Assisted Navigation for a Brain-actuated Intelligent Wheelchair

Ana C. Lopes, Gabriel Pires, and Urbano Nunes

Institute for Systems and Robotics, University of Coimbra, Polo II, 3030-290 Portugal
Email Address: anacris, gpires, urbano@isr.uc.pt

Abstract

This paper presents an Assistive Navigation System (ANS) for a Robotic Wheelchair (RW) relying on a Brain-Computer Interface (BCI), as the Human-Machine Interface (HMI). A two-layer collaborative control approach is proposed to steer the RW, taking into account both, user and machine commands. The first layer, a virtual-constraint layer, is responsible for enabling/disabling the user commands, based on context. More specifically, user commands are enabled for a set of situations requiring user decision, namely, bifurcations, multiple-directions caused by new obstacles in the environment, and deadlocks. The second layer is a user-intent matching responsible for determining the suitable steering command that better fits the user selection, taking into account the user competence to steer the wheelchair, and situation awareness of potential directions at a given location. A P300-based BCI allows the selection of commands to steer the RW. Experimental results using RobChair [1], [2] are presented, showing the effectiveness of the proposed methodologies. The ANS was validated with ten able-bodied participants, and one participant with cerebral palsy, in two different scenarios: a structured known environment, and a structured unknown environment with moving objects. The overall result was 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.

Keywords: Wheelchair Navigation, Brain-Computer Interface, Human-Machine Interfaces, Discrete HMIs, Shared Control, Traded Control, Collaborative Control.

1. Introduction

The main goal of this research work is to contribute with new methods and algorithms for collaborative control in assisted navigation, taking into account user competence in steering a RW with BCI, and context. The proposed assisted navigation is based on a semi-autonomous controller that requires at least two agents, a Human/User Agent (HA/UA) and a Machine Agent (MA), that collaborate to control the robot. Safety issues are a key research topic in assisted navigation, being required to deal with unknown obstacles, and to perceive and interact with humans. When dealing with semi-autonomous systems, care must be taken in the choice and design of the most appropriate system interface. The HMI must be well suited to user needs and capabilities, and the semi-autonomous system must be able to identify and respond to user's commands, in the most appropriate and safe manner.

In this paper, we propose an ANS, which uses a non-invasive BCI. This ANS was designed for people with severe motor disorders such as Amyotrophic Lateral Sclerosis (AMS), and Cerebral Palsy (CP). 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 [3]. 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. Since at least one specific motor skill is required to operate most of HMIs, BCI opens a new communication channel for these users. Bearing these facts in mind, we propose a BCI based on electroencephalography (EEG) that translate brain patterns into commands to steer a RW [4, 5] (videos and project details can be found in [6]). In previous works we have already showed that individuals with severe motor disabilities can use BCI to spell characters with acceptable communication rates [7], [8]. 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 in the environment, or when ambiguous situations occur.

1.1. Related work

In recent years different assisted navigation architectures were developed for intelligent wheelchair platforms, such as RobChair developed at ISR-UC [1], [4]. Other ANSs for RWs are presented in [9, 10, 11, 12, 13, 14]. Most of these architectures are based on semi-autonomous control approaches that consist on sharing, trading, or a combination of both, the control of the tasks between human and machine agents. Shared control incorporates the strength of the human and robot by letting them control different aspects of the system simultaneously in situations that require teamwork. In a shared controller each task has a simultaneous intervention of the robot and the human operator, meaning that the tasks are not mutually exclusive, concerning human and machine command signals [15]. Traded control is a mutually exclusive control approach, meaning that either the human or the robot turns over control to the other. For a traded controller each task is assigned either to the robot or to the human according to their ability to execute it [15]. Semi-autonomous control can also be adaptive when the system is capable of changing the nature and degree of control shared or traded between human and machine during task execution [9]. In [16] 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, adjusts the control signals to achieve the desired goal. If the power assigned to agents is variable, it would be expected that this variability would also depend on user ability to steer the RW. There are some shared control approaches that evaluate user and robot steering efficiencies at each time instant, and combine their commands into a single order [17], [18]. However, the proposed evaluation methodologies are applied with interfaces that are able to provide a continuous signal over time (e.g. joystick), and mainly suited for people with some motor capabilities. In [11] a shared control approach is proposed that continuously estimates the user's intention and determines whether the user needs assistance to achieve his/her intention. An implicit user model is incorporated in the framework, in order to make the execution of both tasks adaptable to a specific user. This technique is also applied with a conventional joystick interface.

Brain-actuated wheelchairs have been researched by several research groups. See a summary of a set of representative works in Table 1. Systems are mainly based on two types of brain control: 1) modulation of sensory-motor rhythms by performing mental tasks (e.g. motor imagery) [19]; and 2) detection of P300 event related potentials through the design of oddball paradigms [20]. While in motor imagery the number of commands is limited to 3 or 4, P300-BCIs can provide a significant number of commands, but depend on external visual stimuli (see [21] 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 instead self-paced by the user. In P300-based systems, the

Table 1: Summary of relevant research works in the field of brain-actuated wheelchairs. In all works the participants were able-bodied.

BCI approach	Asynch. control	Description	N^{o}. of Participants
Left and right imagery [22]	No	Control of real wheelchair (commands: left, right)	6
Motor imagery [23, 24]	Yes	Control of real wheelchair (commands: forward, left, right)	5
Motor imagery (left hand and rest) and words association [25]	Yes	Control of a simulated wheelchair in 3D environment (commands: left, right, forward)	2
Motor imagery (left and right hand) [26]	Yes	Control of 2D simulated wheelchair (commands: go, stop, right, left)	5
P300 visual paradigm built in a virtual 3D reconstruction of environment [27]	No	Control of real wheelchair and simulated wheelchair in virtual environment (selection of local surrounding points)	5
P300 visual paradigm combined with motor imagery [28]	Yes	Control of real wheelchair (selection of high-level destination goals with P300 (e.g. kitchen) and stop detection with motor imagery	5

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 experimental results achieved with brain-actuated wheelchairs until now are very positive, but show that its effective application in real-world scenarios is not yet possible. The low transfer rate and low robustness of BCIs, and the demanding requirements of human centered robots still pose many research challenges, some of them being pursued in this work.

1.2. Overview and contributions

This paper proposes a navigation system based on collaborative control for a brain-actuated intelligent wheelchair. A P300-based BCI that allows the selection of brain commands to steer a RW is presented. To relieve user effort, low-level commands are only issued when there are dynamic changes of the environment or when decision making is needed (e.g. bifurcation). A two-layer collaborative control approach is proposed in order to obtain a safe and effective navigation of the RW, receiving user commands that are issued sparsely. The two-layer collaborative controller includes a virtual-constraint layer and an intent-matching layer. The former is responsible for enabling/disabling user commands, as a function of certain criteria, and the latter determines the suitable maneuvers, taking into account the user competence to steer a RW. The ANS with BCI was validated with eleven participants (10 able-bodied and one motor disabled). Metrics to evaluate the performance of the BCI online system, user, and navigation system are presented and discussed.

