THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2009, Vol.32, No.2, ISSN 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

SLIDING MODE CONTROL FOR TRAJECTORY TRACKING OF AN INTELLIGENT WHEELCHAIR

Razvan Solea[*], Urbano Nunes[**], Adrian Filipescu*, Daniela Cernega*

[*]*Department of Automation and Industrial Informatics, "Dunarea de Jos" University of Galati, 800-008 Galati, Romania*
[**]*ISR - Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal*

Abstract: This paper deal with a robust sliding-mode trajectory tracking controller, for nonholonomic wheeled mobile robots and its experimental evaluation by the implementation in an intelligent wheelchair (RobChair). The proposed control structure is based on two nonlinear sliding surfaces ensuring the tracking of the three output variables, with respect to the nonholonomic constraint. The performances of the proposed controller for the trajectory planning problem with comfort constraint are verified through the real time acceleration provided by an inertial measurement unit.

Keywords: sliding-mode control, trajectory-tracking mode, intelligent wheelchair, mobile robots.

1. INTRODUCTION

This paper proposes a robust sliding-mode trajectory-tracking (SM-TT) controller for nonholonomic wheeled mobile robots (WMR). This controller was primarily intended to be used in an intelligent wheelchair, called RobChair (Pires and Nunes, 2002), (Lopes, *et al.*, 2007). RobChair has been developed for rehabilitation applications and mobility assistance of persons with special needs (e.g. persons with disability), with the purpose of providing them a certain degree of autonomy and independence. RobChair is based on a commercial wheelchair, which has been equipped with an intelligent control system and several sensors.

Motion planning (Laumond, 1998), (LaValle, 2006) and control of WMRs is subject of interest for many researchers. Due to their non-holonomic properties, restricted mobility and their relevance in applications, the trajectory-tracking problem for these systems became a class of control problems. Variable structure control (VSC) emerges as a robust approach in different applications and has been successfully applied in control problems as diverse as automatic flight control, control of electrical motors, of chemical processes, helicopter stability augmentation, space systems and robotics. One particular type of VCS system is the sliding mode control (SMC) approach (Utkin, 1992). The theory of SMC has been applied to various control systems, since it has been shown that this type of nonlinear control exhibits some excellent properties, such as robustness against large parameter variation and disturbances (Slotine and Li, 1991), (Utkin, *et al.*, 1999).

The trajectory-tracking control means tracking desired trajectories predefined or given by path planners. In the last years, there are so ample research works on the trajectory tracking control of mobile robots and autonomous vehicles that various

This paper was recommended for publication by Sergiu Caraman

effective methods and tracking controllers have been developed (Chwa, 2004), (Yang and Kim, 1999), (Oriolo, et al., 2002), (Fierro and Lewis, 1997), (Klancar and Skrjanc, 2007), (Chwa, et al., 2007), (Solea and Nunes, 2007).

According to different control theories, a more refined classification in the sense of kinematics methods can be grouped in five sets: (1) sliding mode based approaches, (2) input-output linearization based approaches, (3) fuzzy based approaches, (4) neural network based approaches and (5) backstepping based approaches. Among all of these kinematics-based methods, considering the stability of tracking control laws, tracking control law designed by sliding mode methodology has been proved one of the best solutions.

Yang and Kim (1999) proposed a sliding mode control law for solving trajectory tracking problems of nonholonomic WMRs in polar coordinates. A kinematics-based SMC controller defined in polar coordinates, for trajectory-tracking of WMRs, is described in (Chwa, 2004). In this method, two controllers are designed to asymptotically stabilize the tracking errors in position and heading direction, respectively. In (Solea and Nunes, 2007) are described the design of SMC-based trajectory-tracking and path-following controllers, and its evaluation by simulations, for autonomous vehicles. A new design of sliding surface was proposed, such that lateral and angular errors are internally coupled (in Cartesian space) in a sliding surface leading to convergence of both variables.

2. CONTROL OF WHEELED MOBILE ROBOTS

The application of SMC strategies in nonlinear systems has been used in recent years (Chwa, 2004), (Yang and Kim, 1999), (Chwa, *et al.*, 2006) , (Solea and Nunes, 2007), (Canudas, et al., 1993), (Dixon, et al., 2000), (Floquet, *et al.*, 2003). A well-studied example of a non-holonomic system is a WMR that is subject to the *rolling without slipping* constraint.

