The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Solution to a Door Crossing Problem for an Autonomous Wheelchair

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

Abstract—**This paper proposes a solution to a door crossing problem in unknown environments for an autonomous wheelchair. 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 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 future safe navigation purposes. The entire system is evaluated in a real time simulator of a robotic wheelchair.**

I. INTRODUCTION

The integration of robotics issues into the medical field has become of a great interest in the recent years. Mechanical devices specially developed for surgery like robot manipulators, control algorithms for tele-operation of those robots and cognitive algorithms for user decision learning are some examples of robotic applications in medicine. On the other hand, service, assistance, rehabilitation and surgery are the more benefited human health-care areas by the recent advances in robotics. Precisely, 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 [20], [21], [22]. In autonomous mode, the robotic wheelchair goes to the chosen destination without any participation of the user in the control. This mode is intended for people who have difficulties to continuously give a control command to guide the wheelchair. In the semi-autonomous mode the user share the control with the robotic wheelchair. In this case only some motor skills are needed from the user.

One of the solutions to the autonomy problem of mobile vehicles provided by the robotics field is the implementation of a SLAM (Simultaneous Localization and Mapping) algorithm [1]. 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 [2]. Although this algorithm is processing time demanding, it becomes a powerful solution when the vehicle has to navigate unsensored or in unknown environments, obtaining a reliable map of it [3]. From its earlier beginning, the SLAM has been implemented in several algorithms [4, 5], being the EKF (Extended Kalman Filter) the most used by the scientific community [6, 7]. 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 [8] whereas the UKF has shown a better performance dealing with non-linear models of the vehicle and the measurements [8].

Despite of the fact that the map built by the SLAM could be of different types -topological, metric, hybrid [8]- the most used map is a metric feature-based map, which extracts some geometrical constrain from the environment -e.g., corners, lines, color patterns [9]-. These features are then used to localize the vehicle inside that environment.

In this paper, a door crossing problem in unknown environments is addressed. It is considered the case of crossing a doorway by an autonomous wheelchair. The vehicle is an 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 environments. The door crossing 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-.

Manuscript received March 1, 2009. This work was supported in part by the CONICET, Argentina and FAPES-Brazil (Process: 39385183/2007).

F. A. C., is with the Institute of Automatics, National University of San Juan, Argentina (e-mail: fauat@inaut.unsj.edu.ar).

C. D. is with the Electrical Engineering Department of Unversidade Federal do Espírito Santo, Brazil, Av. Fernando Ferrari 514, Vitória, Espírito Santo, Brazil (corresponding author; e-mail: celsodelacruz@gmail.com).

R. C. is with the Institute of Automatics, National University of San Juan, Argentina (e-mail: rcarelli@inaut.unsj.edu.ar).

T. B. is with Universidade Federal do Espírito Santo, Brazil (e-mail: teodiano@ele.ufes.br).

This paper is organized as follows. Section II shows the general architecture, explaining the meaning and functionality of each part of the system: doorways detection algorithm, dynamic path planning, EKF-based SLAM and the trajectory tracking controller; section III shows the simulation results of the entire system and section IV are the conclusions of the work.

II. General System Architecture

Figure 1 shows the architecture of the system proposed in this paper.

The system shown in Fig. 1 works as follows. Once a doorway of the environment is detected, 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 [10]. A trajectory controller is implemented 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 SLAM algorithm is a sequential EKF feature-based SLAM [8]. 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 recognize three different doorways disposition inside the environment. Although the general system shown in Fig. 1 does not search for a door, several approaches concerning this problem can be found in the literature [1, 11, 12].

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

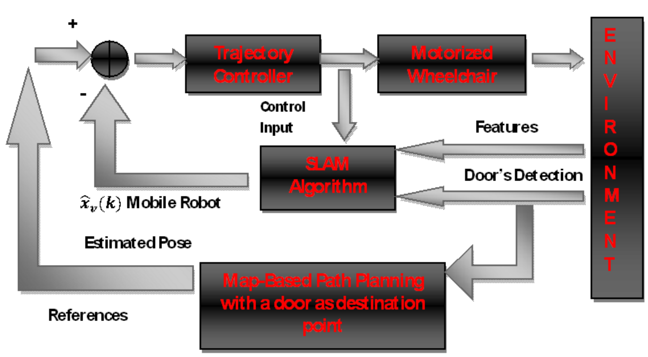

Fig. 1. General architecture of the proposed system.