1.3. Paper organization

This paper is organized as follows: the introduction is presented in Section 1, including a literature review in Subsection 1.1, and main contributions in Subsection 1.2. Section 2 describes the navigation system architecture, including a description of the system mapping (Subsection 2.1), local motion planning (Subsection 2.2), and localization (Subsection 2.3). The system HMI, particularly, P300-based BCI, and the developed paradigm, are presented in Section 3. The proposed collaborative controller is presented in Section 4. Experimental results with able-bodied users and one user with cerebral palsy are presented in Section 5, before drawing some conclusions in Section 6.

2. ANS - Assistive Navigation System

The ANS has been tested in player/stage [29] simulation environment, and in RobChair [4]. It is structured in five main levels, as shown in Fig. 1:

- HMI: provides user intent (sparse steering commands) to the system through a P300-based BCI. Inputs: 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 [30], [31] to the selected goal. This method is implemented online. In

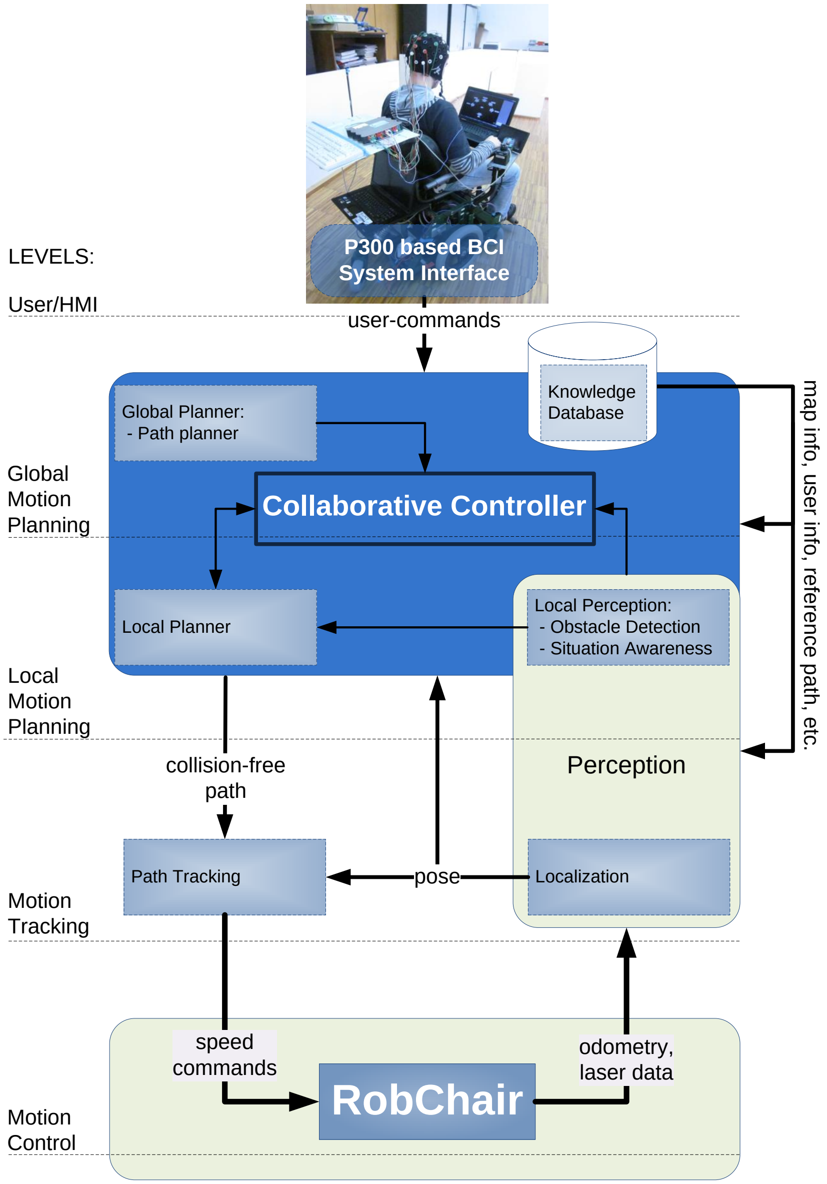


Figure 1: ANS - Assistive navigation system architecture.

the second approach, goals and waypoints are predefined, and multiple reference paths between waypoints are determined offline, using the method proposed in [32]. The second approach was used in the experiments presented in this paper. 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.

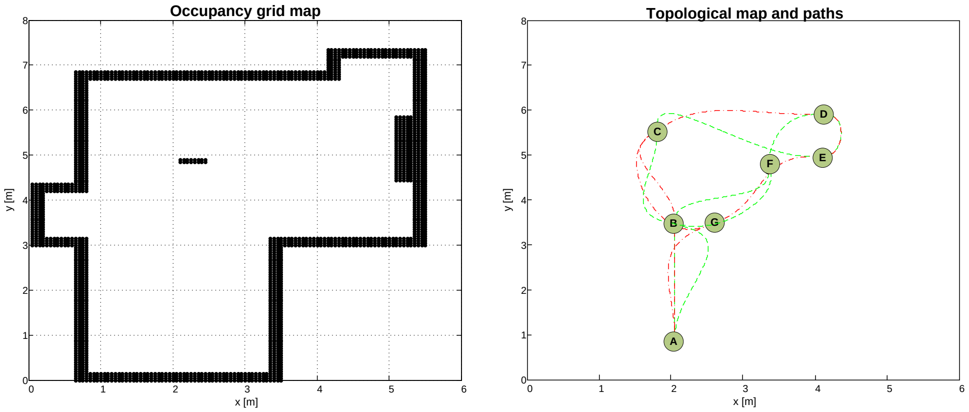
The ANS also integrates a knowledge database that stores, namely, information regarding the working environment, situation-based restrictions, and driving rules. It also stores the global paths provided by the global planner. When required, this information is provided to several levels of the ANS, as depicted in Fig. 1.

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

2.1. 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,



a) b)

Figure 2: A priori map used as the base scenario for the experimental tests: a) grid map; b) topological map with possible paths.

such as the one depicted in Fig. 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. 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.

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 $(width \times length)$ of $(5 \times 5)\ cm^2$. Each grid node has an associated occupancy value related to the presence of obstacles. RobChair is also provided with a topological map containing all possible goals and waypoints (bifurcations, and final destinations) of the environment, as shown in the example of Fig. 2 b). The topological map was constructed offline.