In trajectory-tracking the path has to be followed under time constraints. The path has an velocity profile associated, with each point of the trajectory embedding spatiotemporal information that is to be satisfied by the WMR along the path. Trajectory tracking is formulated as having the WMR following a virtual target WMR which is assumed to move exactly along the path with specified velocity profile.

1.1. Kinematic model of a WMR.

In Figure 1 a WMR with two diametrically opposed drive wheels (radius *R*) and free-wheeling castors (not considered in the kinematics models) is presented. *Pr* is the origin of the robot coordinates system. *L* is the length of the axis between the drive wheels. ω_R and ω_L are the angular velocities of the right and left wheels. Let the pose of the mobile robot be defined by the vector $q_r = [x_r, y_r, \theta_r]^T$, where $[x_r, y_r]^T$ denotes the robot position on the plane and θ_r the heading angle with respect to the *x-axis*. In addition, v_r denotes the linear velocity of the robot and ω_r the angular velocity around the vertical axis. For a unicycle WMR rolling on a horizontal plane without slipping, the kinematic model can be expressed by:

$$(1)\quad \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix}$$

which represents a nonlinear system.

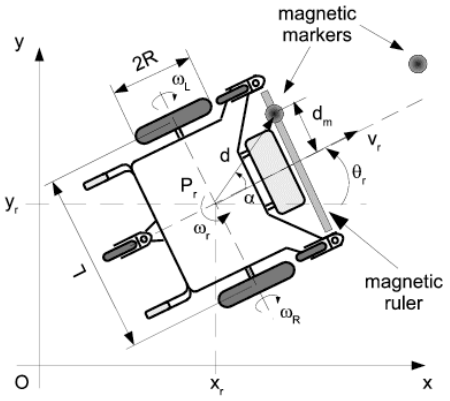

Fig.1. Robotic wheelchair model and symbols.

The system controllability (1) is easily checked using the Lie algebra rank condition for nonlinear systems. However, the Taylor linearization of the system about the origin is not controllable, thus excluding the application of classical linear design approaches.

1.2. Trajectory-tracking

Let us consider first the trajectory-tracking control case. Without loss of generality, it can be assumed that the desired trajectory $q_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T$ is generated by a virtual unicycle mobile robot (see Fig. 2). The kinematics relationship between the virtual configuration $q_d(t)$ and the corresponding desired velocity inputs $[v_d(t), \omega_d(t)]^T$ is similar with (1):

$$(2)\quad \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ \omega_d \end{bmatrix}$$

When a real robot is controlled to move on a desired path it exhibits some tracking errors. This tracking error, expressed in terms of the robot coordinate system, as shown in Fig. 2, is given by

$$(3)\quad \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix}$$

Consequently one gets the error dynamics for trajectory tracking as

$$(4)\quad \begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos\theta_e + \omega_d \cdot y_e \\ \dot{y}_e = v_r \cdot \sin\theta_e - \omega_d \cdot x_e \\ \dot{\theta}_e = \omega_r - \omega_d \end{cases}$$

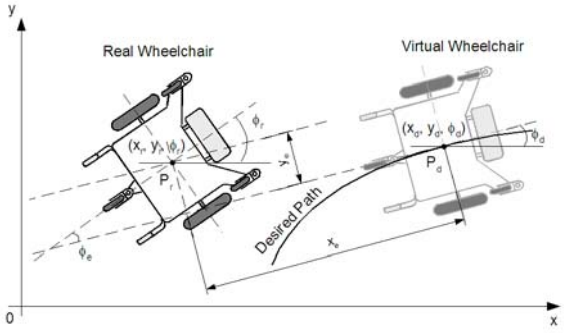

Fig.2. Lateral, longitudinal and orientation errors (trajectory-tracking).

The existence of the rigid law *Pd = Pd(t)* in TT means pulling or dragging the robot to reach the desired path

The lateral error y_e is defined as the distance between the origin of the robot coordinates system (*Pr*) and the desired trajectory (this implies that y_e is perpendicular to the tangent to the desired trajectory at *Pd*).

