

SLAM-based Cross-a-Door Solution Approach for a Robotic Wheelchair

Fernando A. Auat Cheein, Celso De La Cruz, Teodiano F. Bastos and Ricardo Carelli

Instituto de Automatica, National University of San Juan, San Juan, Argentina.

Technological Center II, Federal University of Espirito Santo, Espirito Santo, Brazil

Corresponding author E-mail: fauat@inaut.unsj.edu.ar

Abstract: This paper proposes a solution to the cross-a-door problem in unknown environments for a robotic wheelchair commanded through a Human-Machine Interface (HMI). The problem is solved by a dynamic path planning algorithm implementation based on successive frontier points determination. An adaptive trajectory tracking control based on the dynamic model of the robotic wheelchair is implemented on the vehicle to direct the wheelchair motion along the path in a smooth movement. An EKF feature-based SLAM is also implemented on the vehicle which gives an estimate of the wheelchair pose inside the environment. The SLAM allows the map reconstruction of the environment for safe navigation purposes. The whole system steers satisfactorily the wheelchair with smooth movements through common doorways which are narrow considering the size of the vehicle. Implementation results validating the proposal are also shown in this work.

Keywords: Robotic Wheelchair, SLAM, Trajectory Tracking, Door Crossing, Assistance Robotics

1. Introduction

The integration of robotic issues into the medical field has become of great interest in recent years. Service, assistance, rehabilitation and surgery are the more benefited human health-care areas by the recent advances in robotics. Specifically, autonomous and safe navigation of wheelchairs inside known and unknown environments is one of the important goals in assistance robotics.

A robotic wheelchair can be used to allow people with both lower and upper extremity impairments or severe motor dysfunctions overcome the difficulties in driving a wheelchair. The robotic wheelchair system integrates a sensory subsystem, a navigation and control module, and a user-machine interface to guide the wheelchair in autonomous or semi-autonomous mode (Mazo, M., 2001; Bourhis, G. et al., 2001; Zeng, Q. et al., 2008; Parikh, S. et al., 2007). In autonomous mode, the robotic wheelchair goes to the chosen destination without any participation of the user in the control process. This mode is intended for people who have great difficulties to guide the wheelchair. In the semi-autonomous mode the user shares the control with the robotic wheelchair. In this case only some motor skills are needed from the user.

A door crossing system becomes a very important module for autonomous navigation of vehicles because when combined with wall-following and corridor-following modules renders a complete navigation system for indoor environments. For example the works (Muñoz, R. et al., 2005) and (Poncela, A. et al., 2007) use door crossing, wall-following and corridor-following as skills (ability to accomplish certain sub-goal) in the navigation system for autonomous mobile robots. However, these

works do not consider smooth movements of the vehicle neither narrow doorway-crossing in the design of the door crossing system, which are common requirements in the design of robotic wheelchair navigation architectures. One of the solutions to the autonomy problem of mobile vehicles can be provided by the implementation of a SLAM (Simultaneous Localization and Mapping) algorithm (Siegwart, R. & Nourbakhsh, I. R., 2004). SLAM is a recursive probabilistic algorithm that concurrently builds a map of the environment while it localizes the mobile vehicle at the same time, minimizing errors (Dissanayake, G. et al., 2001). Although this algorithm is processing time-demanding, it becomes a powerful solution when the vehicle has to navigate through unsensored or unknown environments, obtaining a reliable map of it (Kouzoubov, K. & Austin, D., 2004). From its early beginning, the SLAM has been implemented in several algorithms (Chatila, R. & Laumond, J. P., 1985; Siegwart, R. & Nourbakhsh, I. R., 2004), being the EKF (Extended Kalman Filter) the most used by the scientific community (Durrant-Whyte, H. & Bailey, T., 2006; Durrant-Whyte, H. & Bailey, T., 2006a). The Particle Filter (PF) and the Unscented Kalman Filter (UKF) have proven to be better approaches to the SLAM problem. The Particle Filter solves the gaussianity restriction of the models involved in the SLAM (Thrun, S. et al., 2005) whereas the UKF has shown a better performance dealing with non-linear models of the vehicle and the measurements (Thrun, S. et al., 2005). Despite of the fact that the map built by the SLAM could be of different types -topological, metric, hybrid (Thrun, S. et al., 2005)- the most used map is a metric feature-based map, which extracts some geometrical constraints

from the environment -e.g., corners, lines, color patterns (Xi, B. et al., 2008)-. These features are then used to localize the vehicle inside that environment.

Different Human-Machine Interfaces (HMI) have been proposed in the scientific literature to drive a mobile robot motion -including a robotic wheelchair-. A HMI could be any interface that uses somehow, human physiological signals to command a given device (Ferreira, A. et al., 2008). In (Galan, F. et al., 2008) a continuous Brain-Computer control is made to command the mobile robot motion along a predefined path. In (Millan, J. R. et al., 2004), a non-invasive Brain-Computer Interface (BCI) is used to drive a robot by three basic commands and in (Ferreira, A. et al., 2008) different HMI are presented using electromyographic signals to command the robot's movements. All these HMI share the following feature: they use the robotic device as an actuator, leaving it without any kind of intelligence nor autonomy and only with a basic low level behavioral control. The robot is constrained to execute a few basic movements commanded by the HMI, such as: turn to the right/left, move forward, move backward and stop. These commands are generated by a Finite State Machine (Millan, J. R. et al., 2004). When a robot is commanded by this kind of HMI, it is not able to perform activities that demand high precision movements or to reach places in the environment following pre-established trajectories.

In this paper, a solution to the cross-a-door problem in unknown environments for a robotic wheelchair is proposed. The wheelchair is commanded by a HMI that could be adapted to the patient's needs (a BCI or a Muscle-Computer Interface) until a door of the environment is detected. The aim of this work is to present a system that automatically steers the wheelchair with smooth movements through common doorways. These doorways are narrow considering the wheelchair's size, so the use of advanced techniques in the automatic system is required to reach the objectives. The vehicle is of unicycle type with two independent motors and a mini-PC onboard. It is also equipped with a range laser sensor to obtain measurements of the environment. When the patient chooses to cross a detected door, the problem is solved by a dynamic path planning algorithm implementation, based on successive frontier points determination. An adaptive trajectory tracking control based on the dynamic model is implemented on the vehicle. These kinds of controls are designed for mobile vehicles that transport heavy loads (the case of a wheelchair) and/or execute high velocity tasks. The controller drives the wheelchair motion along the path in a smooth movement. An EKF feature-based SLAM is also implemented on the vehicle. The SLAM gives an estimate of the wheelchair pose -position and orientation- inside the environment with minimized errors, which is then used by the trajectory controller. The SLAM allows the map reconstruction of the environment for future safe navigation purposes. The features extracted correspond to lines and corners -concave and convex- of the environment.

