

# Wheelchair Navigation Assisted by Human-Machine Shared-control and a P300-based Brain Computer Interface

Ana C. Lopes, Gabriel Pires, Luís Vaz, and Urbano Nunes

**Abstract**—This paper presents a new shared-control approach for assistive mobile robots, using Brain Computer Interface (BCI) as the Human-Machine Interface (HMI). A P300-based paradigm that allows the selection of brain-actuated commands to steer a Robotic Wheelchair (RW), is proposed. At least one specific motor skill, such as the control of arms, legs, head or voice, is required to operate a conventional HMI. Due to this reason, they are not suited for people suffering from severe motor disorders. BCI may open a new communication channel to these users, since it does not require any muscular activity. The number of decoded symbols per minute (SPM) in a BCI is still very low, which means that users can only provide sparse, and discrete commands. The RW must rely on the navigation system to validate user commands effectively. A two-layer shared-control approach is proposed. The first, a virtual-constraint layer, is responsible for enabling/disabling the user commands, based on certain context restrictions. The second layer is an user-intent matching responsible for determining the suitable steering command, that better fits the user command, taking the user competence on steering the wheelchair into account. Experimental results using Robchair, the RW platform developed at ISR-UC [1], [2] are presented, showing the effectiveness of the proposed methodologies.

## I. INTRODUCTION

This research work aims to develop assistive navigation techniques to increase autonomy of people with severe motor disabilities. Assisted navigation is a challenging research topic, since it has to deal with robots navigating in semi-structured and unstructured dynamically changing environments, with a high level of uncertainty, and being able to interact with human users in a safe manner. Usually assistive navigation is a type of semi-autonomous architecture requiring at least two agents, a human agent, and a machine agent (MA), sharing the control of the robot. Shared-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 the control belongs solely to the robot [3]. In recent years different shared-control architectures were developed for applications in the fields of assistive robotics, such as intelligent wheelchairs [1], [4], and [5], minimally invasive surgery [6], or intelligent mobility assistants [7], and [8]. When dealing with semi-autonomous systems, care must be

taken in the choice and the design of the most appropriate system interface. The HMI must be well suited to the user needs and capabilities, and the semi-autonomous system must be able to identify, and respond to the user requests and commands in the most adequate and safe manner.

People with motor disorders such as amyotrophic lateral sclerosis, progressive muscular dystrophy, cerebral palsy and spinal cord injuries are unable or have great difficulty to control standard interfaces. Brain computer interfaces (BCIs) open a new communication channel that is independent of muscular activity. It has already been shown that these patients can use BCI to spell characters with acceptable communication rates [9], [10], [11]. The use of a BCI to control a robotic wheelchair could help these patients to increase their autonomy and improve their quality of life. The use of BCI control of physical devices is a research topic of growing importance. The number of decoded symbols per minute (SPM) in a BCI is however still very low, which means that users can only provide a few discrete commands per minute (less than 10 SPM). Thus, the control of a wheelchair must rely on a navigation system that receives sparse commands from the user and that performs safe and smooth manoeuvres according to steering information. Several brain-actuated wheelchairs have already been proposed following different neuromechanisms. Motor imagination, a neuromechanism based on sensorimotor rhythms, is used in [12] to detect 3 mental steering commands (forward, left and right) and in [13] to detect left and right mental states. The P300 signal, an event related potential elicited by an oddball paradigm [14] is used in [15] and [16]. P300 is an evoked potential characterized by a positive peak that occurs about 300 ms after the onset of a target event in an oddball paradigm (a random and rare target event among frequent non-target events). In [15] the BCI is used to select high-level predefined locations and in [16] it is used to select low-level locations in a 3D map of the surrounding environment.

This paper proposes a P300-based paradigm that allows the selection of brain-actuated commands to steer a RW. To alleviate user effort, low-level commands are only issued when there are dynamic changes of the environment or when ambiguous situations occur. A two-layer shared-control approach is proposed to obtain a safe and effective navigation of the RW, receiving user commands that are issued sparsely.

## II. NAVIGATION SYSTEM ARCHITECTURE USING BCI

In this section, a navigation architecture capable of providing intelligent motion control of semi-autonomous RW is conceptually described. The architecture presented in Fig.

This work was supported by the Portuguese Foundation for Science and Technology (FCT) under Grant RIPD/ADA/109661/2009, and PTDC/SEN-TRA/099413/2008.