1.3. Sliding-mode control

The SMC is a model-based control strategy in which the controller structure and gains are designed based on the system model. When the system errors become zero, the plant follows the desired trajectory. Under these conditions *s = 0*, and the system state trajectory follows the desired trajectory and is restricted to lie on what is called the *sliding surface* (or *switching surface*).

The function of the SMC is to produce a discontinuous control signal which forces the system states to repeatedly cross and then immediately re-cross the sliding surface until it finally slides along the surface *s = 0*. This kind of motion is referred to as *sliding motion*. In sliding mode, due to the discontinuous characteristics of command, the states would *switch* about the sliding surface rather than lie directly on it. This switching can occur at a high frequency and is called *chattering*. To reduce chattering, the *smoothing boundary layer solution* developed in (Slotine and Li, 1991) is applied to the control law of SMC. The saturation function *sat(s, ψ)* is used to replace the sign function *sgn(s)* inside of boundary layer, $s \le \psi$

Slotine's (1991), (1984) original proposal was based on the generalized case of the n^{th}-order single input type of nonlinear system:

$$(5)\quad x^{(n)} = f(x,t) + b(x,t) \cdot u$$

where *x* is the state variable, $x = \left[x,\ \dot{x},\ \ddot{x},\ \ldots\ x^{(n-1)}\right]$; $x^{(n)}$ is the n^{th}-order derivative of *x*; *f* is a nonlinear function; *b* is the gain and *u* is the control input.

Moreover, a formula for the distance between the state trajectory and the switching manifold of the above system is:

$$(6)\quad s(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \cdot \tilde{x}$$

Afterwards Gao and Hung (1993) proposed the method of reaching mode and reaching law, based on *m*-input n[th]-order systems. In order to assure the attraction of state trajectory onto the switching manifold within the reaching mode, they suggested the control of reaching speed by certain reaching law. Besides the general form,

$$(7)\quad \dot{s} = -Q \cdot g(s) - P \cdot \mathrm{sgn}(s)$$

$$Q = diag\left[q_1, q_2, ..., q_m\right]\ q_i > 0,$$
$$P = diag\left[p_1, p_2, ..., p_m\right]\ p_i > 0,$$

$$\mathrm{sgn}(s) = \left[\mathrm{sgn}(s_1), \mathrm{sgn}(s_2), ..., \mathrm{sgn}(s_m)\right]^T$$
$$g(s) = \left[g(s_1), g(s_2), ..., g(s_m)\right]^T$$
$$s_i \cdot g_i(s_i) > 0, g_i(0) = 0$$

they proposed three particular types of reaching laws:

a) constant rate reaching

$$(8)\quad \dot{s} = -P \cdot \mathrm{sgn}(s)$$

This law forces the switching variable *s* to reach the switching manifold at a constant rate $\left|\dot{s}_i\right| = -p_i$.

b) constant plus proportional rate reaching

$$(9)\quad \dot{s} = -Q \cdot s - P \cdot \mathrm{sgn}(s)$$

By adding the proportional rate term –Q · s, the state is forced to approach the switching manifolds faster when *s* is large.

c) power rate reaching

$$\dot{s} = -Q \cdot |s|^{\alpha} \cdot \mathrm{sgn}(s), \quad 0 < \alpha < 1 \tag{10}$$

This reaching low increases the reaching speed when the state is far away from the switching manifold, but reduces the rate when the state is near the manifold.

Sliding-mode trajectory-tracking control.

Uncertainties which exist in real mobile robot applications degrade the control performance significantly, and accordingly, need to be compensated. In this section, we formulate a SM-TT controller, in Cartesian space, where trajectory-tracking is achieved even in the presence of large initial pose errors and disturbances.

Let us define the sliding surface $s = [s_1\ s_2]^T$ as

$$\begin{aligned} s_1 &= \dot{x}_e + k_1 \cdot x_e \\ s_2 &= \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \theta_e \end{aligned} \tag{11}$$

where k_0, k_1, k_2 are positive constant parameters, x_e, y_e and θ_e are the trajectory-tracking errors defined in (2).

