004.

[Gross 2011] H. Gross, C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, C. Martin, M. Merten and T. Langner. *Progress in Developing a Socially Assistive Mobile Home Robot Companion for the Elderly with Mild Cognitive Impairment.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'11), 2011.

[Gutmann 1998] J.-S Gutmann, W. Burgard, D. Fox and K. Konolige. *An experimental comparison of localization methods.* In In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98), volume 2, pages 736–743, October 1998.

[Gutmann 2002] J.-S Gutmann and D. Fox. *An experimental comparison of localization methods continued.* In In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02), volume 1, pages 454 – 459, 2002.

[Haegele 2001] M. Haegele, J. Neugebauer and R. D. Schraft. *From Robots to Robot Assistants.* In 32nd International Symposium on Robotics, Seoul, Japan, April 2001.

[Hammond 1975] K.R. Hammond, T.R. Stewart, B. Brehmer and D.0. Steinmann. Human judgment and decision processes, chapter Social judgment theory. New York: Academic Press, 1975.

[Hart 1968] P.E. Hart, N.J. Nilsson and B. Raphael. *Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics, vol. 4, pages 100–107, July 1968.

[Hoppenot 2002] P. Hoppenot and E. Colle. *Mobile Robot Command by Man-Machine Co-Operation - Application to Disabled and Elderly People Assistance.* Journal of Intelligent and Robotic Systems archive, vol. 34, no. 3, pages 235–252, July 2002.

[Hu 2010] Z. Hu, L. Li, Y. Luo, Y. Zhang and X. Wei. *A novel intelligent wheelchair control approach based on head gesture recognition.* In Proc. 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, October 2010.

[Huang 2012] D. Huang, K. Qian, D.-Y. Fei, W. Jia, X. Chen and O. Bai. *Electroencephalography (EEG)-Based Brain-Computer Interface (BCI): A 2-D Virtual Wheelchair Control Based on Event-Related Desynchronization/Synchronization and State Control.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 20, no. 3, pages 379 –388, may 2012.

[Iturrate 2009] I. Iturrate, J.M. Antelis, A. Kubler and J. Minguez. *A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation.* IEEE Transactions on Robotics, vol. 25, no. 3, pages 614–627, June 2009.

[J. Shen 2004] J. Ibanez-Guzman J. Shen, T. Ng and B. Sengchew. *A Collaborative Shared-Control System With Safe Obstacle Avoidance Capability.* In Proc. IEEE Int. Conf. on on Robotics, Automation and Mechatronics, 2004.

[Jensfelt 2002] P. Jensfelt, H.I. Christensen and G. Zunino. *Integrated Systems for Mapping and Localization.* In J. Leonard and H. Durrant-Whyte, editors, ICRA-02 SLAM Workshop. IEEE, May 2002.

[Jensfelt 2006] P. Jensfelt, D. Kragic, J. Folkesson and M. Bjorkman. *A framework for vision based bearing only 3D SLAM.* In Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), pages 1944–1950, May 2006.

[Jones 2007] M. W. Jones, E. Morgan, J. E. Shelton and C. Thorogood. *Cerebral Palsy: Introduction and Diagnosis (Part I).* J. Pediatric Health Care, vol. 21, no. 3, pages 146–152, 2007.

[K. Wada 2002] T. Saito K. Wada T. Shibata and K. Tanie. *Analysis of factors that bring mental effects to elderly people in robot assisted activity.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'02), 2002.

[Kavraki 1996] L.E. Kavraki, P. Svestka, J.C. Latombe and M.H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces.* IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pages 566 –580, August 1996.

[Kawamura 1995] K. Kawamura, S. Bagchi, M. Iskarous and Magued Bishay. *Intelligent Robotic Systems in service of the Disabled.* IEEE Transactions on Rehabilitation Engineering, vol. 3, no. 1, pages 14–21, March 1995.

[Khatib 1985] O. Khatib. *Real-time obstacle avoidance for manipulators and mobile robots.* In Proceedings IEEE International Conference on Robotics and Automation, volume 2, pages 500 – 505, March 1985.