This paper is organized as follows. Section 2 shows the general architecture of the system, explaining the meaning and functionality of each part of the system: doors detection algorithm, dynamic path planning, EKF-based SLAM and the trajectory tracking controller; section 3 shows the implementation results of the entire system and section 4 contains the conclusions of the work.

2. General System Architecture

Figure 1 shows the architecture of the system proposed in this paper. This system works as follows. The robotic wheelchair navigates inside the environment commanded only by the user through the HMI. Once a doorway of the environment is detected, the vehicle stops its motion and displays a window to the user, showing him/her the option of crossing the detected door. If the user does not accept this option, he/she continues commanding the wheelchair's motion. On the other hand, if the user accepts the crossing-a-door option, a free-obstacle path is generated between the doorway and the wheelchair's location. The path is dynamically maintained and is based on a variation of the local frontier points method (Tao, T. et al., 2007). A trajectory controller is activated in order to ensure a smooth and time-constrained movement of the vehicle through the environment until it reaches the doorway. The vehicle's pose information used in the controller is generated by a SLAM algorithm. The variables of the SLAM system state are expressed in a fixed coordinate system attached to the floor at the vehicle's initial position in the cross-a-door procedure. The SLAM algorithm is a sequential EKF feature-based SLAM (Thrun, S. et al., 2005). This algorithm extracts corners -concave and convex- and lines of the environment to estimate the wheelchair position and orientation. The control commands and the doorway location are also introduced into the SLAM algorithm. The SLAM system state considers the doorway as a special feature of the environment. The trajectory tracking controller is a switching adaptive controller that takes into account the dynamics of the wheelchair. The doorway detection algorithm used in this paper is able to

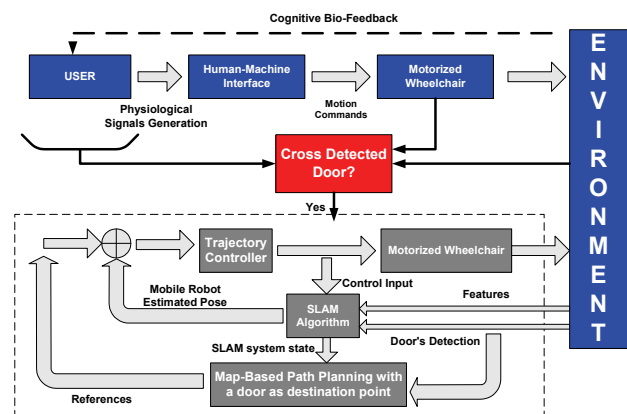


Fig. 1. General architecture of the proposed system.

recognize three different doorways disposition inside the environment. Once the doorway is reached by the vehicle, the motion control returns to the HMI.

Each block's functionality of Fig. 1 will be explained in the following sections.

2.1. Robotic Wheelchair

The autonomous wheelchair used in this work has the kinematics of an unicycle type vehicle with two independent motors -one for each wheel-.

Although odometric measurements were not used in this work, each wheel is equipped with an encoder. The wheelchair also has a range laser SICK®, which takes 181 measurements of the environment in a range of 180 degrees. A mini-pc is also incorporated to the vehicle. Figure 2 shows the schematic of the wheelchair.

In Fig. 2, the point h is the point of interest with coordinates x and y ; the variables u and ω are the linear and angular velocity respectively; x and y represent the location of the vehicle in the fixed coordinate system; ψ is the vehicle's orientation. The kinematic equations of the autonomous wheelchair are summarized in (1) - continuous case- and (2) -discrete case-.

$$\dot{\xi}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\Psi}(t) \end{pmatrix} = \begin{pmatrix} \cos(\Psi(t)) & -a \sin(\Psi(t)) \\ \sin(\Psi(t)) & a \cos(\Psi(t)) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u(t) \\ \omega(t) \end{pmatrix} + \Phi(t) \quad (1)$$

$$\begin{aligned} \xi(k) &= \begin{pmatrix} x(k) \\ y(k) \\ \Psi(k) \end{pmatrix} = \begin{pmatrix} x(k-1) \\ y(k-1) \\ \Psi(k-1) \end{pmatrix} \\ &+ \Delta t \begin{pmatrix} \cos(\Psi(k-1)) & -a \sin(\Psi(k-1)) \\ \sin(\Psi(k-1)) & a \cos(\Psi(k-1)) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u(k) \\ \omega(k) \end{pmatrix} + \Phi(k) \end{aligned} \quad (2)$$

In (1) and (2), Φ is the Gaussian noise associated to the kinematics model of the vehicle; Δt is the system sampled time. In (Bastos Filho, T. F. et al., 2007) there is a complete reference of the robotic wheelchair used in this work.

2.2. Human-Machine Interface

Although the System Architecture shown in Fig. 1 is not restricted to a specific HMI -the HMI specification depends on the patient's capabilities-, a Brain-Computer Interface (BCI) was used in this work.

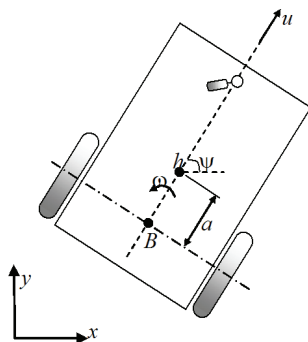


Fig. 2. Schematic of the robotic wheelchair.

This BCI is based on the ERS (event related synchronization) and ERD (event related desynchronization) signals obtained from the occipital lobe of the user. These ERS/ERD events are related to the relaxation and concentration mental states of the visual cortex (Ferreira, A. et al., 2008). When an ERS is presented, the electromyographic (EEG) signal increases up to 50 times its energy in the frequency range of 11 - 13 Hz (relaxation mental state) with respect to an ERD (concentration mental state). This particular event will be used for generating the motion commands. Two non-invasive electrodes are placed on the O1 and O2 positions on the scalp of the patient -see Fig. 3- and the EEG signal acquired is sent to the BCI for its processing and classification. Once the ERS or ERD event is detected and classified, it becomes the input alphabet of an Asynchronous Finite State Machine (AFSM). This AFSM translate the ERS and ERD events to motion commands. Considering that the ERS and ERD signals generation responds to the user's intentions, the output alphabet of the AFSM contains the motion commands of the robotic wheelchair. The six motion commands sent by the HMI to the vehicle are: start motion, turn to the left, turn to the right, move forward, move backward and stop motion. Both, linear and angular motions are executed by a PID-controller which attains velocities appropriate to the user. The BCI also has a low behavioral control law to avoid collisions of the vehicle. The entire BCI used in this work can be found in detail in (Ferreira, A. et al., 2008).