2.2. *Local motion planning*

The local planner is based on the VFH+ algorithm [33] with a few modifications as described below. This method is illustrated in Fig. 3. 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 the 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 grid map, or if it is

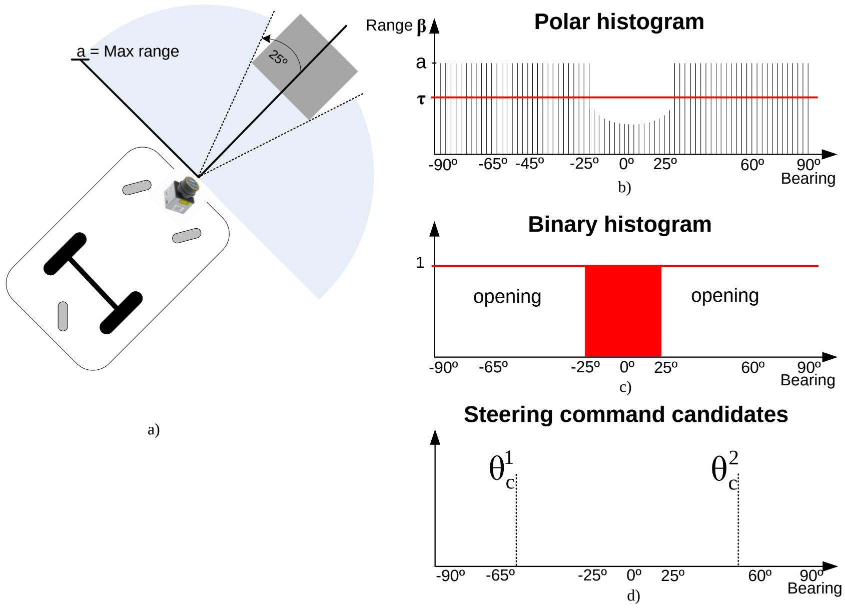

a = Max range
25º
a)
Range β
Polar histogram
a
τ
-90º -65º -45º -25º 0º 25º 60º 90º
Bearing
b)
Binary histogram
1
opening
opening
-90º -65º -25º 0º 25º 60º 90º
Bearing
c)
Steering command candidates
θc1
θc2
-90º -65º -25º 0º 25º 60º 90º
Bearing
d)

Figure 3: Obstacle detection using an adaptation of the VFH+ algorithm. a) The RW detects a new obstacle in the environment; b) Construction of the polar field histogram; c) Construction of the binary field histogram; d) Steering command candidates computed by the local planner.

new to the environment. In case of a new obstacle is detected closer than a certain threshold, not allowing the RW to follow the reference path, the local planner enters in operation. A polar field histogram is directly determined from the laser scan (containing information of all obstacles in its range), assigning to each i^{th} scan sector a value β_i that is equal to the range of that sector. To construct the binary polar histogram, a threshold τ is used in such a way that a value of zero is assigned to sectors with $\beta_i \geq \tau$ (the sector is considered free of obstacles), and a value of one otherwise, as shown in Fig. 3.

Details on how the collaborative controller relates with the other system modules is shown in Fig. 4. The local planner is structured in three modules: steering, local path-planner, and blending (see Fig. 4). The steering module receives the admissible openings from the obstacle detection module, and then calculates the steering command candidates derived from the previous admissible openings, as proposed in [33]. For each candidate direction θ_c^i a cost function is calculated as,

$$\gamma(\theta_c^i) = \mu_1 \Delta(\theta_c^i, \theta_t) + \mu_2 \Delta(\theta_c^i, \theta) \tag{1}$$

where θ_t is the target direction towards the next subgoal in the reference path, and

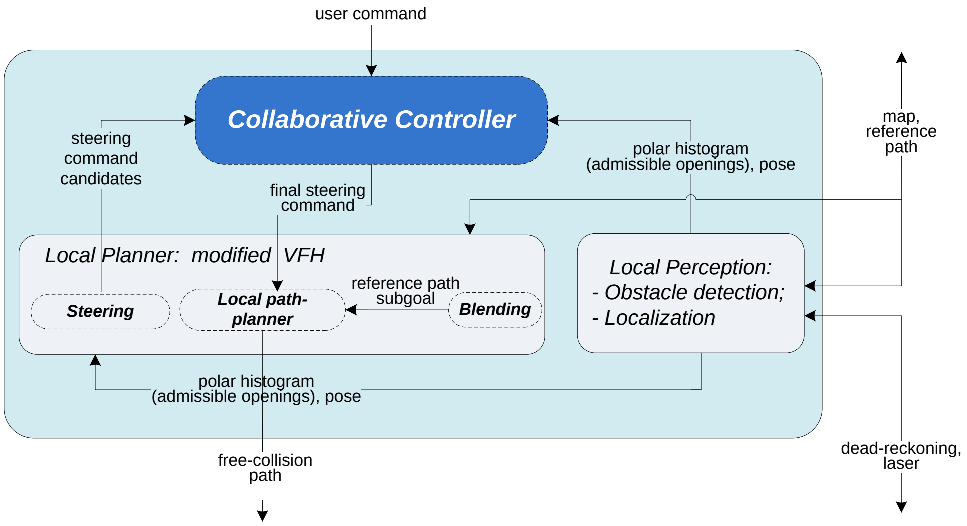


Figure 4: Data flow exchange between the local perception module, local planner, and collaborative controller

θ denotes the cartesian current orientation of the RW. The generic term $\Delta(c_1, c_2)$ gives the absolute angle difference between two generic sectors c_1 and c_2.

The local path-planner determines a free-collision path as proposed in [4], based on the final steering command, φ, provided by the collaborative controller. In [4] we proposed a 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 a certain distance ahead of the RW as illustrated in Fig. 5 (positions $P1$ and $P2$).

As soon as the RW stops detecting new obstacles in the environment, local and reference paths are blended. This is achieved by providing a subgoal of the reference path (provided by the global planner) as the target direction, θ_t, in cost function (1). Therefore, a final local path is planned as a linear interpolation between the current pose of the robot and the subgoal of the reference path (P3 in the example of Fig. 5), which is sufficiently ahead to comply with the geometric constraints of the RW during the contour of the obstacles. Figure 5 shows an example how the local planner works to avoid an obstacle placed in the robot path.