[Kneip 2009] L. Kneip, F. Tache, G. Caprari and R. Siegwart. *Characterization of the compact Hokuyo URG-04LX 2D laser range scanner.* In IEEE International Conference on Robotics and Automation, 2009. ICRA '09., pages 1447 –1454, may 2009.

[Kosuge 2004] K. Kosuge and Y. Hirata. *Human-robot interaction.* In Proc. 2004 IEEE International Conference on Robotics and Biomimetics, Shenyang, China, August 2004.

[Kriegger 2006] K. Kriegger. *Cerebral Palsy: An Overview.* American Family Physician, vol. 73, no. 1, pages 91–100, 2006.

[Kubler 2008] A. Kubler and N. Birbaumer. *Brain-computer interfaces and communication in paralysis: Extinction of goal directed thinking in completely paralysed patients?* Clinical Neurophysiology, vol. 119, no. 11, pages 2658 – 2666, 2008.

[Kuipers 2006] B. Kuipers. *Building and evaluating and intelligent wheelchair.* Technical report, University of Texas at Austin, 2006.

[Latombe 1990] J.C. Latombe. Robot motion planning. Kluwer international series in engineering and computer science: Robotics. Springer, 1990.

[Laumond 1998] J.P. Laumond. Robot motion planning and control. Lecture Notes in Control and Information Sciences. Springer, 1998.

[LaValle 1999] S.M. LaValle and J.J. Kuffner. *Randomized kinodynamic planning.* In Proc. IEEE International Conference on Robotics and Automation (ICRA'99), volume 1, pages 473 –479, 1999.

[LaValle 2006] S.M. LaValle. Planning algorithms. Cambridge University Press, 2006.

[Law 2002] K. Law and Y. Xu. *Shared-Control for Navigation and Balance of a Dinamically Stable Robot.* In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'02), volume 2, Washington, DC, USA, May 2002.

[Leduc 2000] B. Leduc and Y. Lepage. *Health-related quality of life after spinal cord injury.* Journal of Disability ans Rehabilitation, vol. 24, no. 4, pages 196–202, 2000.

[Lee 1993] S. Lee. *Intelligent Sensing and Control for Advanced Teleoperation.* IEEE Control Systems Magazine, vol. 13, no. 3, pages 19–28, June 1993.

[Lee 2006] S. Lee, Y. Choi, K. Jeong and S. Jung. *Development of a Tele-operated Underwater Robotic System for maintaining a light-water type power reactor.* In SICE-ICASE International Joint Conference, 2006.

[Leonard 1992] J. J. Leonard, H. F. Durrant-Whyte and J. I. Cox. *Dynamic map building for an autonomous mobile robot.* Int. J. Rob. Res., vol. 11, no. 4, pages 286–298, August 1992.

[Levine 1999] S. Levine, D. Bell, A. Jaros, R. Simpsons, Y. Koren and J. Borenstein. *The NavChair assistive wheelchair navigation system*. IEEE Transactions on Rehabilitation Engineering, vol. 7, no. 4, pages 443–451, May 1999.

[Li 2011] Q. Li, W. Chen and J. Wang. *Dynamic shared-control for human-wheelchair cooperation*. In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'11), Shangai,China, 2011.

[Lopes 2007] A. Lopes, F. Moita, U. Nunes and R. Solea. *An outdoor guidepath navigation system for AMRs based on robust detection of magnetic markers*. In Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2007), Patras, Greece, 2007.

[Lopes 2009] A. Lopes, U. Nunes and J. Matsuura. *A Rule-Based Lens Modeling Approach for Assisted Navigation Training*. In Proc. IEEE Int. Conf. on Industrial Electronics Society (IECON'09), Porto, Portugal, November 2009.

[Lopes 2011] A. Lopes, G. Pires, L. Vaz and U. Nunes. *Wheelchair Navigation Assisted by Human-Machine Shared-control and a P300-based BCI*. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS11), San Francisco, USA, September 2011.

[Lopes 2012a] A. Lopes, G. Pires and U. Nunes. *RobChair: Experiments Evaluating Brain-Computer Interface to Steer a Semi-autonomous Wheelchair*. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS12), Vilamoura, Portugal, October 2012.