Ana C. Lopes, Gabriel Pires, Luís Vaz, and Urbano Nunes are with the Institute for Systems and Robotics, University of Coimbra, Polo II, 3030-290 Portugal. Ana C. Lopes and Gabriel Pires are also with the Instituto Politécnico de Tomar, and acknowledge a PROTEC research fellowship under Grant SFRH/BD/49251/2008, and Grant SFRH/BD/49881/2009 respectively. {anacris, gpires, lvaz, urbano}@isr.uc.pt

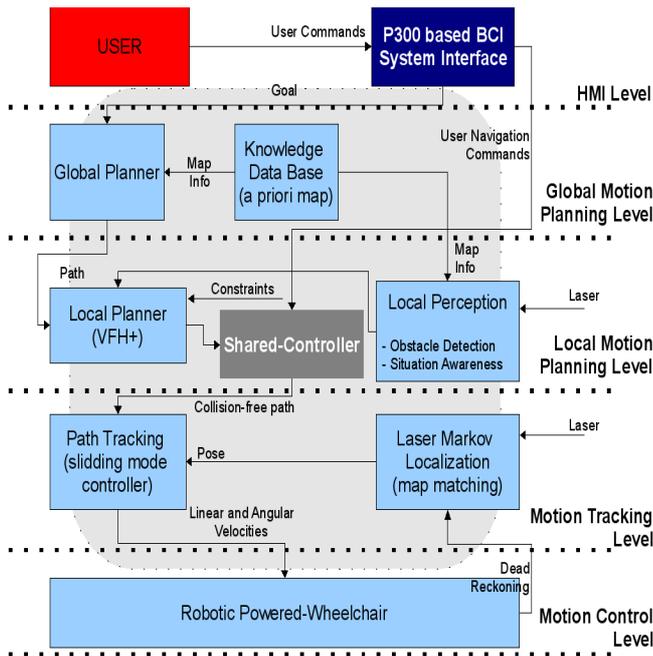


Fig. 1. Assistive navigation system architecture.

1 is structured in five levels: HMI, global motion planning, local motion planning, motion tracking, and motion control. This architecture has been tested in a player/stage [17] simulation environment, and in Robchair [1].

As HMI, a P300-based BCI is used to provide the user intent: the final localization goal, and steering commands issued sparsely. The global planner determines the trajectory to a predefined goal, based on the information provided by the a priori grid map. The local planner calculates new trajectories to avoid new obstacles in the environment. Obstacle detection is carried out based on laser information. A Markov localization system is used to fuse dead-reckoning data with laser map matching [18]. A sliding mode controller is used for path tracking [19]. The shared-controller determines the set of appropriate manoeuvres to reach a predefined goal based on user and machine commands.

### III. SYSTEM INTERFACE: BCI

#### A. Paradigm

The proposed P300-based paradigm is an upgrade of our earlier paradigm introduced in [20] and it is henceforth referred as Arrow Paradigm (Fig. 2). It provides a small set of steering options that includes both low-level commands ('FORWARD', 'RIGHT', 'ROR', 'STOP', 'LEFT', 'ROL', and 'BACK'), high-level commands indicating a goal ('ROOM A', 'ROOM B', 'ROOM C'), and basic interaction communication ('YES', 'NO'). Each symbol/word is intensified during 100 ms and with an inter-stimulus interval (ISI) settled to 75 ms. When the symbol/word is intensified (event occurrence), the color is changed and the size is slightly increased. This aims to enhance user perception of the events. For every round, each symbol/word is intensified once in a

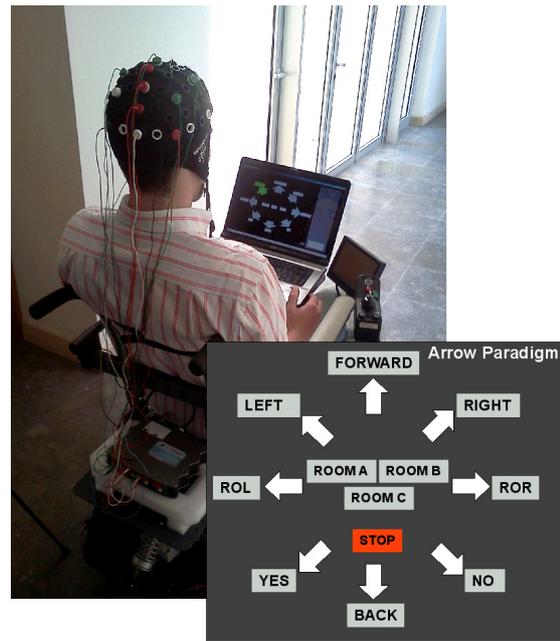


Fig. 2. Overview of the Robchair and screenshot of the Arrow Paradigm.

random order. The target event corresponds to the symbol mentally selected by the user. All other symbols are called standard or non-target events. The total number of events is 12 which establishes a target event probability of 1/12. The data segment (epoch) associated to each event has a duration of 1 second. The detection of a target event can occur after a single round of events or after several rounds, which depends on the performance of the user.