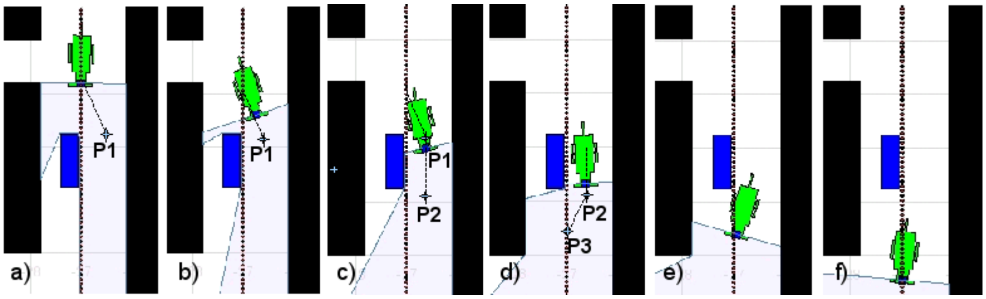


Figure 5: Obstacle avoidance method: a) robot detects a new obstacle and determines a free path to a target pose $P1$ (center point of the free space between the obstacle and the infrastructure in the direction provided by the collaborative controller); b) robot follows the path; c) robot reaches $P1$, and determines a path to pose $P2$; d) robot reaches $P2$ and stops detecting the new obstacle; a local path to blend with reference path $P3$, is determined; e) robot follows the path to reach $P3$; f) robot follows the reference path. The dotted line represents the reference path provided by the global planner [4].

2.3. Localization

Localization is performed using odometry for rough positioning, and laser data for polar scan matching. A grid Markov localization system [34, 35] was designed to fuse odometry with laser polar scan matching. 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 (5×5) cm^2). The localization approach is presented in Algorithm 1. 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 3 in Subsection 2.3.3) is used to weight each state $\bar{x}_{ijk,t}$ with the likelihood of observing the current scan measurement r_t, taking the map 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 center defined by x_{ij}, and 21 orientations defined by the set $\{-10^o, -9^o ... 0^o ... +9^o, +10^o\}$. Since grid Markov localization approach is, in general, computationally expensive, three techniques to reduce the computational effort, were applied:

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

Algorithm 1 MarkovLocalization(x_{t-1}, u_t, 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
for all $i, j$ do
    //for the 21 orientations
    for all $k$ do
        $p_{ijk,t} \leftarrow \eta PolarScanMatching(\bar{x}_{ijk,t}, r_t, m)$
    end for
end for
return($x_t$ $with$ $max(p_{ijk,t})$)
```

- 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^{o}). The orientation search resolution was 0.017 *rad*;
- The measurement update is applied whenever the RW travels approximately 30 *cm*, instead of being applied every time a new odometry value is calculated.

2.3.1. 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 in 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 [36].

2.3.2. 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 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 2 shows the $VirtualScan(\bar{x}_{ijk,t}, m)$ algorithm used to determine the virtual scan from pose estimation $\bar{x}_{ijk,t}$. 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 between $MAXRANGE$ and $\bar{x}_{ijk,t}$, in the direction of $\theta_{bg}(l) + \bar{\theta}_{ijk,t}$ ($l = \{1, ..., N\}$ represents the number of the scan sector), are evaluated in terms of occupation. If a cell with position (X_r, Y_r) is occupied, $m(X_r, Y_r) = 1$, then the range of the reference scan in the direction $\theta_{bg}(l) + \bar{\theta}_{ijk,t}$ has been found. If we want to make sure that the virtual scan truthfully represents the measured scan (denoted as the current scan), we must consider the effects of small measurement noise, errors due to unexpected objects, errors due to failures to detect objects, and random unexplained noise. In [35], 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 noise in Algorithm 2. Considering $\hat{r}_t^*(n)$ as the range measurement 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 [37] for white surfaces.

Algorithm 2 $VirtualScan(\bar{x}_{ijk,t}, m)$