[Lopes 2012b] A.C. Lopes, G. Pires and U. Nunes. *Assisted Navigation for a Brain-actuated Intelligent Wheelchair Robotics and Autonomous Systems*. Robotics and Autonomous Systems, 2012. Accepted for publication.

[Lu 1994] F. Lu and E. E. Milios. *Robot pose estimation in unknown environments by matching 2D range scans*. In In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94), pages 935 –938, June 1994.

[Mallet 2002] P. Mallet and G. Schöner. *WAD project where attractor dynamics aids wheelchair navigation*. In Proc. IEEE International Conference on Intelligent Robots and Systems (IROS'02), 2002.

[Matsumoto 2001] Y. Matsumoto, T. Ino and T. Ogsawara. *Development of Intelligent wheelchair system with face and gaze based interface*. In Proc. of the 10th IEEE International Workshop on Robot and Human Intereactive Communication, 2001.

[McLachlan 2005] S. McLachlan, J. Arblaster, D. K. Liu, J. V. Miro and L. Chenoweth. *A Multi-Stage Shared-Control Method for an Intelligent Mobility Assistant*. In Proc. IEEE Int. Conf. on Rehabilitation Robotics (ICORR'05), Chicago, USA, June 2005.

[Montemerlo 2002] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.* In Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002. AAAI.

[Montemerlo 2003] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit. *FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges.* In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003. IJCAI.

[Montesano 2010] L. Montesano, M. Díaz, S. Bhaskar and J. Minguez. *Towards an Intelligent Wheelchair System for Users With Cerebral Palsy.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 18, no. 2, pages 193–202, April 2010.

[Nicolescu 2001] M. N. Nicolescu and M. J. Mataric. *Learning and Interacting in Human-Robot Domains.* IEEE Transactions on Systems, Man and Cybernetics, Part A. Systems and Humans, vol. 31, no. 5, pages 103–112, September 2001.

[Noreau 2000] L. Noreau and P. Fougeyrollas. *Long-term consequences of spinal cord injury on social participation: the occurrence of handicap situations.* Journal of Disability and Rehabilitation, vol. 22, no. 4, pages 170–180, 2000.

[Nudehi 2003] S.S. Nudehi, R. Mukherjee and M. Glodoussi. *A Haptic Interface design for Minimally Invasive Telesurgical Training and Collaboration in the Presence of Time Delay.* In Proc. IEEE Int. Conf. on Decision and Control, 2003.

[Nunes 2003] U. Nunes, J.A. Fonseca, L. Almeida, R. Araújo and R. Maia. *Using distributed systems in real-time control of autonomous vehicles.* Robotica, vol. 21, no. 3, pages 271–281, June 2003.

[Nunes 2007] Urbano Nunes and L. Bento. *Data fusion and path-following controllers comparison for autonomous vehicles.* Nonlinear Dynamics, vol. 49, pages 445–462, 2007. 10.1007/s11071-006-9108-y.

[of Inequalities 2010] Observatory of Inequalities. *População com deficiência e incapacidade na União Europeia: elevadas taxas de inactividade (in Portuguese).* online: http://observatorio-das-desigualdades.cies.iscte.pt, 2010.

[Ong 2005] K. W. Ong, G. Seet and S. K. Sim. Cutting edge robotics, chapter Sharing and trading in a Human-Robot System, pages 467–496. pro literatur Verlag, 2005.

[Papanikolopoulos 1992] N. P. Papanikolopoulos and P. K. Khosla. *Shared and traded telerobotic visual control.* In Proc. IEEE International Conference on Robotics and Automation (ICRA'92), 1992.

[Paul 2010] R. Paul and P. Newman. *FAB-MAP 3D: Topological mapping with spatial and visual appearance.* In 2010 IEEE International Conference on Robotics and Automation (ICRA), pages 2649 –2656, may 2010.

[Payne 1993] J.W. Payne, J.R. Bettman and E.J. Johnson. The adaptive decision maker. Cambridge University Press, 1993.

[Pearl 1982] Judea Pearl and Jin H. Kim. *Studies in Semi-Admissible Heuristics.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 4, pages 392–399, July 1982.