#### B. EEG signal acquisition and static analysis

EEG signals are acquired with a g.tec gUSBamp amplifier from 12 Ag/Cl electrodes at positions Fz, Cz, C3, C4, CPz, Pz, P3, P4, PO7, PO8, POz and Oz of the international extended 10-20 standard system. The electrodes were referenced to the left or right ear lobe and the ground was placed at AFz. Signals were sampled at 256 Hz and filtered through 0.1-30 Hz bandpass filter and a 50 Hz notch filter.

1) *BCI online results:* A static analysis was carried out to characterize the used BCI system [11]. The experiments were performed with 3 able-bodied volunteers. To assess the online performance it was asked to each user to select a set of 15 consecutive commands. This task was performed without movement of the wheelchair. The number of repetitions (Nrep) needed to obtain a 100% classification accuracy, and the respective time needed to issue a command (trial time - TT) are shown in Table I. Participants A and C can issue a command each 10.6 s and participant B each 6.7 s.

#### C. Inter-operating systems and BCI calibration

The BCI module and the navigation module run on two different computer systems that communicate through TCP/IP. The BCI system sends steering commands to the navigation system and receives requests from it. The navigation module can also activate or deactivate the BCI system.

TABLE I

RESULTS OF ONLINE EXPERIMENTS FOR A 100% ACCURACY AFTER 15 CONSECUTIVE STEERING COMMANDS.

Participant	NRep	TT (s)
A	5	10.6
B	3	6.7
C	5	10.6

Online operation of the BCI system must be preceded by a calibration phase that takes about 5 minutes. During the calibration, a dataset with 90 target epochs and 990 non-target epochs is gathered. This labeled dataset is then used to obtain the classification algorithms. The classification approach uses the same methodology as in our previous work, which showed state of the art results with a P300-based row-column speller [11]. It uses a statistical spatial filter that cascades a Fisher beamformer and a Max-SNR beamformer (C-FMS). The twelve input channels are transformed into two high SNR projections, which are then feed to a naïve Bayes classifier (NB). Because the signal-to-noise ratio (SNR) of the signal is very low, it is necessary to combine several repetitions of the same event. The number of repetitions depends on the user performance. In online operation the classifier algorithm is applied to each event and then the target associated with the highest classification score is selected.

#### IV. LOCALIZATION AND MAPPING

The assistive navigation architecture is provided with an a priori grid map of the environment. Localization is performed using dead-reckoning data (odometry) for rough positioning, and laser data for map matching. A Markov localization system was designed to fuse odometry with laser map matching [21], [18]. The Belief  $Bel(x_t)$  is evaluated for every possible state  $x_t$ , however, in this case localization method was only applied locally, regarding the odometry information. Markov localization is composed by two stages: a prediction stage, and a correction stage, as presented in Algorithm 1, where  $Bel^-(x_t)$ , and  $Bel(x_{t-1})$  are the predicted posterior and prior probabilities, respectively, and  $Bel(x_t)$  represents the posterior probability for the correction stage.

In the correction stage, a map matching method is applied, where the sensor measurement model compares a local map  $m_{local}$  (given by a laser range scan) to the predicted map  $m_{ref}$  (based on a priori map  $m$ ), such that the more similar  $m_{ref}$  and  $m_{local}$ , the larger  $P(m_{local}|x_t, m)$ . It is important to highlight that map matching is only applied to a submap of the global map, which is determined based on odometry data, and taken the odometry model into account. To improve the performance of the Markov localization system, two filters were applied before map matching, namely: a distance filter, and a dynamic obstacle filter. The former reduces the maximum range to a predefined threshold ( $DistFilter(m_{local})$ ), and the latter removes all range data resulting from dynamic obstacles ( $DynamicObstFilter(m'_{local})$ ), which are not represented in the a priori map. Map matching is summarized in Algorithm 2, where the  $ReferenceMap(m, x_t)$  function

determines the predicted map to be matched to the filtered local map  $m''_{local}$ ,  $\bar{m}$  is the average map value, and  $\theta_{bg}$  represents the scan bearing.

Figure 3 shows some experimental results of the implementation of the Markov Localization algorithm using Robchair.

---

#### Algorithm 1 MarkovLocalization( $Bel(x_{t-1}), u_t, m_{local}, m$ )

---

```

for all  $x_t \in m_s$  do
  //Prediction Stage
   $Bel^-(x_t) \leftarrow \int P(x_t|u_{t-1}, x_{t-1})Bel(x_{t-1})dx_{t-1}$ 
  //Correction Stage
   $Bel(x_t) \leftarrow MapMatching(m_{local}, m, x_t)$ 
end for
return  $Bel(x_t)$ 