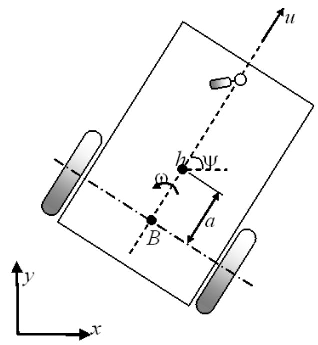

Fig. 2. Schematic of the wheelchair.

A. Autonomous Wheelchair

The autonomous wheelchair used in this work can be considered as an unicycle type vehicle with two independents motors -one for each wheel-[13]. Although odometric measurements were not used in this work, each wheel is equipped with a 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* represents the point at which all control actions will be considered. *u* and ω are the linear and angular velocity respectively. *x* and *y* represent the location of the vehicle in a global coordinate system; Ψ is the vehicle's orientation. The kinematics equations of the autonomous wheelchair are summarized in (1) -continuous case- and (2) -discrete case-.

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

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

In (1) and (2), Φ is the Gaussian noise associated to the kinematics model of the vehicle; Δt is the system sampled time.

B. Doorways 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 [14]. 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 below, then the doorway is recognized as such. The three cases of doorways according to their disposition inside the environment that the robot is able to detect are shown in Fig. 3.

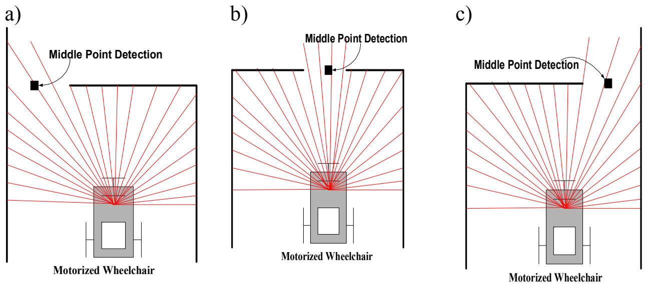

Fig. 3. Doorway's detection. a) Shows a doorway to the left of the vehicle; b) the doorway is in the middle of a wall; c) the doorway is to the right of the autonomous wheelchair. In all cases, the detected doorway is represented by its middle point.

Once an open doorway is detected, it is represented by its middle point as it shown in Fig. 3.

The middle point of the doorway is represented in the global coordinate system of the environment and has its covariance matrix attached to it [15].

C. 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 [10]. This method finds empty spaces at the limits of the range sensor measurements and drives the motion of the mobile robot to those zones. In this paper, a similar concept was implemented, although, in this case, the nodes of the path are found by frontier points analysis. The algorithm works as follows.

- Let δ be the distance between the vehicle and the middle point of the doorway.
- Let Δd be the distance between nodes of the path. 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 windowed search of the frontier point associated to that laser range.
- Now consider that the range of the laser is of $\delta - 2\Delta d$. This procedure continues until a node is near the vehicle. In this paper, the last node obtained by the frontier points method is located at distance of 0.5 m to the wheelchair. The distance of 0.5 m was established considering that the maximum radio of the vehicle is 0.6 m.
- Once the node generation is completed, they are joined by a sp-Line [16] in order to obtain a path that belongs to C^2 [17]. A path that does not belong to C^2 means that it is not a kinematics plausible path for an unicycle vehicle navigation [18].
- 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 made in real time; the vehicle does not stop its motion.
- All nodes are determined in a local reference frame attached to the vehicle and then converted to the global reference frame of the system.

The general algorithm mentioned before is presented in Algorithm 1.

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, Δd = 0.2 m was adopted. Considering that a frontier point is a point at the limit of the laser range, if two or more consecutive frontier points exist at a same time, a criterion of choice must be adopted. In this work, the chosen point from a set of consecutive frontier points will be the mean point, as it is shown in Fig. 4.a. Sentences (v) to (vii) in Algorithm 1 show the angle windowed procedure. This procedure is necessary to avoid situations like the one shown in Fig. 4.b, where the mean frontier point could carry to a non navigable path. Thus, considering that the laser sensor can take 181 measurements from 0º - 180º then, the actual range length (sentence (iii)) will determine the size of the window where a mean frontier point will be seek. Figure 4.c shows this situation.

Algorithm 1- General structure of the Path Planning Method

Doorway Detection	(i)
if (Doorway Detection = TRUE)	(ii)
for distance = δ:Δd:0.5	(iii)
WSD = Window Size Determination	(iv)
for window = 1:WSD:181	(v)
P_{close} = closest frontier point to previous node	(vi)
end for	(vii)
Path = [Path P_{close}]	(viii)
end for	(ix)
end if	(x)

As it is shown in Fig. 4.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 ensured that the number of angles of the laser involved in the frontier point detection will cover a neighborhood equal to the wide of the wheelchair -see Fig. 4.c-. Figure 5.a shows an example situation of the path planning whereas Fig. 5.b shows a final path with the secure navigable zone. Finally, if two or more windows give a possible node, the one closer to the previous node is chosen.