```
for all $n = 1 : N$ do
    for all $l = \Delta : -1 : 1$ 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
            $\hat{r}_t^*(n) \leftarrow r_{nl}$
            $\hat{r}_t(n) \leftarrow (0.98\hat{r}_t^*(n)) + \sigma * Randn(1)$
            break;
        end if
    end for
end for
return($\hat{r}_t$)
```

2.3.3. *Polar Scan Matching*

The laser scan matching method compares the current scan r_t, given by a laser scanner, with a virtual scan $\hat{r}_t$. The latter is determined based on the grid map m, such that the more similar r_t and $\hat{r}_t$ are, the larger $p(r_t|x_{ijk,t},m)$ 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 substituting estimates of the covariances and variances based on a sample, as follows:

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

where N is the number of scan points, and $\bar{r}$ is the average range value, as follows

$$\bar{r} = \frac{1}{2N}\sum_{i=1}^{N}(\hat{r}_t(i)+r_t(i)) \tag{3}$$

Laser scan matching considers

$$P(r_t|\bar{x}_{ijk,t},m) = max(r_{\hat{r}_t,r_t},0) \tag{4}$$

as the probability of the current scan conditioned by the map m and position $\bar{x}_{ijk,t}$. Algorithm 3 presents the polar scan matching algorithm, where r_t' and r_t'' represent the current scan after applying a distance filter and dynamic object filter respectively. N'' corresponds to the number of points in r_t''.

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 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 uncertainty. On 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 proposed BCI 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,

Algorithm 3 PolarScanMatching($\bar{x}_{ijk,t}, r_t, m$)

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

that occurs typically around 300 ms after the relevant event occurs [20] (see Fig. 6a)). The BCI has been designed as a visual oddball paradigm, where the symbols flash randomly (see Fig. 6b)). 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.

3.1. P300-based BCI paradigm

The paradigm comprises seven steering commands, θ_{UA}, encoded by the following symbols: FORWARD, RIGHT45, RIGHT90, BACKWARD, LEFT45, LEFT90 and STOP (Fig. 6b)). 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. 6c)). 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

$$TT = N_{rep} \times N_s \times SOA + 1 \qquad (5)$$

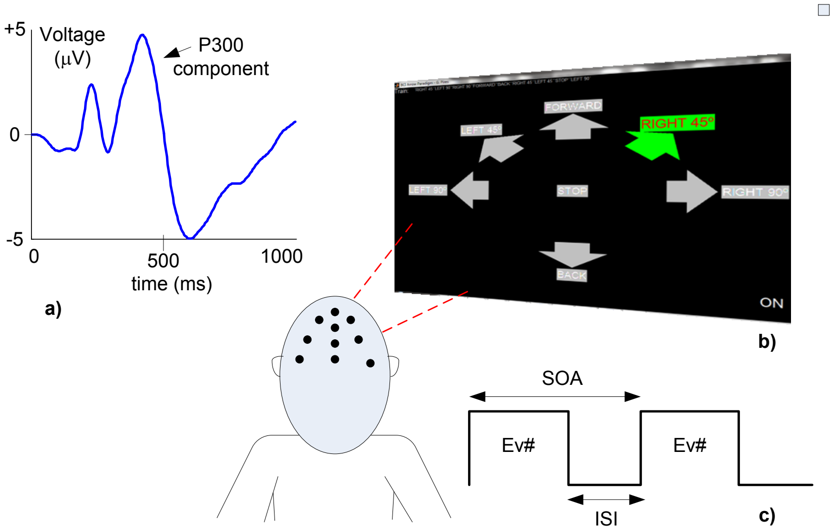


Figure 6: Overview of the BCI system: a) P300 component of the ERP: positive deflection occurring approximately at 300 ms after the onset of the relevant stimulus; b) Visual arrow-paradigm encoding 7 symbols for steering control; c) Temporal diagram of the events ($Ev\# \in \{1 \cdots 7\}$ represents the code of the event).

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, $TT = 7 \times 0.175 \times 3 + 1 = 4.675\ s$). 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 settled to select the number of repetitions. Compared with our last arrow paradigm approach [4], we reduced the number of symbols from 11 to 7 in order to reduce the TT.

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 (see Fig. 9). 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 that we have implemented and described in [7]. This classification methodology was already thoroughly tested and validated with success in experiments made by able-bodied and motor disabled participants in our previous study [8]. 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 most likely symbol mentally selected by the user (see details in [7]). 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.

4. Two-layer collaborative controller

The collaborative controller receives commands from two agents: a human agent and a machine agent. The user issues BCI-actuated commands θ_{UA}. The proposed controller includes a virtual-constraint layer and an intent-matching layer. The former is responsible to enable/disable user commands, as a function of certain criteria, and the latter determines the suitable maneuvers, taking into account his/her steering competence, as outlined in Fig. 7. The Assisted Navigation Training Framework (ANTF) approach [38] is used to evaluate the user steering competence. With this approach users are sorted in one of three possible learning stages: beginner, average, or advanced. In [38] we proposed the Rule-Based Lens (RBL) model for the ANTF, which characterizes the achievement rate *ra*, of each user in steering the RW. The achievement rate consists in the correspondence rate between human judgments (user steering command to solve a certain navigation task) and the actual value of the environmental criterion (expected steering command to solve that particular navigation task).

4.1. Virtual-constraint layer

The Virtual-Constraint Layer (VCL) is responsible to enable/disable the commands provided by the user subject to context. This layer is required because the BCI system is continuously providing navigation commands independently of user intention, while in the disabled state, the RW disregards commands from the user. The VCL enables user commands according to the following perceived situations:

- $S1$: Multiple possible directions due to a bifurcation;

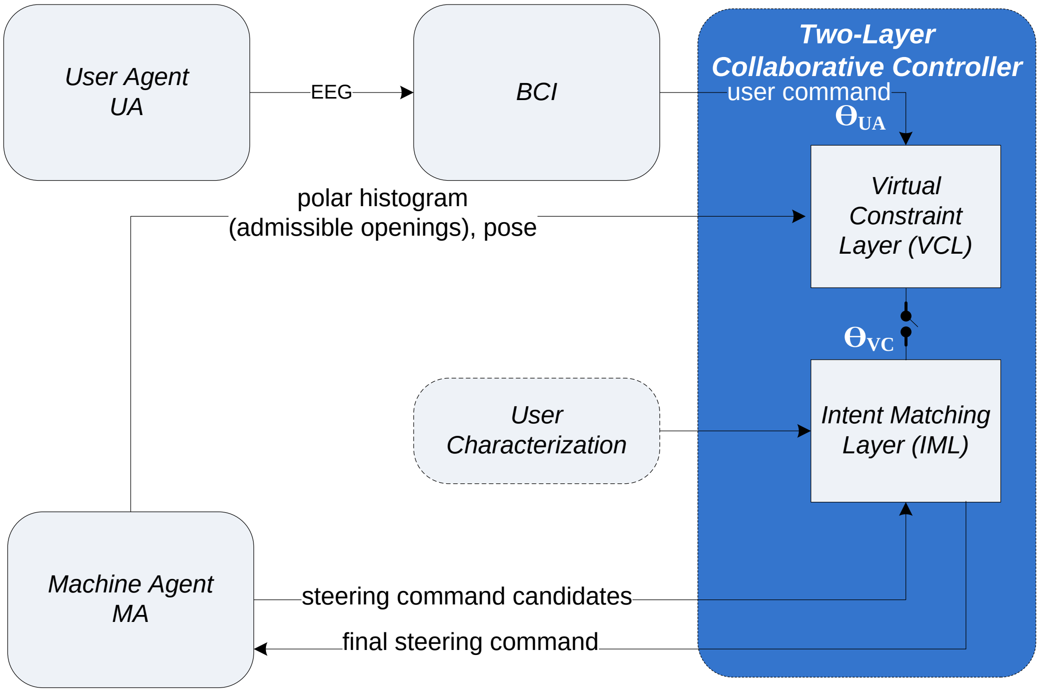


Figure 7: Two-layer collaborative control architecture.

- $S2$: Multiple possible directions to avoid an obstacle;
- $S3$: Solving a deadlock 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 situations. If new obstacles appear close to the RW, the obstacle detection module provides the admissible openings to the VCL. In case there are multiple admissible openings, a situation S2 is perceived, and, on the other hand, if there are no admissible openings the VCL perceives the situation as a deadlock, corresponding to situations S3, or S4. Situation S1 is determined with the aid of the localization system, which provides the VCL with the current localization. When one of those situations occur, the user is requested to select a desired steering command, through the visual arrow paradigm shown in Fig. 6. Additionally, the VCL also takes into account constraints related to the user steering competence, as follows:

$$\begin{aligned}
&Basic\,User:\\
&\theta_{VC}=\begin{cases}\theta_{UA} & if(S1\ \&\ \theta_{UA}\in\{F,\ L90,\ R90\})\\ \theta_{UA} & if(S2\ \&\ \theta_{UA}\in\{F,\ L,\ R\})\\ \theta_{UA} & if(S4\ \&\ \theta_{UA}\in\{L90,\ R90\})\\ none & otherwise\end{cases}\\
&Average\,User:\\
&\theta_{VC}=\begin{cases}\theta_{UA} & if(S1\ \&\ \theta_{UA}\in\{F,\ L90,\ R90\})\\ \theta_{UA} & if(S2\ \&\ \theta_{UA}\in\{F,\ L,\ R\})\\ \theta_{UA} & if(S3\mid S4)\\ none & otherwise\end{cases}\\
&Advanced\,User:\\
&\theta_{VC}=\begin{cases}\theta_{UA} & if(S1\ \&\ \theta_{UA}\in\{F,\ L90,\ R90\})\\ \theta_{UA} & if(S2\mid S3\mid S4)\\ none & otherwise\end{cases}
\end{aligned} \tag{6}$$

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

4.2. Intent-matching layer

This layer determines the final steering command to the RW, based on user-intent constrained by the VCL, and taking into account a set of candidate directions proposed by the MA. The errors $e^i_{\theta_{UA}}$, $i=1,...,n$ between the user command and the directions provided by the MA are calculated as follows:

$$e^i_{\theta_{UA}}=\theta_{VC}-\theta^i_{MA},\ i=1,...,n \tag{7}$$

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 (1). A normalization factor is applied so that the sum of all machine weights is equal to one. For all candidate directions a cost function $g(\theta)$ is defined as follows:

$$g(\theta)=\eta_{MA}\cdot e^i_{\theta_{MA}}+\eta_{UA}\cdot ra\cdot e^i_{\theta_{UA}} \tag{8}$$

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

$$e^i_{\theta_{MA}}=\theta^i_{MA}-\theta_{MA}\ i=1,...,n \tag{9}$$

The weight η_{UA} is defined according to the user steering competence, and *ra* denotes his/her achievement rate, which varies according to $ra \in [0...1]$. The final direction θ is the one that minimizes the cost function $g(\theta)$. For a deadlock perceived situation (S3 and S4), the MA is not able to determine any free direction, and, in these cases, the user should command the RW backwards or perform pure rotations left or right (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 reduced or expanded according to the characteristics of each bifurcation. In the experimental tests presented below, all bifurcations could be solved with LEFT90 or RIGHT90 commands, and for that reason these were the only admissible commands to solve the S1 situation. The RW remained stopped for other commands.

5. Experimental results

In this section 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 [6].

5.1. Characterization of participants

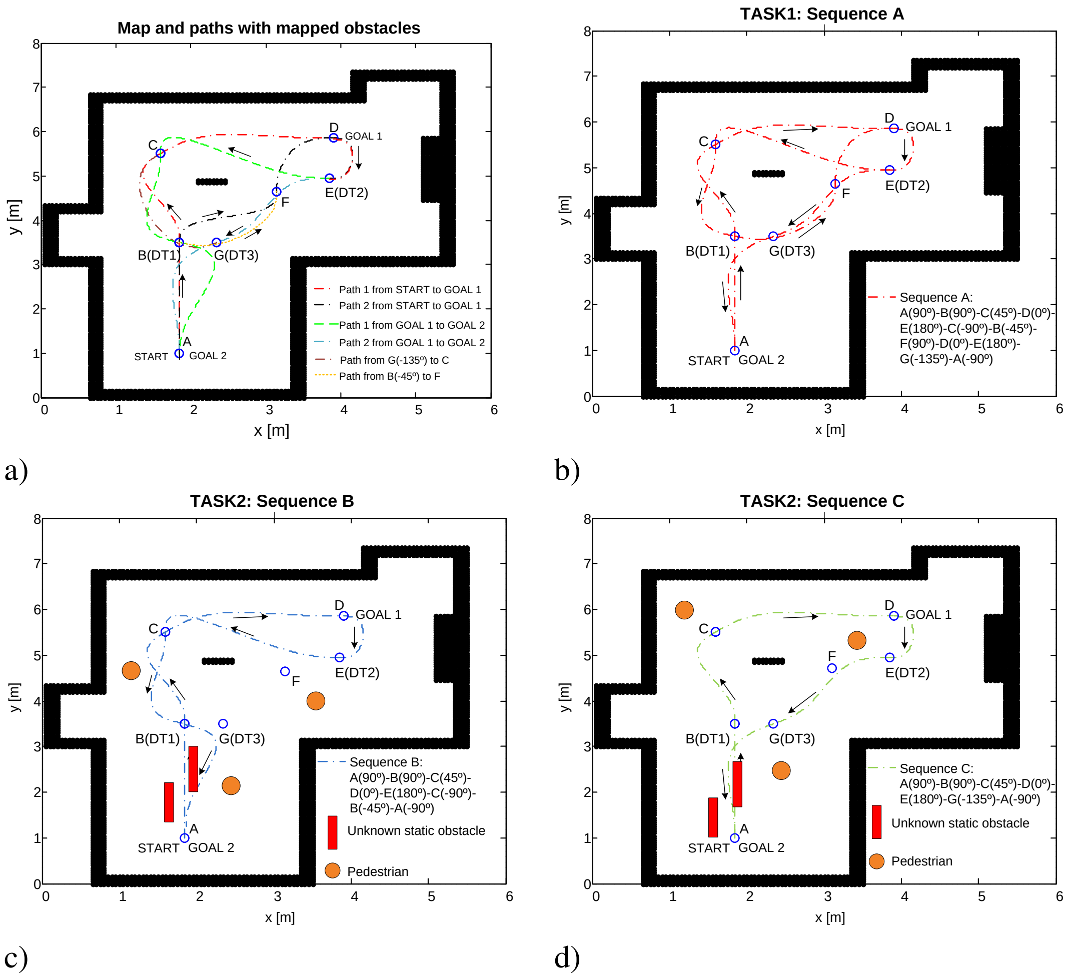
Participants without disabilities included: aged between 25 and 40 years old; approximate number of participants from both genders; higher education; all right-handed; without any relevant history of psychiatric or neurological disorders. Ten able-bodied participants, six male and four female, were selected, 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 2 shows the most relevant data concerning participant characterization. All participants used the ANS for the first time.

Table 2: 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

5.2. 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. 8 b)); the second task, denoted by TASK2, was carried out in



a)

b)

c)

d)

Figure 8: 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;

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. 8 b), c)). Figure 9 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 3. The Decision Target (DT) points were B, E and G, where the user

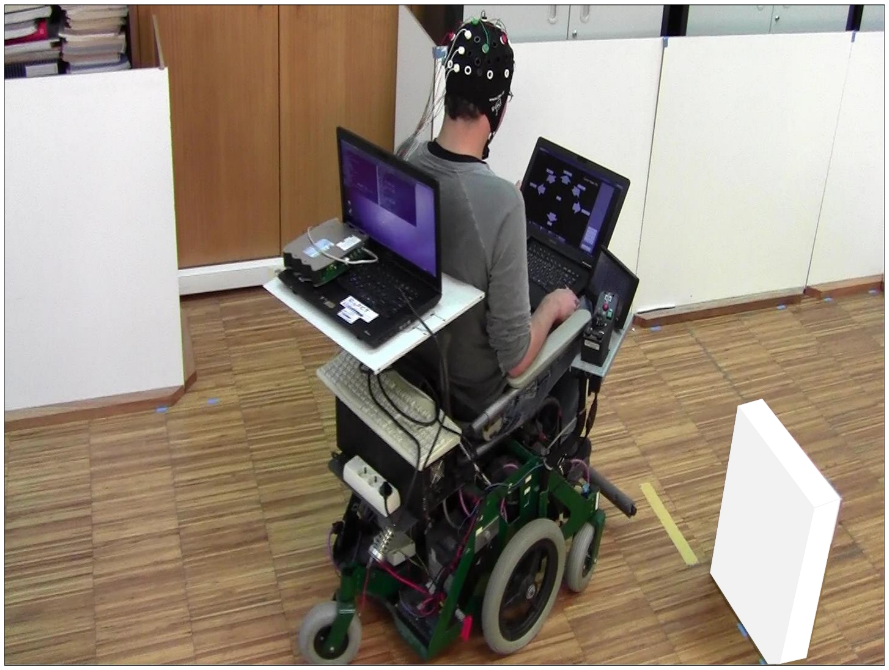


Figure 9: Robchair in test scenario.

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 3) 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 3 for TASK2.

For all experiments presented in this paper, η_{UA} and *ra* parameters were tuned

Table 3: 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	$\uparrow$	A	$\uparrow$	A	$\uparrow$
B (DT)	$\uparrow$	B (DT)	$\uparrow$	B (DT)	$\uparrow$
C	$\nearrow$	C	$\nearrow$	C	$\nearrow$
D	$\rightarrow$	D	$\rightarrow$	D	$\rightarrow$
E (DT)	$\leftarrow$	E (DT)	$\leftarrow$	E (DT)	$\leftarrow$
C	$\downarrow$	C	$\downarrow$	G (DT)	$\swarrow$
B (DT)	$\searrow$	B (DT)	$\searrow$	A	$\downarrow$
F	$\uparrow$	A	$\downarrow$		
D	$\rightarrow$				
E (DT)	$\leftarrow$				
G (DT)	$\swarrow$				
A	$\downarrow$				

manually, as proposed in [4], 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. The cost parameters in (1) were as follows: $\mu_1 = 2$ and $\mu_2 = 1$.

5.3. Assessment of overall performance

The metrics proposed in [27], [39] 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;
- 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);

Table 4: 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 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 4 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 of 77% was registered for TASK1. To calculate the BCI accuracy in TASK1, a

Table 5: 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

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.

5.4. Assessment of online BCI on road

Results of online BCI performance for TASK1 and TASK2, with known and unknown obstacles, are presented in Tables 5 and 6, respectively. According to Table 5, 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 5. 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

Table 6: 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 Subgoals