```

---



---

#### Algorithm 2 MapMatching( $m_{local}, m, x_t$ )

---

```

 $m'_{local} \leftarrow DistFilter(m_{local})$ 
 $m''_{local} \leftarrow DynamicObstFilter(m'_{local})$ 
 $m_{ref} \leftarrow ReferenceMap(m, x_t)$ 
for all points  $\in m''_{local}$  do
   $\bar{m} = \frac{1}{2N} \sum_{\theta_{bg}} (m_{ref} + m''_{local})$ 
end for
for all points  $\in m''_{local}$  do
   $\rho_{m_{ref}, m''_{local}, x_t} = \frac{\sum_{\theta_{bg}} (m_{ref} - \bar{m})(m''_{local} - \bar{m})}{\sqrt{\sum_{\theta_{bg}} (m_{ref} - \bar{m})^2 \sum_{\theta_{bg}} (m''_{local} - \bar{m})^2}}$ 
end for
 $P(m_{local}|x_t, m) = \max(\rho_{m_{ref}, m''_{local}, x_t}, 0)$ 
return ( $P(m_{local}|x_t, m)$ )

```

---

#### V. PLANNING AND NAVIGATION

##### A. Global planner

The A\* algorithm [22], [23] finds 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, which is an estimation of the distance from the current node to the goal.

##### B. Obstacle detection and local planner

A vector field histogram [24] is determined for obstacle detection purposes, using laser range finder data. The local perception module includes a context situation module that is responsible to determine if an obstacle (detected by the obstacle detection module) is new to the environment. Every time a new obstacle is detected, the local planner is activated, and the VFH+ method is used to determine the best free steering direction. Obstacle detection is carried out using a laser range finder located at the front of the RW. These methodologies are only applied to new static obstacles. In case of approaching a moving obstacle, it stops.

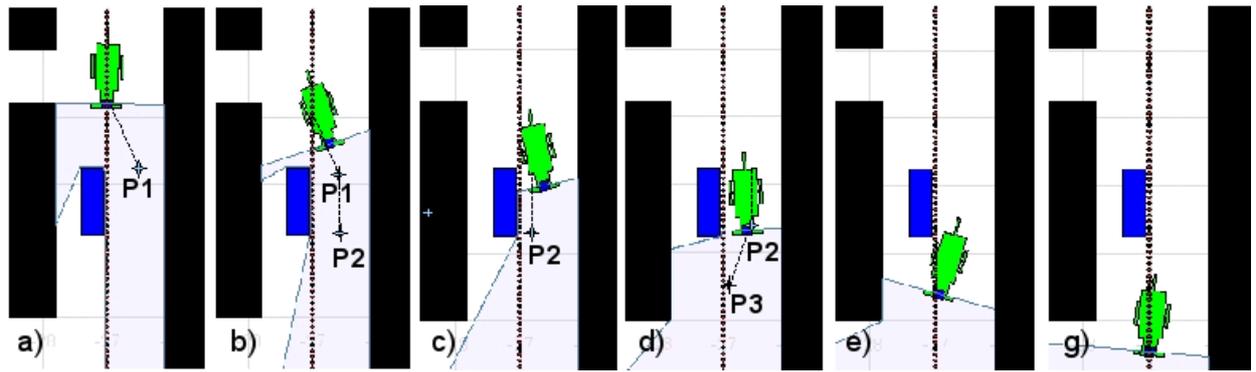


Fig. 4. Obstacle avoidance based on VFH+ local planner: a) robot detects the obstacle and determines a free path to a target pose  $P1$  (center point of the free space between the obstacle and the infra-structure); b) robot follows the path; c) robot reaches  $P1$ , and determines a path to pose  $P2$ ; 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.

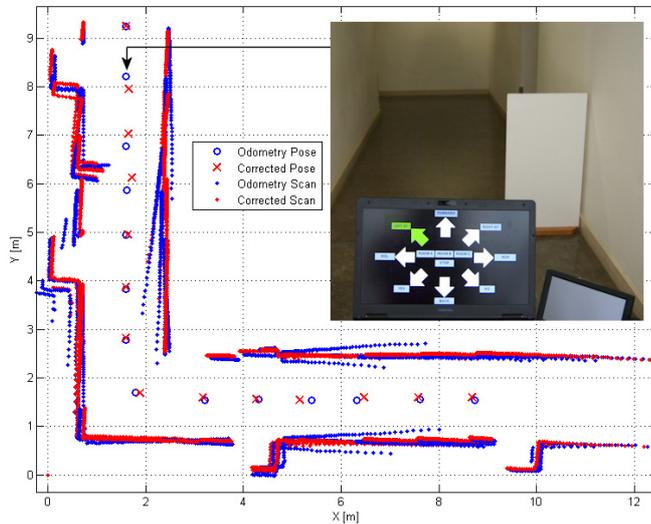


Fig. 3. Markov Localization results: odometry pose; and scan mapping based on odometry pose; corrected pose and scan mapping based on corrected pose.