Finally, 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-.

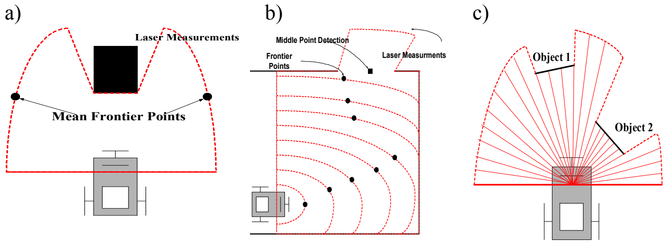

Fig. 4. Considerations about the Algorithm 1; a) Two mean frontier points are found. Each of them belongs to a set of consecutive frontier points. b) Nodes generation without angle windowed considerations. The path generated could be not navigable. c) Two objects of the same length and normal to the laser measurements have different size of windows. Thus, *object 1* needs four angles whereas *object 2* needs 7 measurements.

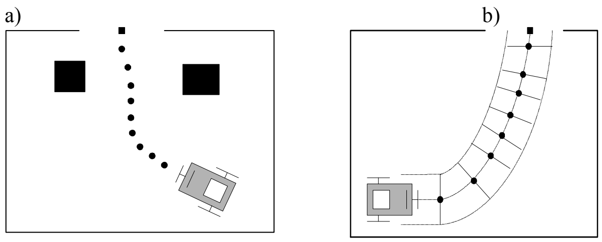

Fig. 5. An example of path generation; a) nodes generation example; b) final path generation. The result is a smooth and kinematics constrained path.

D. SLAM Algorithm

The SLAM algorithm implemented in this work is a sequential EKF feature-based SLAM [8]. 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 in the secondary map [18]. Equation (3) shows the system state structure and its covariance matrix. All elements of the SLAM system state are referenced to a global coordinate system. Once a doorway is detected, the origin of the global coordinate system is attached to the wheelchair actual pose.

$$\hat{x}(k|k) = \begin{bmatrix} \hat{x}_v(k|k) \\ \hat{x}_{door}(k|k) \\ \hat{x}_m(k|k) \end{bmatrix}$$
$$P(k|k) = \begin{bmatrix} P_{vv}(k|k) & P_{door,v}(k|k)^T & P_{m,v}(k|k)^T \\ P_{door,v}(k|k) & P_{door,door}(k|k) & P_{m,door}(k|k)^T \\ P_{m,v}(k|k) & P_{m,door}(k|k) & P_{m,m}(k|k) \end{bmatrix} \quad (3)$$

In (3), $\hat{x}(k|k)$ is the system state estimate; $\hat{x}_v(k|k) = [\hat{x}_{v,x}(k|k) \quad \hat{x}_{v,y}(k|k) \quad \hat{x}_{v,\Psi}(k|k)]^T$ is the estimated pose of the vehicle; $\hat{x}_{door}(k|k)$ represents the Cartesian coordinates of the doorway's middle point at the global reference frame; $\hat{x}_m(k|k)$ represents the map of the environment. $\hat{x}_m(k|k)$ 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 $\hat{x}_m(k|k)$ 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 [6, 7]. A sequential EKF [8] 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 [14, 19].

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

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

E. Adaptive Trajectory Tracking Controller

The adaptive tracking control of the robot system is obtained from [23]. In that work a switching adaptive trajectory tracking controller based on the dynamic model is proposed. This controller is described in the following subsections.

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.

Let us consider the following dynamic model of the mobile robot [24]:

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

where θ_i^0 is the i-th model parameter that is 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.

2. Tracking control

The tracking control is [23]

$$\nu_{ref} = \hat{D}M(\upsilon - N) + T_a\hat{\theta}, \quad \upsilon = \ddot{h}_d + K_1\dot{\tilde{h}} + K_2\tilde{h}, \quad \tilde{h} = h_d - h$$

where:

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

$$N = \begin{bmatrix} -u\omega\sin\psi - a\omega^2\cos\psi \\ u\omega\cos\psi - a\omega^2\sin\psi \end{bmatrix}, \ T_a = \begin{bmatrix} 0 & 0 & -\omega^2 & u & 0 & 0 \\ 0 & 0 & 0 & 0 & u\omega & \omega \end{bmatrix},$$

K_1 and K_2 are 2x2 definite positive diagonal matrices, $h_d(t)$ defines the desired trajectory, and $\hat{\theta}$ is the vector of estimated parameter.