[Pedersen 2003] L. Pedersen, D. Kortenkamp, D. Wettergreen and I. Nourbakhsh. *A survey of space robotics.* In 7th International Symposium on Artificial Intelligent, Robotics and automation in Space, Nara, Japan, May 2003.

[Pfurtscheller 1998] G. Pfurtscheller, N. Christa, A. Scholgl and K. Lugger. *Separability of EEG Signals Recorded During Right and Left Motor Imagery Using Adaptive Autoregressive Parameters.* IEEE Trans. Rehab. Eng., vol. 6, no. 3, pages 316–324, September 1998.

[Philips 2007] J. Philips, J. del R. Millan, G. Vanacker, E. Lew, F. Galan, P.W. Ferrez, H. Van Brussel and M. Nuttin. *Adaptive Shared Control of a Brain-Actuated Simulated Wheelchair.* In Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on, pages 408 –414, june 2007.

[Pires 2002] Gabriel Pires and Urbano Nunes. *A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic.* Journal of Intelligent and Robotic Systems, vol. 34, pages 301–314, July 2002.

[Pires 2009] G. Pires and U. Nunes. *A Brain Computer Interface methodology based on a visual P300 paradigm.* In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 4193 –4198, oct. 2009.

[Pires 2011a] Gabriel Pires, Urbano Nunes and Miguel Castelo-Branco. *Comparison of a Row-column Speller vs a Novel Lateral Single-character Speller: assessment of BCI for severe motor disabled patients.* Clinical Neurophysiology (In press), 2011. doi:10.1016/j.clinph.2011.10.040.

[Pires 2011b] Gabriel Pires, Urbano Nunes and Miguel Castelo-Branco. *Statistical spatial filtering for a P300-based BCI: Tests in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis.* Journal of Neuroscience Methods, vol. 195, no. 2, pages 270–281, 2011.

[Pires 2012a] G. Pires. *Biosignal Classification for Human Interface with Devices and Surrounding Environments.* PhD thesis, Institute for Systems and Robotics - University of Coimbra, 2012.

[Pires 2012b] G. Pires, U. Nunes and M. Castelo-Branco. *Comparison of a Row-column Speller vs a Novel Lateral Single-character Speller: assessment of BCI for severe motor disabled patients.* Clinical Neurophysiology, vol. 123, no. 6, pages 1168–1181, 2012.

[Poncela 2009] A. Poncela, C. Urdiales, E. J. Pérez and F. Sandoval. *A new efficiency-weighted strategy for continous human/robot cooperation in navigation.* IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 39, no. 3, pages 486–500, May 2009.

[Rahok 2009] S.A. Rahok and O. Koichi. *Odometry correction with localization based on landmarkless magnetic map for navigation system of indoor mobile robot.* In 4th International Conference on Autonomous Robots and Agents, ICARA 2009, pages 572 –577, February 2009.

[Randria 2007] I. Randria, M.M.B. Khelifa, M. Bouchouicha P. and Abellard. *A Comparative Study of Six Basic Approaches for Path Planning Towards an Autonomous Navigation.* In Proc. 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2007), pages 2730 –2735, November 2007.

[Rantanen 2003] E. M. Rantanen and A. Nunes. *Taxonomies of measures in air traffic control.* In 12th International Symposium on Aviation Psychology, 2003.

[Rebsamen 2010] B. Rebsamen, Guan Cuntai, Zhang Haihong, Wang Chuanchu, Teo Cheeleong, M.H. Ang and E. Burdet. *A Brain Controlled Wheelchair to Navigate in Familiar Environments.* IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 18, no. 6, pages 590–598, Dec 2010.

[Reid 1979] D. Reid. *An algorithm for tracking multiple targets.* IEEE Transactions on Automatic Control, vol. 24, no. 6, pages 843 – 854, December 1979.

[rob 2012] *RobChair Project Web Page, online:http://robchair.isr.uc.pt*, 2012.

[Rofer 2000] T. Rofer and A. Lankenau. *Architecture and applications of the Bremen Autonomous Wheelchair.* Information Sciences, vol. 126, no. 1-4, pages 1–20, 2000.

[Roh 2005] S. Roh and H. R. Choi. *Differential-Drive In-Pipe Robot for Moving Inside Urban Gas Pipelines.* IEEE Transactions on Robotics, vol. 21, no. 1, pages 1–17, February 2005.