If a door is detected during the navigation commanded by the user's EEG signals, a display with a question - concerning the crossing the door action- shows up to the user in the visualization panel of the robotic wheelchair. The robot stops its motion and waits for the user response whether he/she accepts or rejects crossing the detected door. This situation is shown in Fig. 4.

If the user accepts to cross the detected door, then the procedures shown in the following sections are carried out; otherwise the motion control returns to the BCI.

In the visualization panel shown in Fig. 4, the option selection is also governed by the BCI. In this specific case, if an ERS is presented during the first 15 seconds after the door's detection, means that the user accepts to cross the door, otherwise he/she rejects that action.

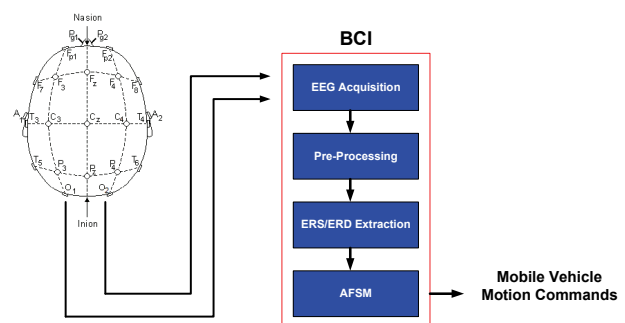


Fig. 3. The BCI Architecture. The EEG signal is acquired from the occipital lobe of the user, then it is filtered and scaled to an appropriate signal value; finally, the ERS/ERD events are extracted and sent to the Asynchronous Finite State Machine which converts them in motion commands.

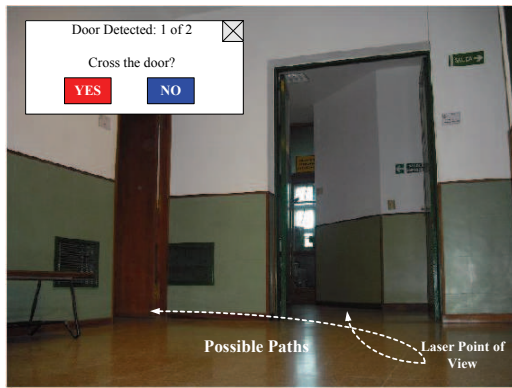


Fig. 4. Visualization of a door detection procedure. If more than one door is detected, the system first asks for the rightest door. If the user rejects it, the system asks for the second rightest door and continues so far until a door -or none- is selected.

2.3. Doorway Detection Procedure

The doorways detection algorithm implemented in this work is based on an adaptive clustering algorithm which uses the information contained in the laser histogram measurements to detect the doorways of the environment (Fukunaga, K., 1990). Only an open doorway is considered. When a possible doorway is recognized, the features -lines and corners- surrounding the doorway are analyzed. Thus, if a doorway and their surrounding features match with one of the three cases shown in Fig. 5, then the doorway is recognized as such. The three cases of doorways, classified according to their disposition inside the environment that the robot is able to detect, are shown in Fig. 5.

Once an open doorway is detected, it is represented by its middle point as it is shown in Fig. 5. The middle point of the doorway is represented in the fixed coordinate system, and has its covariance matrix attached to it (Guivant, J. E. & Nebot, E. M., 2001).

2.4. Path Planning Algorithm

Once an open doorway is detected in the environment, a feasible path is generated from the vehicle's position to the middle point of the doorway allowing the autonomous wheelchair to cross it. The path planning algorithm implemented in this work is based on a variation of the frontier points method (Tao, T. et al., 2007). This method finds empty spaces at the limits of the range sensor measurements and drives the motion of the mobile robot to these zones. The algorithm works as follows.

- Let δ be the distance between the vehicle and the middle point of the doorway.
- Let the distance between nodes of the path be selected as Δd . Let the middle point of the doorway be the last node of the path, and the vehicle's position (position of h in Fig. 2) the first node.
- Then, let suppose that the laser range is of $\delta - \Delta d$. The next node of the path is obtained by an angle windowing search of the frontier point associated to that laser range.

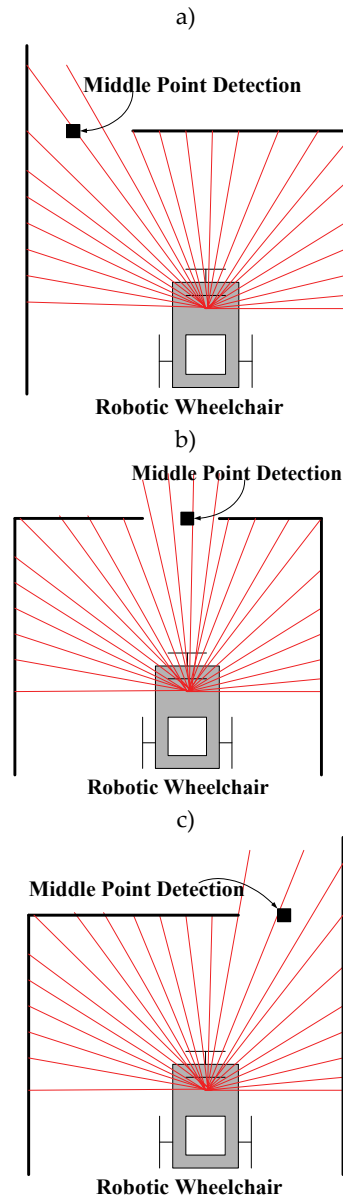


Fig. 5. Doorway's detection. a) Shows a doorway to the left of the vehicle; b) the doorway is in the middle of a wall; c) the doorway is to the right of the autonomous wheelchair. In all cases, the detected doorway is represented by its middle point.

- Now consider that the range of the laser is of $\delta - 2\Delta d$. This procedure continues until a node is close to the vehicle. In this paper, the last node obtained by the frontier points method is located at a distance of 0.5 m to the wheelchair. The distance of 0.5 m was established considering that the larger size of the vehicle is 0.6 m.
- Once the node generation is completed, they are joined by a sp-Line (Gentle, J. E. et al., 2004) in order to obtain a kinematically plausible path for the unicycle vehicle navigation (Laumond, J. P., 1998). A path that does not belong to C^2 means that it is not a kinematics plausible path for an unicycle vehicle navigation (Siegwart, R. & Nourbakhsh, I. R., 2004).

- The path is also dynamically maintained. Once the vehicle reaches the closest node to it, the entire path is reformulated. This situation is useful for environments with moving agents. The path generation is built in real time; the vehicle does not stop its motion.
- All nodes are determined in a local reference frame attached to the vehicle.