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

of an admissible 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 6 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 6. 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.

Table 7: Metrics to evaluate the navigation system for task 1

	Task 1			
	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

5.5. Assessment of navigation performance

To evaluate the navigation system performance we used the metrics proposed in [27, 40, 39], namely:

- 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 and 8, 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 [27]. In TASK1 the average speed was 0.19 m/s, but in TASK2, this value decreased for 0.11 m/s. This result indicates

Table 8: Metrics to evaluate the navigation system for task 2

	Task 2			
	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

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.

5.6. 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 [38] to evaluate users at the end of the experiments.

5.6.1. Environment and human models

The ANTF with BCI uses two main types of cues for decision-making, as presented in Table 9: Type I ($X1$ to $X5$) is related to bifurcations, and Type II ($X6$ to

Table 9: 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

$X8$) 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 Genetic Based Policy Capture (GBPC) method described in [41, 42]. The models presented in Table 10 are simplified, because they only encode possible situations that may occur 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 10, there are a few considerations concerned with how the assisted navigation should react to Type II cues (see Table 9 and Table 10). 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

Table 10: 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	$Y4$	$Y1 \cup Y2 \cup Y3 \cup Y4$

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 10, $\hat{Y}s$ column represents the model for a beginner user, while $\hat{Y}e$ 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.

5.6.2. Assessment of user performance

Table 11 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 [42], environmental predictability, *Re*, human

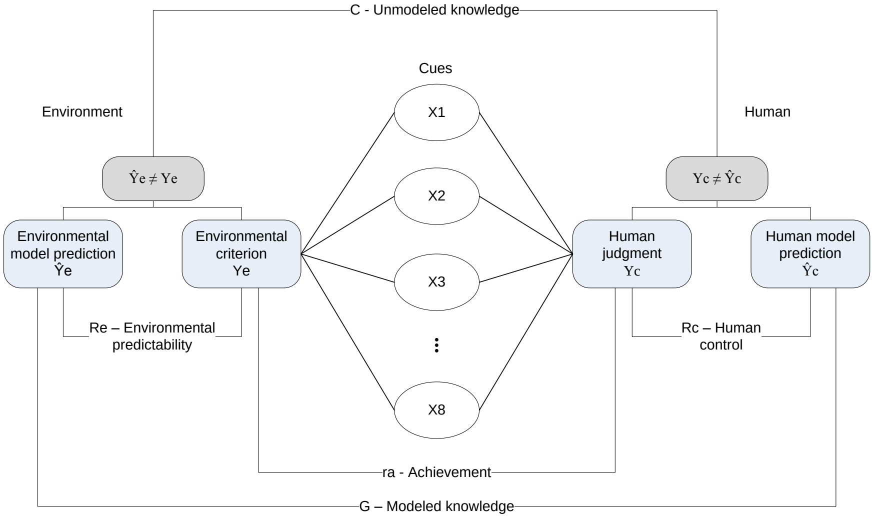


Figure 10: Rule-based lens model parameters based on [42]: environmental predictability, *Re*, human control, *Rs*, achievement, *ra*, for achievement, modeled knowledge, *G*, and unmodeled knowledge, *C*.

control (conformance with human model, see Fig. 10), *Rs*, achievement, *ra*, modeled knowledge, *G*, and unmodeled knowledge, *C*, were computed according to the following expressions (see Fig. 10:

$$Re = \frac{\sum_{i=1}^{n} Ie_i}{n} \quad , \quad Ie_i = \left\{ \begin{array}{l} 1\ if\ Ye_i = \hat{Y}e_i, \\ 0\ otherwise \end{array} \right\} \tag{10}$$

$$Rs = \frac{\sum_{i=1}^{n} Is_i}{n} \quad , \quad Is_i = \left\{ \begin{array}{l} 1\ if\ Ys_i = \hat{Y}s_i, \\ 0\ otherwise \end{array} \right\} \tag{11}$$

$$ra = \frac{\sum_{i=1}^{n} Ir_i}{n} \quad , \quad Ir_i = \left\{ \begin{array}{l} 1\ if\ Ye_i = Ys_i, \\ 0\ otherwise \end{array} \right\} \tag{12}$$

$$G = \frac{\sum_{i=1}^{n} IG_i}{n} \quad , \quad IG_i = \left\{ \begin{array}{l} 1\ if\ \hat{Y}e_i = \hat{Y}s_i, \\ 0\ otherwise \end{array} \right\} \tag{13}$$

$$C = \frac{\sum_{i=1}^{n} IC_i}{n} \quad , \quad IC_i = \left\{ \begin{array}{l} 1\ if\ Ie_i = Is_i = 0, \\ 0\ otherwise \end{array} \right\} \tag{14}$$

To analyze the results presented in Table 11, the following classification grades were taken into consideration for parameters *Re*, *Rs*, *ra*, and *G*:

Table 11: 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. *Re* stands for environmental predictability, *Rs* for human control, *ra* for achievement, *G* for modeled knowledge, and *C* for unmodeled knowledge.

Participant	NRep	*Re*	*Rs*	*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%

- Excellent for results between 85% and 100%;
- Very Good for results between 75% and 84%;
- 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, *Rs*, 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 *Rs* are similar to the online BCI results for the parameter DC presented in Table 6, 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 *Re* should be 100%, if the wheelchair's navigation system worked as expected. According to Table 11 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 11), 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.

6. Conclusions and Future Work

This paper presents an assisted navigation system based on collaborative control, which 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 virtual-constraint layer, and an intent-matching layer. The ANS was successfully tested with eleven participants, 10 able-bodied and one motor disabled. 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. The ANS also gave indications of good adaptability to different navigation scenarios.

In the actual stage of the project, 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. Moreover, context dependent BCI is also being researched, since it can significantly improve the flexibility of the interface. Adaptation is not carried out in an automatic manner. This means that the human progress in 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. 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. Moreover, 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 also presents some limitations in cluttered environments. Additional sensor data is required to obtain a more reliable localization system. Currently, the map is given to the system and is not updated. The integration of SLAM strategies is in progress.

7. Acknowledgments

This work was supported by the Portuguese Foundation for Science and Technology (FCT) under Grant RIPD/ADA/109661/2009.