[Roman 1993] T. H. Roman. *Robotic applications in PSE and G's nuclear and fossil power plants.* IEEE Transactions on Energy Conversion, vol. 8, no. 3, pages 584–592, 1993.

[Rothrock 2003] L. Rothrock and A. Kirlik. *Inferring rule-based strategies in dynamic judgment tasks: toward a noncompensatory formulation of the lens model.* IEEE Transactions on Systems, Man, and Cybernatics - Part A: Systems and Humans, vol. 33, no. 1, pages 58–72, 2003.

[Sagrebin 2008] M. Sagrebin, J. Pauli and J. Herwig. *Behavior based robot localisation using stereo vision.* In Proceedings of the 2nd international conference on Robot vision, RobVis'08, pages 427–439, Berlin, Heidelberg, 2008. Springer-Verlag.

[Schlogl 2007] A. Schlogl, J. Kronegg, J. E. Huggins and S. G. Mason. Evaluation criteria for BCI research, chapter 19, pages 327–342. MIT Press, 2007.

[Se 2002] Stephen Se, David Lowe and Jim Little. *Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks.* International Journal of Robotics Research, vol. 21, pages 735–758, 2002.

[Sheridan 1992] T. B. Sheridan. Telerobotics, automation, and human supervisory control. Cambridge, Mass : The Mit Press, cop, 1992.

[Silva 2005] V. Silva, J.A. Fonseca, U. Nunes and R. Maia. *Communications Requirements for Autonomous Mobile Robots: Analysis and Examples.* In 6th IFAC Int. Symposium on Fieldbus Systems and their Applications (FeT 2005), Mexico, 2005.

[Simon 1996] H.A. Simon. The science of the artificial. MIT Press, 3rd edition édition, 1996.

[Simpson 1998] R. Simpson, S. Levine, D. Bell, L. Jaros, Y. Koren and J. Borenstein. *NavChair: An assistive wheelchair navigation system with automatic adaptation.* In Vibhu Mittal, Holly Yanco, John Aronis and Richard Simpson, editors, Assistive Technology and Artificial Intelligence, volume 1458 of *Lecture Notes in Computer Science*, pages 235–255. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055982.

[Solea 2009] Razvan Solea and Urbano Nunes. *Robotic wheelchair trajectory control considering user comfort.* Informatics in Control, Automation and Robotics LNEE, Springer, vol. 37, pages 113–125, 2009.

[Steiner 2012] Franz Steiner. *25 Spectacular 3D Robots Relationship with Human-The Vision of Future by Franz Steiner.* online in http://webneel.com/webneel/blog/25-3d-robots-relationship-human-vision-future-franz-steiner, 2012.

[Steinfeld 2006] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz and M. Goodrich. *Common metrics for human-robot interaction.* In Proc. of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction (2006), ACM, pages 33–40, 2006.

[T. Fong 2001] C. Thorpe T. Fong and C. Baur. *A Safeguarded Teleoperation Controller.* In Proc. IEEE International Conference on Advanced Robotics (ICAR'01), 2001.

[T. Sawaragi 2003] M. Ishizuka T. Sawaragi Y. Horiguchi. *Modeling ecological expert: comparative analysis of human skill development with fuzzy controller.* Soft Computing, vol. 7, pages 140–147, 2003.

[Taha 2008] T. Taha, J.V. Miró and G. Dissanayake. *POMDP-based long-term user intention prediction for wheelchair navigation.* In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'08), Pasadena, CA, USA, May 2008.

[Tahboub 1999] K.A. Tahboub and H.H. Asada. *A compliant semi-autonomous reactive control architecture applied to robotic holonomic wheelchairs.* In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 665 –670, 1999.

[Tahboub 2001] Karim Tahboub. *Natural and Manmade Shared-Control Systems: An Overview.* In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'01), pages 2655–2660, Seoul, Korea, May 2001.

[Takeda 1997] H. Takeda, N. Kobayashi, Y. Matsubara and T. Nishida. *Towards ubiquitous human-robot interaction.* In In Working Notes for IJCAI-97 Workshop on Intelligent Multimodal Systems, pages 1–8, 1997.