The general algorithm mentioned before is presented in Algorithm 1.

```

Doorway Detection                                (i)
if (Doorway Detection = TRUE)                    (ii)
  for distance =  $\delta$ : $\Delta d$ :0.5                      (iii)
    WSD = Window Size Determination              (iv)
    for window = 1:WSD:181                      (v)
      Pclose = closest meanfrontier point to      (vi)
        previous node
    end for                                      (vii)
    Path = [Path                               (viii)
      Pclose]
  end for                                      (ix)
end if                                         (x)

```

Algorithm 1. General structure of the Path Planning Method

The Algorithm 1 describes the process of finding a path through the environment to reach the doorway. If a doorway is detected (sentence (i)) then sentences from (ii) to (x) are executed. Thus, from a range of δ to 0.5 m the vehicle searches for possible nodes with a minimum distance of Δd between them (sentences (iii) to (ix)). In this work, $\Delta d = 0.2$ m was adopted. Also, the representing frontier point from a set of consecutive frontier points will be the mean point, as it is shown in Fig. 6.a. Sentences (v) to (vii) in Algorithm 1 show the angle windowing procedure. This procedure is necessary to avoid situations like the one shown in Fig. 6.b, where the mean frontier point could carry to a non navigable path. Thus, considering that the laser sensor can take 181 measurements from 0 - 180 degrees then, the current range length (sentence (iii)) will determine the size of the window where a mean frontier point will be seek. As it is shown in Fig. 6.c, as smaller is the range length, more angles will be taken into account in the windowing.

In this work, the function that manages the size of the window is a linear function. This function ensures that the number of angles of the laser involved in the frontier point detection will cover a neighborhood equal to the width of the wheelchair -see Fig. 6.c-. Thus, for example, if the range of the laser is set to 1 m, i.e., the frontier points remain at a distance of 1 m from the wheelchair, then approximately 30 degrees will be needed to cover the width of the wheelchair at that distance. The angle windowing starts at 0 degree and searches for a mean frontier point between 0 and 30 degrees. Then, the window moves one degree and searches in the interval of 1 to 31 degrees. This procedure is repeated until the 180 degrees of the laser are covered. From all the mean frontier points obtained at this instance, the closest mean

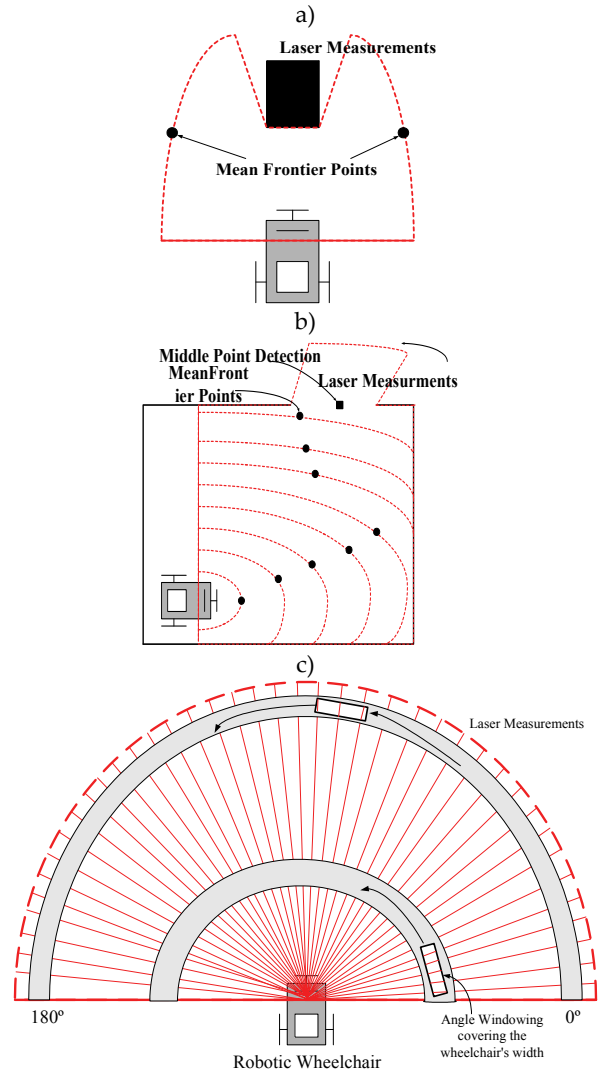


Fig. 6. Considerations about the Algorithm 1; a) Two mean frontier points are found. Each of them belongs to a set of consecutive frontier points. b) Nodes generation without angle windowing considerations. The path generated could be non navigable. c) Two examples of the angle windowing. When the range of the laser is close to the wheelchair, the window covers a higher number of degrees than when the range is longer.

frontier point to the last node of the path will be chosen as the current node.

Figure 7.a-b shows an example situation of the path planning whereas Fig. 7.c shows a final path with the secure navigable zone given by the angle windowing procedure.

The fact that the path generation is a dynamic procedure which is updated every time the vehicle reaches a node allows moving-obstacle avoidance -e.g. a person temporarily blocking the doorway-. This is due to the fact that, as will be mentioned in the following section, the position of the door remains stored in the SLAM system state. According to this, the vehicle does not lose its navigation's objective. Nevertheless it is worth mentioning that this avoidance procedure is restricted to obstacles that are not totally blocking the doorway.

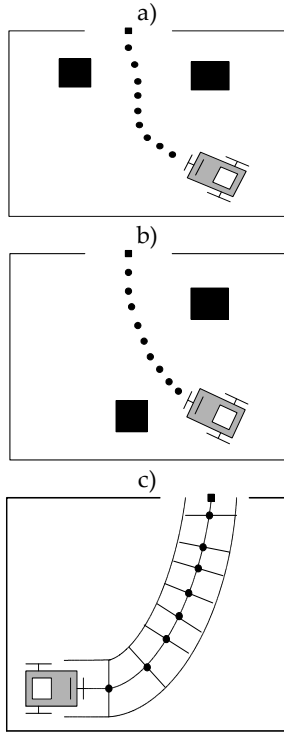


Fig. 7. Two examples of nodes generation with different environment disposition (a and b); c) final path generation. The result is a smooth and kinematics constrained path.

2.5. SLAM Algorithm

The SLAM algorithm implemented in this work is a sequential EKF feature-based SLAM (Thrun, S. et al., 2005). The features extracted from the environment correspond to lines and corners -concave and convex-. The system state is composed by the vehicle estimated pose, the doorway's position and the features of the environment. For visualization and map reconstruction purposes, a secondary map is maintained.