1) *VFH+ Algorithm:* The VFH+ [24] module determines a cost function  $g(c)$  for a candidate direction  $c$  as follows:

$$g(c) = \mu_1 \Delta(c, k_t) + \mu_2 \Delta(c, \theta_i) + \mu_3 \Delta(c, k_{n,i-1}) \quad (1)$$

where  $k_t$  is the target direction,  $\theta_i$  gives the current direction of the RW, and  $k_{n,i-1}$  gives the previously selected direction of motion. The generic term  $\Delta(c_1, c_2)$  gives the absolute angle difference between two sectors  $c_1$  and  $c_2$ . Terms  $\mu_1$ ,  $\mu_2$  and  $\mu_3$  are the cost parameters. In order to achieve a good blending between the global and the local planner, the target direction  $k_t$  is a subgoal of the reference path, i.e. is the next subgoal provided by the global planner. Figure 4 shows how the local planner works to avoid one obstacle placed in the robot trajectory. The VFH+ method is used to calculate the best steering direction to avoid the obstacles in the environment. Using a simple path-planner method three local paths are determined according to the best steering direction provided by the VFH+ module. As depicted in Fig.

4, after detecting the obstacle, a new path is planned on the direction provided by the VFH+ module, in order to reach target pose  $P1$  (center point of the free space between the obstacle and the infra-structure). After reaching  $P1$  a new path is calculated in order to reach  $P2$ . A third path to  $P3$  is then planned to merge the local path to the global one.

A sliding mode path-following controller was used as the system path tracker [19].

### C. Shared-Control

The shared-control module depends on the user's ability to steer the RW. In this sense, user characterization, must be carried out previously. An assistive navigation training framework (ANTF) [25], was developed with two main goals in mind: to train users to carry out navigation tasks, in an autonomous manner (not requiring continuous help of therapists), and to characterize user models on steering a powered wheelchair. The ANTF is intended to train users to decide the best set of manoeuvres to reach a predefined final goal, as well as to train them using the system interface, in this case a brain computer interface. The ANTF platform classifies users in three stages of development, according to their steering capabilities, namely: beginner, average, and advanced user. For each stage of development the user has a capability efficiency rate  $r_a$  [25] that is used in the intent-matching layer of the proposed shared-controller.

The shared-control architecture receives commands from two agents: a user agent (UA), and a machine agent (MA). The user issues BCI-actuated commands  $\theta_{UA}$  using the BCI. The proposed shared-control architecture 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 manoeuvres, taken into account her/his steering competence, as outlined in Fig. 6.

1) *Virtual-Constraint Layer:* The virtual-constraint layer (VCL) is responsible for enabling/disabling the user commands. This layer is required because the BCI system is continuously providing navigation commands, independently of user. While in a disabled state, the system becomes

autonomous. User commands are enabled by the VCL according to the following perceived situations:

- S1: Multiple possible directions to avoid an obstacle;
- S2: Solving a deadlock moving backwards;
- S3: Solving a deadlock with left/right pure rotations.

The VCL includes a situation aware module responsible for detecting the occurrence of the referred perceived situations. When one of those situations occur, the user is requested to choose a desired steering command, through the visual arrow paradigm shown in Fig. 2. Additionally, the VCL also takes into account constraints related to the user steering competence, as follows:

*Basic User :*

$$\theta_{VC} = \begin{cases} \theta_{UA} & \text{if}(S1 \ \&\& \ \theta_{UA} \in \{L, F, R\}) \\ \theta_{UA} & \text{if}(S3 \ \&\& \ \theta_{UA} \in \{ROL, ROR\}) \\ 0 & \text{otherwise} \end{cases}$$

*Average User :*

$$\theta_{VC} = \begin{cases} \theta_{UA} & \text{if}(S1 \ \&\& \ \theta_{UA} \in \{L, F, R\}) \\ \theta_{UA} & \text{if}(S2||S3) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

*Advanced User :*

$$\theta_{VC} = \begin{cases} \theta_{UA} & \text{if}(S1 \ || \ S2 \ || \ S3) \\ 0 & \text{otherwise} \end{cases}$$

where  $L \equiv LEFT$ ,  $F \equiv FORWARD$ ,  $R \equiv RIGHT$ ,  $ROL$  and  $ROR$  denote a pure rotation left and right, respectively.

2) *Intent-Matching Layer:* This layer determines the final steering command to the RW, based on user-intent properly modified by the VCL, and taking into account a set of steering directions proposed by the MA outputs. The error  $e_{\theta_{UA}}^i$  between the user command, and the directions provided by the MA is calculated as follows:

$$e_{\theta_{UA}}^i = \theta_{VC} - \theta_{MA}^i \quad i = 1, \dots, n \quad (3)$$

where  $n$  is number of MA proposed directions. Each proposed direction has an associated machine weight, which is defined by the machine steering weight vector  $\eta_{MA} = [\eta_{MA}^1 \ \eta_{MA}^2 \ \dots \ \eta_{MA}^n]^T$ . For a perceived situation S1, the MA always determines a steering direction  $\theta_{MA}$ , which is selected according to the cost function (1). For all the candidate directions a cost function  $g(\theta)$  is defined as follows:

$$g(\theta) = \eta_{MA} \cdot e_{\theta_{MA}}^i + \eta_{UA} \cdot r_a \cdot e_{\theta_{UA}}^i \quad (4)$$

where  $e_{\theta_{MA}}^i$  is the error between the selected direction, and each of the proposed directions,

$$e_{\theta_{MA}}^i = \theta_{MA}^i - \theta_{MA} \quad i = 1, \dots, n \quad (5)$$