[Tanaka 2005] K. Tanaka, K. Matsunaga and H.O. Wang. *Electroencephalogram-Based Control of an Electric Wheelchair.* IEEE Transactions on Robotics, vol. 21, no. 4, pages 762–766, Aug 2005.

[Tani 2011] A. Tani, G. Endo, E. F. Fukushima, S. Hirose, M. Iribe and T. Takubo. *Study on a Practical Robotic Follower to Support Home Oxygen Therapy Patients -Development and Control of a Mobile Platform.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'11), 2011.

[Thrun 1999] S. Thrun, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz. *MINERVA: A Second-GenerationMuseum Tour-Guide Robot.* In IEEE International Conference on Robotics and Automation (ICRA'99), Detroit, USA, May 1999.

[Thrun 2001] S. Thrun, D. Fox, W. Burgard and F. Dellaert. *Robust Monte Carlo localization for mobile robots.* Artificial Intelligence, vol. 128, no. 1-2, pages 99–141, 2001.

[Thrun 2002] S. Thrun. *Robotic Mapping: A Survey.* In G. Lakemeyer and B. Nebel, editors, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann Series in Artificial Intelligence, 2002.

[Thrun 2006] S. Thrun, W. Burgard and D. Fox. Probablistic robotics. The MIT Press, 2006.

[Ulrich 2000] I. Ulrich and J. Borenstein. *VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots.* In Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00), Leuven, Belgium, 2000.

[van der Heijden 2004] F van der Heijden, R P W Duin, D de Ridder and D M J Tax. Classification, parameter estimation and state estimation, an engineering approach using matlab. John Wiley and Sons, 2004.

[Vanacker 2007] G. Vanacker, J. del R. Millán, E. Lew, P.W. Ferrez, F. Galán, J. Philips, H. Van Brussel and M. Nuttin. *Context-Based Filtering for Assisted Brain-Actuated Wheelchair Driving.* Comput. Intell. Neurosci, vol. 2007, 2007.

[Vanhooydonck 2010] Dirk Vanhooydonck, Eric Demeester, Alexander HÃ¼ntemann, Johan Philips, Gerolf Vanacker, Hendrik Van Brussel and Marnix Nuttin. *Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model.* Robotics and Autonomous Systems, vol. 58, no. 8, pages 963 – 977, 2010.

[Wade 2011] E. Wade, A. Parnadi and M. J. Matarić. *Using socially assistive robotics to augment motor task performance in individuals post-stroke.* In Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS'11), 2011.

[Wasson 2001] G. S. Wasson and J. P. Gunderson. *Variable Autonomy in a Shared Control Pedestrian Mobility Aid for the Elderly.* In IJCAI'01 Workshop on Autonomy, Delegation and Control, Seattle, Washington, USA, August 2001.

[Web 2012] *Webester's Online Dictionary.* http://www.websters-online-dictionary.org/, July 2012.

[WRD 2011] *World Report on Disability.* Technical report, World Health Organization (WHO), 2011.

[Yanco 2004] Holly A. Yanco. *Classifying Human-Robot Interaction: An Updated Taxonomy.* In Proc IEEE SMC, pages 2841–2846, 2004.

[Yin 2006] J. Yin and L. Rothrock. *A rule-based lens model.* International Journal of Industrial Ergonomics, vol. 36, pages 499–509, 2006.

[Yoerger 1987] D. Yoerger and J. Slotine. *Human Robot Synergism and Virtual Telerobotic Control.* In Proc. IEEE International Conference on Robotics and Automation (ICRA'87), 1987.

[Zhai 1992] S. Zhai and P. Milgram. *Human Robot Synergism and Virtual Telerobotic Control.* In Proc. 25th Annual Conference of the Human Factors Association of Canada, 1992.

[Zhai 1999] S. Zhai, C. Morimoto and S. Ihde. *Manual And Gaze Input Cascaded (MAGIC) Pointing.* In In CHI99, pages 246–253. ACM Press, 1999.

[Zhang 1992] Z. Zhang and O. D. Faugeras. *Estimation of displacements from two 3D frames obtained from stereo.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 12, pages 1141 –1156, December 1992.