This secondary map stores the beginning and ending points of the segments associated with the lines of the environment. Thus, the secondary map allows finite wall's representation. The secondary map is updated according to the feature correction in the SLAM system state, and if a new feature is added to that system state, it is also added to the secondary map (Auat Cheein, F. A. et al., 2008). Equation (3) shows the system state structure and its covariance matrix. All elements of the SLAM system state are referenced to the fixed coordinate system.

$$\hat{\xi}(k|k) = \begin{bmatrix} \hat{\xi}_v(k|k) \\ \hat{\xi}_{door}(k|k) \\ \hat{\xi}_m(k|k) \end{bmatrix} \quad (3)$$

$$P(k|k) = \begin{bmatrix} P_{vv}(k|k) & P_{door,v}(k|k)^T & P_{m,v}(k|k)^T \\ P_{door,v}(k|k) & P_{door,door}(k|k) & P_{m,door}(k|k)^T \\ P_{m,v}(k|k) & P_{m,door}(k|k) & P_{m,m}(k|k) \end{bmatrix}$$

In (3), $\hat{\xi}(k|k)$ is the system state estimate; $\hat{\xi}_v(k|k) = [\hat{\xi}_{v,x}(k|k) \ \hat{\xi}_{v,y}(k|k) \ \hat{\xi}_{v,\psi}(k|k)]^T$ is the estimated pose of the vehicle; $\hat{\xi}_{door}(k|k)$ represents the Cartesian coordinates of the doorway's middle point in the fixed reference frame; $\hat{\xi}_m(k|k)$ represents the map of the environment. $\hat{\xi}_m(k|k)$ is composed by the Cartesian coordinates that define a corner and the polar coordinates that define a line in the environment. The order in which lines and corners appear in $\hat{\xi}_m(k|k)$ depends on the moment they were detected. $P(k|k)$ is the covariance matrix associated to the SLAM system state; $P_{vv}(k|k)$ is the covariance of the vehicle pose; $P_{door,door}(k|k)$ is the covariance associated to the position of the doorway's middle point, and $P_{mm}(k|k)$ is the covariance of the features of the environment. The rest of the elements of $P(k|k)$ are the cross-correlation matrices.

The covariance matrix initialization techniques and the EKF definition can be found in (Durrant-Whyte, H. & Bailey, T., 2006; Durrant-Whyte, H. & Bailey, T., 2006a). A sequential EKF (Thrun, S. et al., 2005) was implemented in order to reduce computational costs.

As was stated before, corners are defined in the Cartesian system whereas lines are in the polar system. Equations (4) and (5) show the features' models from a local reference frame attached to the mobile robot.

In (4) w_R and w_β represent the gaussian noises associated to the corner model; x_{corner} and y_{corner} are the Cartesian coordinates of the corner. In (5), w_ρ and w_α are the gaussian noises associated with the line parameters. The method used for line and corner extractions corresponds to adaptive clustering algorithms and can be found in (Fukunaga, K., 1990; Garulli, A. et al., 2005). Figure 8 shows the meaning of each variable presented in (4) - (5).

$$z_{corner}(k) = h_i[\xi_v(k), p_i(k), w(k)] = \begin{bmatrix} z_R \\ z_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{(x(k) - x_{corner})^2 + (y(k) - y_{corner})^2} \\ \text{atan}\left(\frac{y(k) - y_{corner}}{x(k) - x_{corner}}\right) - \Psi(k) \end{bmatrix} + \begin{bmatrix} w_R \\ w_\beta \end{bmatrix} \quad (4)$$

$$z_{line}(k) = h_i[\xi_v(k), p_i(k), w(k)] = \begin{bmatrix} z_\rho \\ z_\alpha \end{bmatrix} = \begin{bmatrix} \rho - x(k)\cos(\alpha) - y(k)\sin(\alpha) \\ \alpha - \Psi(k) \end{bmatrix} + \begin{bmatrix} w_\rho \\ w_\alpha \end{bmatrix} \quad (5)$$

Remark. One important use of the map stored by the SLAM is in the obstacle avoidance procedure. When a moving agent partially blocks the door, only a partial information of the door is obtained by the sensor laser. The rest of the information of the door is obtained using the map.

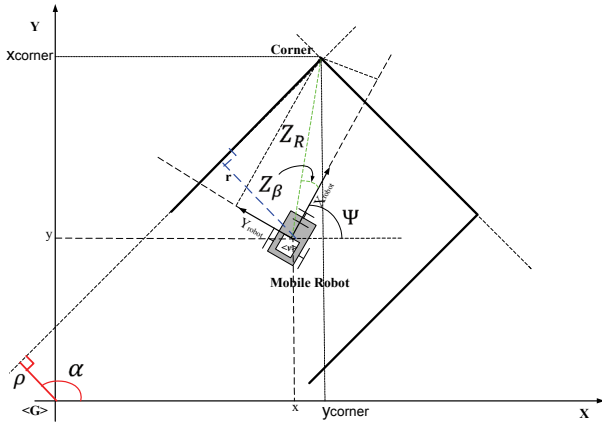


Fig. 8. Feature extraction from the environment. The corners are represented by their Cartesian coordinates whereas a line by their polar coordinates.

2.6. Adaptive Trajectory Tracking Controller

It is important to consider the vehicle's dynamics in addition to its kinematics because wheelchairs carry heavy loads. One characteristic of the control law based on the dynamic model is that the control signals are related to accelerations and not just to velocities as in the control law based on the kinematic model. Therefore, the resulting movements are more smooth using the first control law.

There are many proposed control laws based on the dynamic model in the mobile robot literature that can be used to control a robotic wheelchair. One of the approaches used to design the control laws is that based on feedback linearization (De La Cruz, C. & Carelli, R., 2008). This control approach has good performance; however, it depends on model parameters. When model parameters are unknown, an adaptive control for adjusting these parameters is required (Barzamini, R. et al., 2006), (De La Cruz, C. et al., 2008).

The adaptive tracking control of the robot system is based on (De La Cruz, C. et al., 2008). In that work a switching adaptive trajectory tracking controller based on the dynamic model is proposed. This controller is briefly described in the following subsections.

2.6.1. Dynamic model

The mobile robot is illustrated in Fig. 2, where $h = [x \ y]^T$ is the point that is required to track a trajectory, u and ω are the linear and angular velocities, ψ is the heading of the robot, and a is a distance from the center of the axle to the point of control of the robotic wheelchair.

Let us consider the following dynamic model of the mobile robot (De La Cruz, C. & Carelli, R., 2008):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u \cos \psi - a \omega \sin \psi \\ u \sin \psi + a \omega \cos \psi \\ \omega \\ \frac{\theta_3^0}{\theta_1^0} \omega^2 - \frac{\theta_4^0}{\theta_1^0} u \\ -\frac{\theta_5^0}{\theta_2^0} u \omega - \frac{\theta_6^0}{\theta_2^0} \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\theta_1^0} & 0 \\ 0 & \frac{1}{\theta_2^0} \end{bmatrix} \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix}$$

where θ_i^0 is the i -th model parameter that is a function of mass, moment of inertia, motor parameters and parameters of the low level servo control, and u_{ref} and ω_{ref} are the linear and angular reference velocities. Generally, these reference velocities are common input signals in commercial robots. Therefore, to keep compatibility with other autonomous vehicles, it is useful to express the robotic wheelchair model in a suitable way by considering linear and angular reference velocities as control signals.