The weight  $\eta_{UA}$  is defined according to the user steering competence, and  $r_a$  denotes her/his efficiency rate, which varies according to  $r_a \in [0..1]$ . The selected direction  $\theta$  is the one that minimizes the cost function  $g(\theta)$ . For a deadlock perceived situation (S2 and S3), the MA is not able to determine any free direction, and, in that case, the user must move the RW backwards or perform pure rotations left or right (commands BACK, ROL, and ROR, respectively) to leave the deadlock, and attain a S1 situation.

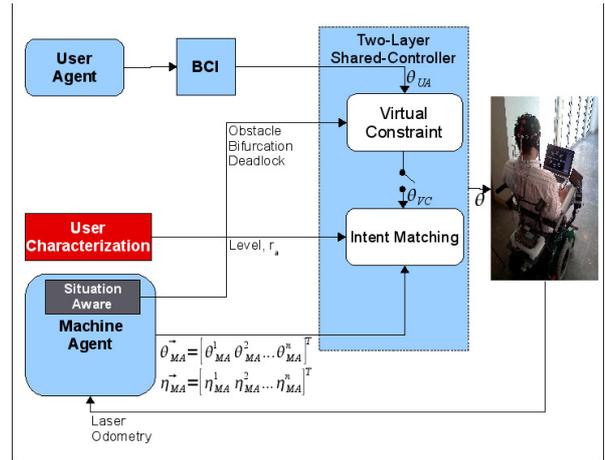


Fig. 5. Two-layer shared-control architecture.

## VI. EXPERIMENTAL RESULTS

### A. System setup - Robchair

Figure ?? illustrates the Robchair control architecture. The wheelchair is composed by two motorized rear wheels, and 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. It has been equipped with several devices such as two power-driver modules, which can guarantee an independent control of each motor, joystick, and several sensors such as: magnetic sensor ruler, laser range finder (LSR), inertial measurement unit (IMU), and ultrasound. The wheelchair also includes an industrial embedded PC, mounted on the front, powered by Linux with RTAI for real-time processing. The embedded PC connects to sensors and actuators through CAN fieldbus. The platform connects to external devices, such as an industrial laptop, through a wireless link.

### B. Wheelchair on road

The user can not be continuously issuing commands because it would be tiresome. The navigation system was designed to reduce the user effort to a minimum. In the current experiments the user selects global goals, and the navigation module follows the paths determined by the global planner to reach the goals. The navigation module considers the BCI input commands according to user steering competence, and according to situation awareness, more precisely, if situations S1, S2 or S3 occur. These situations were experimentally tested, and results are presented in the sequel. An automatic switch on/off of the BCI system directly controlled by the user is also being researched but this issue is beyond the scope of this paper.

1) *Direction selection to avoid an obstacle (S1):* Figure 7 shows results related to the navigation of a wheelchair in an office-type building scenario. These results are related to situation S1 described as multi-direction possibilities to avoid new obstacles in the trajectory. This experiment was simulated in player/stage environment for three types of

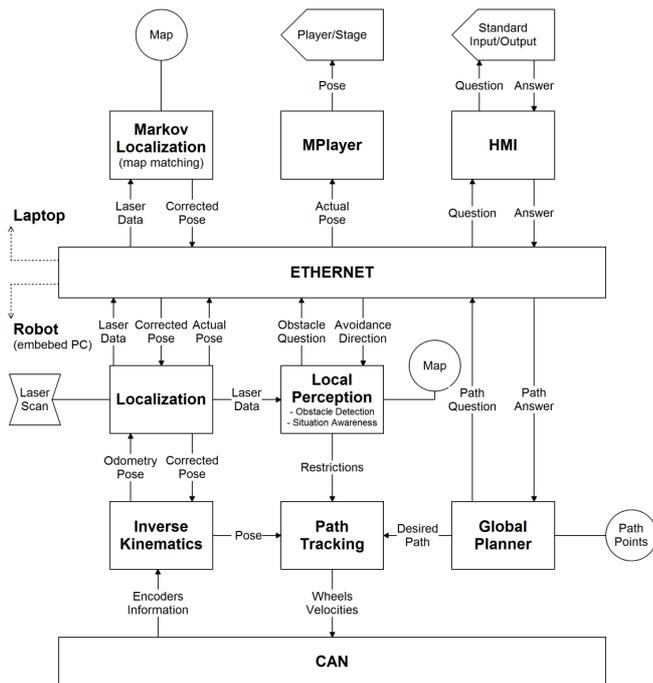


Fig. 6. Robchair data flow diagram.