3. Adaptive law

The adaptive law is [23]

$$\dot{\hat{\theta}} = K_A^{-1}Y^TPe_T$$

where:

$$Y = \begin{bmatrix} 0 \\ M^{-1}\hat{D}^{-1}T \end{bmatrix}, \ T = \begin{bmatrix} T_{11} & 0 & -\omega^2 & u & 0 & 0 \\ 0 & T_{22} & 0 & 0 & u\omega & \omega \end{bmatrix},$$

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

K_A is a 6x6 definite positive diagonal matrix. The matrix P is defined as follows: $P = P^T > 0$ such that

$$A_K{}^TP + PA_K = -Q; \quad Q = Q^T > 0$$

where

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

and I is an identity matrix.

4. Projection algorithm

Sometimes, it is required that a estimated parameter avoids some values. For example, in this adaptive law is required to avoid $\hat{\theta}_1 = 0$ and $\hat{\theta}_2 = 0$ to calculate $\hat{D}^{-1}$. A projection algorithm can be used to reach this objetive. The projection algorithm used in this work is [23]

$$\hat{\theta}_i = l_i \quad if \quad \hat{\theta}_i \le l_i - \zeta_i$$

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

5. Switching control

The switching adaptive scheme is as follows [23]:

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

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 $V_{eT} \le C_{V3}$, where $V_{eT} = e_T{}^TPe_T$ 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, were 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 to prevent 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.

III. Simulation Results

In this section, the simulation results of the entire system are shown. A real time simulator that considers the dynamics of the robotic wheelchair was used. The dynamic model used in the simulator was identified and validated using a real robotic wheelchair. The simulator was developed at the Electrical Engineering Department of the Federal University of Espirito Santo, Brazil. The autonomous wheelchair, in the simulator, is equipped with range laser SICK®. The maximum range adopted was of 10 m; the sampled time of the system was of 0.1 seconds; the models used 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 to 0.7 m from the center of the wheels axle. All simulations, integrating doorways detection, dynamic path planning, SLAM and trajectory control, were real time realizations.

A. 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 doorway. Figure 6 shows this situation.

As it is shown in Fig. 6, the dynamic path planning gives certain autonomy to the vehicle navigation allowing avoiding moving obstacles. Considering that the doorway 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.

B. *SLAM Results*

Figure 7 shows two representative cases of the SLAM algorithm applied to door crossing problem in this paper. As it is shown in Fig. 7, the number of features of the environment is relative low. This situation does not compromise the processing time of the system. The vehicle shows a smooth and stable navigation. It stops its motion once the doorway is reached.

In Fig. 7, lines -representing walls of the environment- are represented by solid lines; the beginning and ending point of each segment -extracted from the secondary map- are represented by crosses and the corners of the environment are circles. The path travelled by the vehicle is drawn with solid red line. As it is shown in Fig. 7, the SLAM system state is consistent with the control law implemented, driving the vehicle's movements to the detected doorway 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 [1] but in this case, the map obtained will not be useful for further navigation purposes due to non-minimized errors of the features.

IV. CONCLUSIONS

An efficient solution to the door crossing problem for a robotic wheelchair 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 were used in the entire system. The SLAM allowed the map reconstruction of the environment for future safe navigation purposes and provided vehicle's pose information for the trajectory controller without using odometric data. The entire system was evaluated in a real time simulator which considers the dynamics of the robotic wheelchair. Common situations as obstacle avoidance and different kind of doorways were considered in the simulations.

In future works, the passageway navigation will be integrated with this system and will be implemented the navigation between rooms.

ACKNOWLEDGMENT

The authors would like to thanks to the CONICET-Argentina for partially founding this project. This work was also supported by FAPES-Brazil, process 39385183/2007.

REFERENCES

[1] R. Siegwart, I. R. Nourbakhsh. "Introduction to Autonomous Mobile Robots". The MIT Press. 2004.

[2] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. "A solution to the simultaneous localisation and map building (SLAM) problem." *IEEE Trans. Robotics and Automation, 2001.*

[3] K. Kouzoubov, D. Austin. "Hybrid Topological/Metric Approach to SLAM". Proc. of the IEEE International Conference on Robotics and Automation. 2004.

[4] R. Chatila and J.P. Laumond, "Position referencing and consistent world modeling for mobile robots," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 138-145, St. Louis, USA, 1985.

[5] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. "A solution to the simultaneous localisation and map building (SLAM) problem." *IEEE Trans. Robotics and Automation,* 2001.