2.6.2. Tracking control

The tracking control is (De La Cruz, C. et al., 2008)

$$v_{ref} = \hat{D}M(v - N) + T_a \hat{\theta}$$

$$v = \ddot{h}_d + K_1 \dot{\tilde{h}} + K_2 \tilde{h}, \quad \tilde{h} = h_d - h,$$

where:

$$v_{ref} = \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix}, \quad \hat{D} = \begin{bmatrix} \hat{\theta}_1 & 0 \\ 0 & \hat{\theta}_2 \end{bmatrix}, \quad M = \begin{bmatrix} \cos \psi & \sin \psi \\ -\frac{1}{a} \sin \psi & \frac{1}{a} \cos \psi \end{bmatrix},$$

$$N = \begin{bmatrix} -u \omega \sin \psi - a \omega^2 \cos \psi \\ u \omega \cos \psi - a \omega^2 \sin \psi \end{bmatrix}, \quad h = \begin{bmatrix} x \\ y \end{bmatrix},$$

$$T_a = \begin{bmatrix} 0 & 0 & -\omega^2 & u & 0 & 0 \\ 0 & 0 & 0 & 0 & u \omega & \omega \end{bmatrix}, \quad \hat{\theta} = [\hat{\theta}_1 \ \hat{\theta}_2 \ \dots \ \hat{\theta}_6]^T,$$

K_1 and K_2 are 2×2 definite positive diagonal matrices, $h_d(t)$ defines the desired trajectory, $\hat{\theta}_i$ is the i -th estimated parameter.

2.6.3. Adaptive law

The adaptive law is (De La Cruz, C. et al., 2008)

$$\dot{\hat{\theta}} = K_A^{-1} Y^T P e_T$$

where:

$$Y = \begin{bmatrix} 0 \\ M^{-1} \hat{D}^{-1} T \end{bmatrix}$$

$$T = \begin{bmatrix} T_{11} & 0 & -\omega^2 & u & 0 & 0 \\ 0 & T_{22} & 0 & 0 & u \omega & \omega \end{bmatrix}$$

$$\begin{bmatrix} T_{11} \\ T_{22} \end{bmatrix} = M(\ddot{h} - N), \quad e_T = \begin{bmatrix} \tilde{h} \\ \dot{\tilde{h}} \end{bmatrix}$$

K_A is a 6×6 definite positive diagonal matrix. The matrix P is defined as follows: $P = P^T > 0$ such that

$$A_K^T P + P A_K = -Q; \quad Q = Q^T > 0$$

where

$$A_K = \begin{bmatrix} 0 & I \\ -K_2 & -K_1 \end{bmatrix},$$

and I is an identity matrix.

2.6.4. Projection algorithm

Some values of estimated parameters need to be avoided. For example, in the adaptive law it is required that $\hat{\theta}_1 = 0$ and $\hat{\theta}_2 = 0$ be avoided to allow calculation of \hat{D}^{-1} . A projection algorithm can be used to reach this objective. The projection algorithm used in this work is (De La Cruz, C. et al., 2008)

$$\hat{\theta}_i = l_i \text{ if } \hat{\theta}_i \leq l_i - \zeta_i$$

where l_i is the minimum possible value of θ_i^0 ; $\zeta_i > 0$; and $l_i - \zeta_i > 0$.

2.6.5. Switching control

The switching adaptive scheme is as follows (De La Cruz, C. et al., 2008):

$$\dot{\hat{\theta}} = \begin{cases} K_{A1}^{-1} Y^T P e_T & \text{if } Con = 1 \\ K_{A2}^{-1} Y^T P e_T & \text{if } Con = 2 \\ 0 & \text{if } Con = 3 \end{cases}$$

The variable *Con* becomes 1 when a new desired trajectory starts. The variable *Con* switches from 1 to 2 when $e_T \in S_A$ the first time in a desired trajectory. The set S_A is a set of non high control errors. The variable *Con* switches from 2 to 3 when $V_{eT} \leq C_{V3}$. Where $V_{eT} = e_T^T P e_T$ and C_{V3} is a positive constant. The variable *Con* switches from 3 to 2 when $V_{eT} > C_{V3}$.

The gain K_{A1}^{-1} is less than the gain K_{A2}^{-1} . This is considered, in the switching scheme, to reduce high accelerations at the beginning of the trajectory because of high initial control error. Through simulations and experiments it was observed that high gain K_A^{-1} and high control error at the beginning of a trajectory leads to high accelerations of the mobile robot. In some applications, such as wheelchair control, it is very important to avoid high accelerations.

The parameter updating law works as an integrator and, therefore, can cause robustness problems in case of measurement errors, noise or disturbances. One possible way for preventing parameter drifting is by turning off the parameter updating when the control error is smaller than a boundary value. This is done in the switching scheme.

3. Implementation Results

In this section, the implementation results of the entire system are shown. The experiments were carried out at the Electrical Department of the Federal University of Espirito Santo, Brazil. The robotic wheelchair, is equipped with a range laser SICK®. The maximum range adopted was of 10 m; the sampled time of the system was of 0.1 seconds; the models of the vehicle and the features are the ones shown in (2), (4) and (5). The point of control h -see Fig. 2- is located at 0.7 m from the center of the wheels axle. All implementations, integrating doors detection, dynamic path planning, SLAM and trajectory control, were real time realizations.

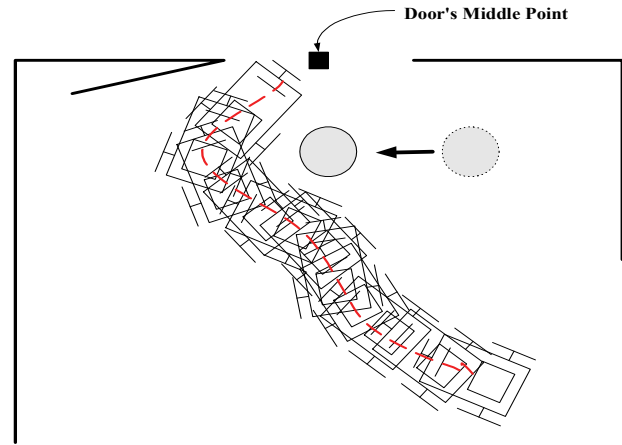


Fig. 9. Obstacle avoidance during the navigation. The path travelled by the vehicle is drawn in red dashed line.