References

[1] G. Pires, U. Nunes, A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic, Journal of Intelligent and Robotic Systems 34 (2002) 301–314.

[2] A. Lopes, F. Moita, U. Nunes, 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.

[3] A. Kubler, N. Birbaumer, Brain-computer interfaces and communication in paralysis: Extinction of goal directed thinking in completely paralysed patients?, Clinical Neurophysiology 119 (2008) 2658 – 2666.

[4] A. Lopes, G. Pires, L. Vaz, 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 (IROS'11), San Francisco, USA.

[5] A. Lopes, G. Pires, 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 (IROS'12), Vilamoura, Portugal.

[6] Robchair project web page, online:http://robchair.isr.uc.pt, 2012.

[7] G. Pires, U. Nunes, M. 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 195 (2011) 270–281.

[8] G. Pires, U. Nunes, 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 123 (2012) 1168–1181.

[9] S. Levine, D. Bell, A. Jaros, R. Simpsons, Y. Koren, J. Borenstein, The NavChair assistive wheelchair navigation system, IEEE Transactions on Rehabilitation Engineering 7 (1999) 443–451.

[10] T. Taha, J. Miró, 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.

[11] D. Vanhooydonck, E. Demeester, A. Hüntemann, J. Philips, G. Vanacker, H. V. Brussel, M. Nuttin, Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model, Robotics and Autonomous Systems 58 (2010) 963 – 977.

[12] F. Leishman, O. Horn, G. Bourhis, Smart wheelchair control through a deictic approach, Robotics and Autonomous Systems 58 (2010) 1149 – 1158.

[13] R. Grasse, Y. Morère, A. Pruski, Assisted navigation for persons with reduced mobility: path recognition through particle filtering (condensation algorithm), Journal of Intelligent and Robotic Systems (2010) 19–57.

[14] X. Perrin, R. Chavarriaga, F. Colas, R. Siegwart, J. del R. Millán, Brain-coupled Interaction for Semi-autonomous Navigation of an Assistive Robot, Robotics and Autonomous Systems 58 (2010) 1246–1255.

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

[16] T. Carlson, Y. Demiris, Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload, IEEE Transactions on Systems, Man, and Cybernatics - Part B: Cybernetics 42 (2012) 876–888.

[17] A. Poncela, C. Urdiales, E. J. Pérez, F. Sandoval, A new efficiency-weighted strategy for continuous human/robot cooperation in navigation, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 39 (2009) 486–500.

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

[19] G. Pfurtscheller, N. Christa, A. Scholgl, K. Lugger, Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters, IEEE Trans. Rehab. Eng. 6 (1998) 316–324.

[20] L. Farwell, E. Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event related brain potentials, Electr. and Clin. Neuroph. 70 (1988) 510–523.

[21] G. Pires, 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, pp. 4193 –4198.

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

[23] G. Vanacker, J. del R. Millán, E. Lew, P. Ferrez, F. Galán, J. Philips, H. V. Brussel, M. Nuttin, Context-based filtering for assisted brain-actuated wheelchair driving, Comput. Intell. Neurosci 2007 (2007).

[24] J. Philips, J. del R. Millán, G. Vanacker, E. Lew, F. Galan, P. Ferrez, H. Van Brussel, M. Nuttin, Adaptive shared control of a brain-actuated simulated wheelchair, in: Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on, pp. 408 –414.

[25] F. Galán, M. Nuttin, E. Lew, P. Ferrez, G. Vanacker, J. Philips, J. del R. Millán, A brain-actuated wheelchair: Asynchronous and non-invasive brain-computer interfaces for continuous control of robots, Clinical Neurophysiology 119 (2008) 2159 – 2169.

[26] D. Huang, K. Qian, D.-Y. Fei, W. Jia, X. Chen, 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 20 (2012) 379 –388.

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

[28] B. Rebsamen, G. Cuntai, Z. Haihong, W. Chuanchu, T. Cheeleong, M. Ang, E. Burdet, A brain controlled wheelchair to navigate in familiar environments, IEEE Transactions on Neural Systems and Rehabilitation Engineering 18 (2010) 590–598.

[29] B. Gerkey, R. T. Vaughan, 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.

[30] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems, Science, and Cybernetics SSC-4 (1968) 100–107.

[31] J. Pearl, J. H. Kim, Studies in semi-admissible heuristics, IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (1982) 392–399.

[32] R. Solea, U. Nunes, Robotic wheelchair trajectory control considering user comfort, Informatics in Control, Automation and Robotics LNEE, Springer 37 (2009) 113–125.

[33] I. Ulrich, J. Borenstein, VFH+: Reliable obstacle avoidance for fast mobile robots, in: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00), Leuven, Belgium.

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

[35] S. Thrun, W. Burgard, D. Fox, Probablistic Robotics, The MIT Press, 2006.

[36] A. Diosi, 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.

[37] L. Kneip, F. Tache, G. Caprari, R. Siegwart, Characterization of the compact hokuyo urg-04lx 2d laser range scanner, in: IEEE International Conference on Robotics and Automation, 2009. ICRA '09., pp. 1447 –1454.

[38] A. Lopes, U. Nunes, 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.

[39] L. Montesano, M. Díaz, S. Bhaskar, J. Minguez, Towards an intelligent wheelchair system for users with cerebral palsy, IEEE Transactions on Neural Systems and Rehabilitation Engineering 18 (2010) 193–202.

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

[41] L. Rothrock, 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 33 (2003) 58–72.

[42] J. Yin, L. Rothrock, A rule-based lens model, International Journal of Industrial Ergonomics 36 (2006) 499–509.