users, according to their steering competence: basic, average, and advanced. According to Fig. 7, when position 1 is reached, the navigation system faces an ambiguous situation, and the user is requested to select the desired direction. The final steering direction is then calculated according to (4). For these experiments, machine steering weights  $\eta_{MA}^i$  are equal to 1, and user weights are as follows:  $\eta_{UA}^{basic} = 2$ ,  $\eta_{UA}^{average} = 3$ , and  $\eta_{UA}^{advanced} = 4$ . An efficiency rate  $r_a = 1$  was also considered in all experiments. According to user weights, and considering an efficiency 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 always have less power than the MA. Of course, with lower efficiency rates user power can be reduced to nearly zero, and the system may become purely autonomous, for any type of users. The results depicted in Fig. 7 are similar for all types of users, with weights and efficiency rate described previously. Figure 8 shows a S1 experiment using Robchair, the real platform. Robchair detects an obstacle, and the MA proposes two steering directions due to the door opening, which results in an inappropriate steering direction, leading to a dubious situation. The user chooses the appropriate steering direction (LEFT command), the obstacle is avoided, and Robchair is able to reach the final localization goal.

2) *The deadlock problem (S2, S3):* Figure 9 shows how different users solve the deadlock problem, after selecting the wrong direction. In case of basic users, they are not allowed to move backwards, and they can only perform pure rotations (ROL or ROR) if they are in a deadlock situation, where the MA is not able to determine any free steering direction.

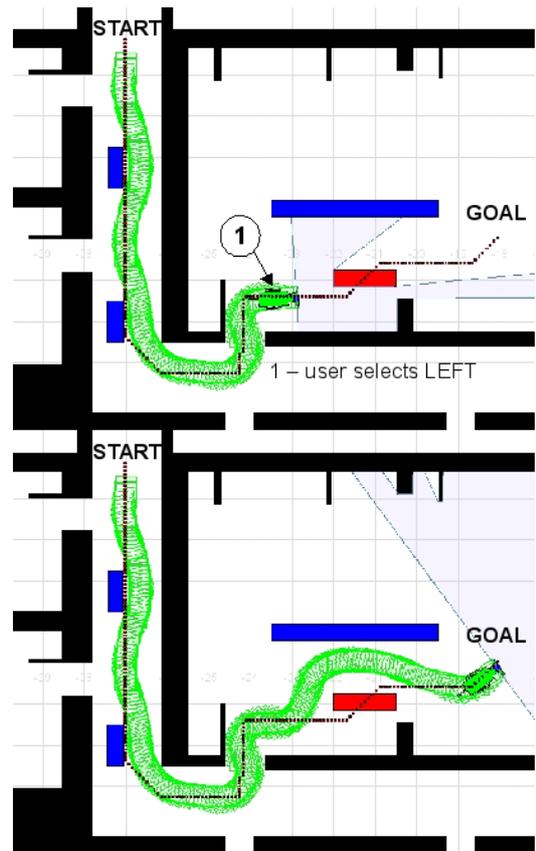


Fig. 7. Direction selection to avoid an obstacle. 1 - user selects to move left to avoid the obstacle.

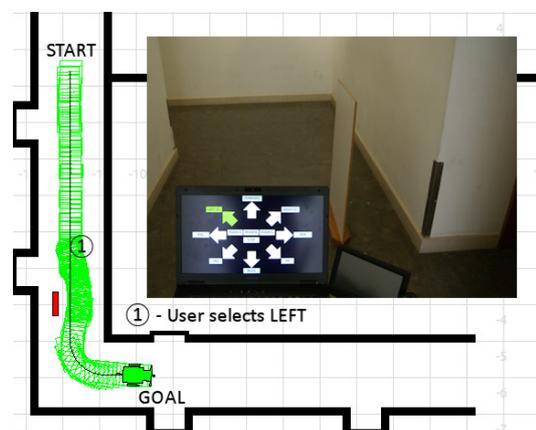


Fig. 8. Direction selection to avoid an obstacle. 1 - user selects to move left to avoid the obstacle. The MA proposes two steering directions due to the door opening.

Average and advanced users can solve the deadlock because they are allowed to move backwards in case of entering the deadlock.