3.1. Path Planning and Obstacle Avoidance Results

The dynamic path planning based on successive frontier points -as presented in the previous section- allows obstacle avoidance when, for example, a moving agent partially blocks the door. Figure 9 shows this situation.

As it is shown in Fig. 9, the dynamic path planning gives certain autonomy to the vehicle navigation allowing to avoid moving obstacles. Considering that the door is treated like any other feature of the environment and its coordinates remain in the SLAM system state, the target is not lost and the path can be changed.

3.2. SLAM Results

Figure 10 shows two representative cases of the SLAM algorithm applied to cross-a-door problem in this paper. As it is shown in Fig. 10, the number of features of the environment is relatively low. As a consequence, the processing time of the system is not compromised. The vehicle shows a smooth and stable navigation. It stops its motion once the door is reached.

In Fig. 10, raw laser measurement are drawn with yellow points; lines -representing walls of the environment- are represented by solid black segments; the beginning and ending point of each segment -extracted from the secondary map- are represented by red crosses and the corners of the environment are green circles. The path travelled by the vehicle is drawn with solid red line. As it is shown in Fig. 10, the SLAM system state is consistent with the control law implemented, directing the vehicle's movements to the detected door of the environment. Up to this point, the implementation of an SLAM algorithm to provide the vehicle pose to the controller could be replaced by an EKF feature-based localization algorithm (Siegwart, R. & Nourbakhsh, I. R., 2004) but in this case, the map obtained will not be useful for further navigation purposes due to non-minimized errors of the features.

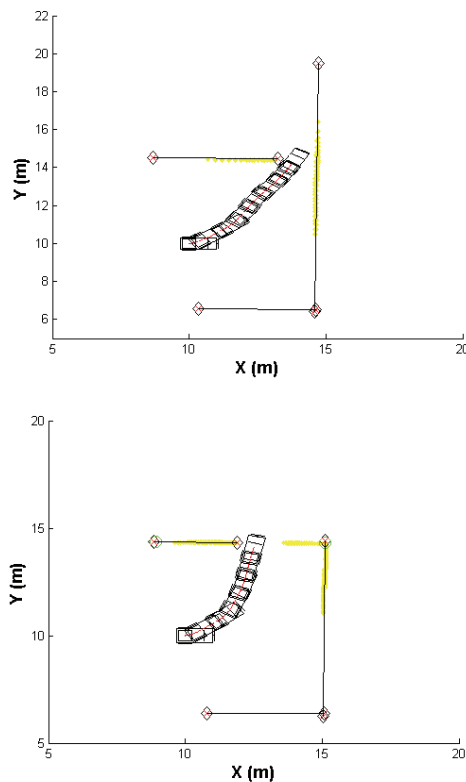


Fig. 10. SLAM results for the wheelchair navigation. In both cases, the robot successfully reaches the door. The width of the doors was of 0.80 m.

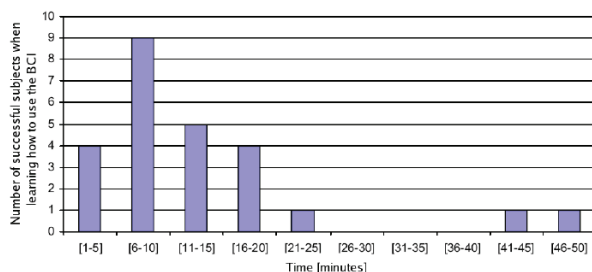


Fig. 11. BCI learning time statistics in a population of 25 cognitive normal people. The average time was of 15 minutes.

3.3. BCI Results

The BCI used in this work was tested in a population of 25 cognitive normal people. They have shown an average time of 15 minutes in learning how to command de BCI generating the appropriate ERS/ERD signals, and to drive the robot. Figure 11 shows the statistical result of the average learning time experiments.

4. Conclusions

An efficient solution to the cross-a-door problem for a robotic wheelchair governed by a Human-Machine Interface was proposed. A dynamic path planning algorithm based on successive frontier points determination, an adaptive trajectory tracking control based on the dynamic model and an EKF feature-based

SLAM are used in the entire system. The SLAM allows the map reconstruction of the environment for future safe navigation purposes -e.g. when an obstacle temporarily blocks the doorway- and provides vehicle's pose information for the trajectory controller without using odometric data. The entire system was evaluated in real time realizations. Common situations as obstacle avoidance and different kind of doors were considered in the experiments.

In future works, a passageway navigation will be integrated with this system and analyze the navigation from one room to another.

5. Acknowledgment

The authors would like to thanks to the National Council for Scientific and Technical Research (CONICET) for partially founding this project. This work was also supported by FAPES-Brazil, project: Adaptive Dynamic Controller Applied to a Robotic Wheelchair Commanded by Brain Signals, process 39385183/2007.

6. References

- Anguelov, D.; Koller, D.; Parker, E. & Thrun, S. (2004). Detecting and Modeling Doors with Mobile Robots, IEEE International Conference on Robotics and Automation, pp. 3777- 3784, ISBN: 0-7803-8232-3, Barcelona, Spain, April 2004, IEEE.
- Auat Cheein, F. A.; di Sciascio, F. & Carelli, R. (2008). Planificación de Caminos y Navegación de un Robot Móvil en Entornos Gaussianos mientras realiza tareas de SLAM (in spanish, Path Planning and Mobile Robot Navigation in Gaussian Environments while executes a SLAM algorithm), Jornadas Argentinas de Robotica, August 2008, Bahia Blanca, Buenos Aires, Argentina.
- Barzamini, R.; Yazdizadeh, A.R. & Rahmani, A.H., (2006). A new adaptive tracking control for wheeled mobile robot. in IEEE Conference on Robotics, Automation and Mechatronics, pp. 1-6.
- Bastos Filho, T. F.; Ferreira, A.; Silva, R.; Celeste, W. C. & Sarcinelli, M., (2007). Human-Machine Interface Based on EMG and EEG Signals Applied to a Robotic Wheelchair, *Journal of Physics Conference Series*, 1:1/012088-10. ISSN: 1742-6596.
- Biswas, R.; Limketkai, B.; Sanner, S. & Thrun, S. (2002). Towards object mapping in dynamic environments with mobile robots, Proceedings of the International Conference on Intelligent Robots and Systems, ISBN: 0-7803-7399-5, Switzerland, IEEE.
- Bourhis, G.; Horn, O.; Habert, O. & Pruski, A. (2001). An autonomous vehicle for people with motor disabilities, *IEEE Robotics and Automation Magazine*, Vol. 8, No. 1,(March 2001), pp. 20-28, ISSN: 1070-9932.
- Chatila, R. & Laumond, J. P. (1985), Position referencing and consistent world modeling for mobile robots, Proceedings of IEEE International Conference on Robotics and Automation, pp. 138-145, St. Louis, USA.