[6] H. Durrant-Whyte H., T. Bailey. "Simultaneous localization and mapping (SLAM): part I essential algorithms". IEEE Robotics and Automation Magazine, vol. 13, pp. 99-108. 2006.

[7] H. Durrant-Whyte H., T. Bailey. "Simultaneous localization and mapping (SLAM): part II state of the art". IEEE Robotics and Automation Magazine, vol. 13, pp. 108-117. 2006.

[8] Thrun S., Burgard W., Fox D.. *Probabilistic Robotics*. *MIT Press*, Cambridge, 2005.

[9] B. Xi, R. Guo, F. Sun and Y. Huang. "Simulation Research for Active Simultaneous Localization and Mapping Based on Extended Kalman Filter". Proc. of the IEEE International Conference on Automation and Logistics. 2008.

[10] T. Tao, Y. Huang, F. Sun, T. Wang. "Motion Planning for SLAM Based on Frontier Exploration". In: Proc. of the IEEE International Conference on Mechatronics and Automation. 2007.

[11] D. Anguelov, D. Koller, E. Parker, S. Thrun. "Detecting and Modeling Doors with Mobile Robots". IEEE International Conference on Robotics and Automation. 2004.

[12] R. Biswas, B. Limketkai, S. Sanner, S. Thrun. "Towards object mapping in dynamic environments with mobile robots". In: Proc. IROS, 2002.

[13] T. F. Bastos Filho, A. Ferreira, R. Silva, W. C. Celeste, M. Sarcinelli. "Human-Machine Interface Based on EMG and EEG Signals Applied to a Robotic Wheelchair"**.** *Journal of Physics Conference Series*, **1**:1/012088-10. 2007.

[14] S. Theodoris, K. Koutroumbas. *Pattern Recognition*. Elsevier. 2003.

[15] J. E. Guivant, E. M. Nebot. "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation". IEEE Transactions on Robotics and Automation, 17:242-257. 2001.

[16] L. Bauwens, L. Cizkova, et al. *Computational Statitistics*. Springer. 2004.

[17] J. P. Laumond. Robot Motion, Planning and Control. Lecture Notes in Control and Information Science 229. Springer. 1998.

[18] F. A. Auat Cheein, F. di Sciascio, R. Carelli. "Planificación de Caminos y Navegación de un Robot Móvil en Entornos Gaussianos mientras realiza tareas de SLAM". Jornadas Argentinas de Robotica. 2008.

[19] A. Garulli, A. Giannitrapani, A. Rosi, A. Vicino. "Mobile robot SLAM for line-based environment representation". In: Proceedings of the 44th IEEE Conference on Decision and Control, 2005.

[20] M. Mazo. "An integral system for assisted mobility". *IEEE Robotics and Automation Magazine*, vol. 8(1), pp. 46-56, 2001.

[21] Q. Zeng, Ch.L. Teo, B. Rebsamen and E. Burdet. "A collaborative wheelchair system". *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16(2), pp. 161-170, 2008.

[22] S.P. Parikh, V. Grassi, V. Kumar and J. Okamoto. "Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform". *IEEE Intelligent Systems*, vol. 22(2), pp. 33-41, 2007.

[23] C. De La Cruz, R. Carelli and T.F. Bastos. "Switching adaptive control of mobile robots". *IEEE International Symposium on Industrial Electronics – ISIE08*, pp. 835-840, 2008.

[24] C. De La Cruz and R. Carelli. "Dynamic model based formation control and obstacle avoidance of multi-robot systems". *Robotica*, vol. 26(3), pp. 345-356, 2008.

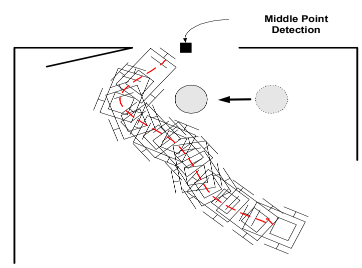

Fig. 6. Obstacle avoidance during the navigation. The path travelled by the vehicle is drawn in dashed line.

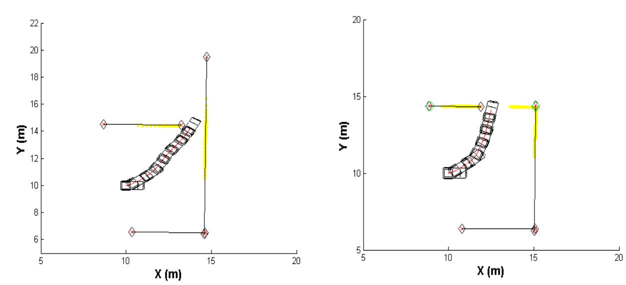

Fig. 7. SLAM results for the wheelchair navigation.