## VII. CONCLUSION AND FUTURE WORK

This paper presents an assistive navigation architecture based on shared-control, and using P300-based BCI paradigm, which allows the selection of brain-actuated com-

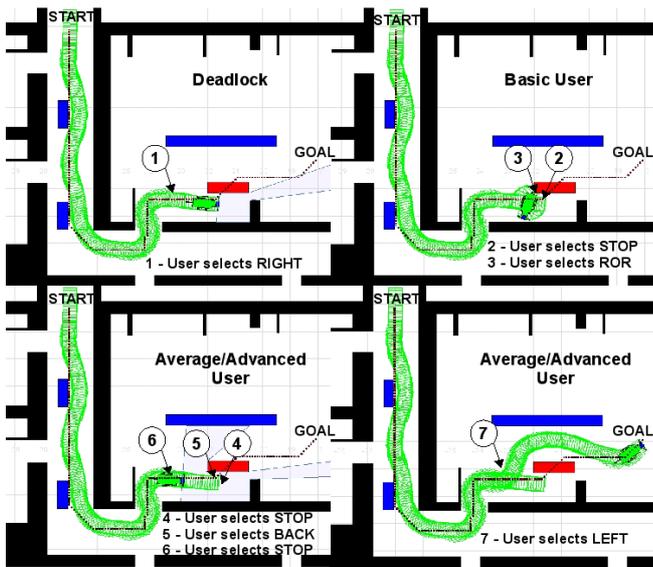


Fig. 9. Experimental results: each user tries to solve the deadlock problem: 1 - user selects RIGHT and enters the deadlock; 2, 3 - basic user stops and tries to perform a pure rotation right (ROR) but gets stuck; 4,5 - average/advanced user stops and move backwards (BACK); 6,7 - average/advanced user stops and selects LEFT.

mands. Since BCI-actuated commands are issued sparsely, an assistive navigation architecture based on a two-layer shared controller was designed, and implemented in player/stage environment and in Robchair (ISR-UC wheelchair platform). The shared-control architecture, includes a virtual-constraint layer, and an intent-matching layer.

In the actual stage of the project, the BCI paradigm always presents a constant set of commands to users (all the arrow paradigm options shown in Fig. 2). Research is being carried out to have a variable set of commands, which is function of the situation. An automatic switch on/off of the BCI system directly controlled by the user is also being researched. A new dynamic local planner that takes into account robot geometry to plan more efficient obstacle avoidance manoeuvres, and that be able to effectively avoid slow dynamic obstacles, such as people and other moving objects is also being pursued.

## REFERENCES

- [1] Gabriel Pires and Urbano Nunes. A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic. *Journal of Intelligent and Robotic Systems*, 34:301–314, July 2002.
- [2] A. Lopes, F. Moita, U. Nunes, and R. Solea. An outdoor guideway 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] R. Mukherjee S. S. Nudehi 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.
- [7] D. K. Liu J. V. Miro S. McLachlan, J. Arblaster 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.
- [8] K. Law and Y. Xu. Shared-control for navigation and balance of a dynamically stable robot. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA’02)*, volume 2, Washington, DC, USA, May 2002.
- [9] Andrea Kubler, Boris Kotchoubej, Jochen Kaiser, Jonathan R. Wolpaw, and Niels Birbaumer. Brain-computer communication: Unlocking the locked in. *Psychological Bulletin*, 127(3):358 – 375, 2001.
- [10] G. Townsend, B.K. LaPallo, C.B. Boulay, D.J. Krusienski, G.E. Frye, C.K. Hauser, N.E. Schwartz, T.M. Vaughan, J.R. Wolpaw, and E.W. Sellers. A novel P300-based brain-computer interface stimulus presentation paradigm: Moving beyond rows and columns. *Clin. Neurophysiol.*, 121(7):1109–1120, 2010.
- [11] 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*, 195(2):270–281, 2011.
- [12] 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*, 119(9):2159 – 2169, 2008.
- [13] K. Tanaka, K. Matsunaga, and H.O. Wang. Electroencephalogram-based control of an electric wheelchair. *IEEE Transactions on Robotics*, 21(4):762–766, Aug 2005.
- [14] 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.*, 70(6):510–523, 1988.
- [15] 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*, 18(6):590–598, Dec 2010.
- [16] 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*, 25(3):614–627, June 2009.
- [17] 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.
- [18] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2006.
- [19] Razvan Solea and Urbano Nunes. Robotic wheelchair trajectory control considering user comfort. *Informatics in Control, Automation and Robotics LNEE, Springer*, 37:113–125, 2009.
- [20] Gabriel Pires, Miguel Castelo-Branco, and Urbano Nunes. Visual P300-based BCI to steer a wheelchair: A bayesian approach. In *Proc. IEEE Int. Conf. of the Engineering in Biology and Medicine Society (EMBC2008)*, Vancouver, Canada, Aug 2008.
- [21] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
- [22] P.E. Hart, N.J. Nilsson, and B. Raphael. Determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, July 1968.
- [23] Judea Pearl and Jin H. Kim. Studies in semi-admissible heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:392–399, July 1982.
- [24] 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.
- [25] 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, Nov.