- De La Cruz, C.; Carelli, R. & Bastos, T. F. (2008). Switching adaptive control of mobile robots, *IEEE International Symposium on Industrial Electronics – ISIE08*, pp. 835-840, ISBN: 978-1-4244-1665-3, July 2008, IEEE, Italy.
- De La Cruz, C. & Carelli, R., (2008). Dynamic model based formation control and obstacle avoidance of multi-robot systems. *Robotica*, Vol. 26, No. 3, (May 2008), pp. 345-356, ISSN: 0263-5747.
- Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H. F. & Csorba, M. (2001). A solution to the simultaneous localisation and map building (SLAM) problem, *IEEE Trans. Robotics and Automation*, Vol. 17, No. 3, (June 2001), pp. 229-241, ISSN: 1042-296X.
- Durrant-Whyte, H. & Bailey, T., (2006). Simultaneous localization and mapping (SLAM): part I essential algorithms, *IEEE Robotics and Automation Magazine*, Vol. 13, No. 2, (June 2006), pp. 99-108, ISSN: 1070-9932.
- Durrant-Whyte, H. & Bailey, T., (2006a). Simultaneous localization and mapping (SLAM): part II state of the art, *IEEE Robotics and Automation Magazine*, vol. 13, No. 3, (September 2006), pp. 108-117, ISSN: 1070-9932.
- Ferreira, A.; Celeste, W. C.; Auat Cheein, F.; Bastos, T. F.; Sarcinelli, M. & Carelli, R. (2008). Human Machine Interfaces based on EMG and EEG applied to Robotic Systems, *Journal of NeuroEngineering and Rehabilitation*, Vol. 5, No. 10, (March 2008), ISSN: 1743-0003.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Elsevier, ISBN: 978-0-12-269851-4.
- Galan, F.; Nuttin, M.; Lew, E.; Ferrez, P. W.; Vanacker, G.; Philips, J. & Millan, J. R., (2008). A brain-actuated wheelchair: asynchronous and non-invasive Brain-computer interfaces for continuous control of robots, *Clinical Neurophysiology*, Vol. 119, No. 9, (July 2008), pp. 2159-2169, ISSN: 1388-2457.
- Garulli, A.; Giannitrapani, A.; Rosi, A. & Vicino, A. (2005). Mobile robot SLAM for line-based environment representation, *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2041-2046, ISBN: 0-7803-9567-0, December 2005.
- Gentle, J. E.; Hardle, W. & Mori, Y. (2004). *Handbook of Computational Statistics*, Springer, ISBN: 9783540404644, Germany.
- Guivant, J. E. & Neboit, E. M., (2001). Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, (June 2001), pp. 242-257, ISSN: 1042-296X.
- Kouzoubov, K. & Austin, D. (2004). Hybrid Topological/Metric Approach to SLAM, *Proc. of the IEEE International Conference on Robotics and Automation*, Piscataway, N. J. (Ed.), pp. 872-877, ISBN: 0-7803-8232-3, IEEE, New Orleans, Los Angeles.
- Laumond, J. P. (1998). *Robot Motion, Planning and Control. Lecture Notes in Control and Information Science* 229, Springer, ISBN: 3-540-76219-1, Germany.
- Mazo, M., (2001). An integral system for assisted mobility, *IEEE Robotics and Automation Magazine*, Vol. 8, No.1, (March 2001), pp. 46-56, ISSN: 1070-9932.
- Millan, J. R.; Renkens, F.; Mourino, J. & Gerstner, W., (2004). Non-Invasive Brain-Actuated Control of a Mobile Robot by Human EEG, *IEEE Transactions on Biomedical Engineering*, Vol. 51, No. 6, (June 2004), pp. 1026-1033, ISSN: 0018-9294.
- Muñoz-Salinas, R.; Aguirre, E.; García-Silvente, M. & Gómez, M., (2005). A multi-agent system architecture for mobile robot navigation based on fuzzy and visual behaviour, *Robotica*, Vol. 23, pp. 689-699, , ISSN: 0263-5747.
- Parikh, S. P.; Grassi, V.; Kumar, V. & Okamoto, J., (2007). Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform, *IEEE Intelligent Systems*, Vol. 22, No. 2, (April 2007), pp. 33-41, ISSN: 1541-1672.
- Poncet, A.; Urdiales, C. & Sandoval, F., (2007). A CBR approach to Behaviour-Based Navigation for an Autonomous Mobile Robot, *IEEE International Conference on Robotics and Automation*, pp. 3681-3686, ISBN: 1-4244-0602-1, Rome, Italy.
- Siegmund, R. & Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*, The MIT Press, ISBN: 0-262-19502-X, Massachusetts.
- Tao, T.; Huang, Y.; Sun, F. & Wang, T. (2007). Motion Planning for SLAM Based on Frontier Exploration, *Proc. of the IEEE International Conference on Mechatronics and Automation*, pp. 2120-2125, ISBN: 978-1-4244-0828-3, August 2007.
- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, MIT Press, ISBN: 978-0-262-20162-9, Massachusetts.
- Xi, B.; Guo, R.; Sun, F. & Huang, Y., (2008). Simulation Research for Active Simultaneous Localization and Mapping Based on Extended Kalman Filter, *Proc. of the IEEE International Conference on Automation and Logistics*, pp. 2443-2448, ISBN: 978-1-4244-2502-0, September of 2008, IEEE.
- Zeng, Q.; Teo, C. L.; Rebsamen, B. & Burdet, E. (2008). A collaborative wheelchair system, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 16, No. 2, (April 2008), pp. 161-170, ISSN: 1534-4320.
- Wai, R. J. & Chen, P. C. (2006). "Robust Neural-Fuzzy-Network Control for Robot Manipulator Including Actuator Dynamics," *IEEE Transactions on Industrial Electronics*, Vol. 53, No. 4, pp. 1328-1349.
- Wang, D.; & Huang, J. (2002). "Adaptive Neural Network Control for a Class of Uncertain Nonlinear Systems in Pure-Feedback Form," *Automatica* 38, pp. 1365-1372.
- Zhang, T.; Ge, S. S. & Hang, C. C. (2000). "Adaptive Neural Network Control for Strict-Feedback Nonlinear Systems Using Backstepping Design," *Automatica* 36, pp. 1835-1846.
- Zhang, Y.; Peng, P. Y. & Jiang, Z. P. (2000). "Stable Neural Controller Design for Unknown Nonlinear Systems Using Backstepping," *IEEE Transactions on Neural Networks*, Vol. 11, No. 6, pp. 1347-1360.

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821