If s_1 converges to zero, trivially x_e converges to zero. If s_2 converges to zero, in steady-state it becomes $\dot{y}_e = -k_2 \cdot y_e - k_0 \cdot \mathrm{sgn}(y_e) \cdot \theta_e$. For $y_e < 0 \Rightarrow \dot{y}_e > 0$ if only if $k_0 < k_2 |y_e| / |\theta_e|$. For $y_e > 0 \Rightarrow \dot{y}_e < 0$ if only if $k_0 < k_2 |y_e| / |\theta_e|$. Finally, it can be known from s_2 that convergence of y_e and $\dot{y}_e$ leads to convergence of θ_e to zero.

From the time derivative of (11) and using the reaching law defined in (9):

$$\begin{aligned} \dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot s_1 - p_1 \cdot \mathrm{sgn}(s_1) \\ \dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \dot{\theta}_e = \\ &= -q_2 \cdot s_2 - p_2 \cdot \mathrm{sgn}(s_2) \end{aligned} \tag{12}$$

From (4) and (12), and after some mathematical manipulation, the output commands of the trajectory-tracking controller are determined:

$$\begin{aligned} \dot{v}_c &= \frac{1}{\cos\theta_e} \cdot (-q_1 \cdot s_1 - p_1 \cdot \mathrm{sgn}(s_1) - k_1 \cdot \dot{x}_e - \\ &- y_e \cdot \dot{\omega}_d - \dot{y}_e \cdot \omega_d + v_r \cdot \dot{\theta}_e \cdot \sin\theta_e + \dot{v}_d) \\ \omega_c &= \frac{1}{v_r \cdot \cos\theta_e + k_0 \cdot \mathrm{sgn}(y_e)} \cdot (-q_2 \cdot s_2 - k_2 \cdot \dot{y}_e - \\ &- p_2 \cdot \mathrm{sgn}(s_2) + x_e \cdot \dot{\omega}_d + \dot{x}_e \cdot \omega_d - \dot{v}_r \cdot \sin\theta_e) + \omega_d \end{aligned} \tag{13}$$

Let us define $V = \frac{1}{2} \cdot s^T$ as a Lyapunov function candidate, therefore its time derivative is

$$\begin{aligned} \dot{V} &= s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = \\ &= s_1 \cdot \left(-q_1 \cdot s_1 - p_1 \cdot \mathrm{sgn}(s_1)\right) + \\ &+ s_2 \cdot \left(-q_2 \cdot s_2 - p_2 \cdot \mathrm{sgn}(s_2)\right) = \\ &= -s^T \cdot q \cdot s - p_1 \cdot |s_1| - p_2 \cdot |s_2| \end{aligned}$$

For $\dot{V}$ to be negative semi-definite, it is sufficient to choose q_i and p_i such that $q_i, p_i > 0$.

The SM-TT architecture with an on-line robot's pose estimator, fusing odometry with absolute position data as described in (Lopes, et.al., 2007), is depicted in Fig. 3.

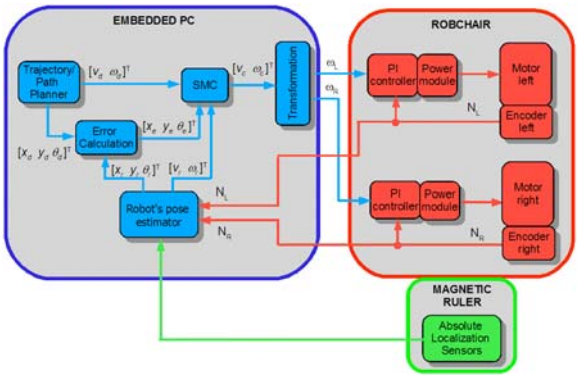

Fig.3. SM-TT control architecture with a robot's pose estimator, fusing odometry and absolute position data

RobChair has a two-level control architecture (see Fig. 3. High-level control algorithms (including desired motion generation) are written in *C* and run with a sampling time of *Ts = 50 ms* on an embedded PC, which also provides a user interface with real-time visualization and a simulation environment. The PC communicates through a CAN bus with several devices. Wheel velocity commands,

$$\omega_R = \frac{v_c + \frac{L}{2} \cdot \omega_c}{R}, \quad \omega_L = \frac{v_c - \frac{L}{2} \cdot \omega_c}{R} \tag{14}$$

are sent to the PI controllers and the encoder measurements N_R and N_L are received in the robot's pose estimator for odometric computations.

The low-level control layer is in charge of the execution of the wheels velocity control. For each wheel, a microcontroller implements a digital PI with a cycle time of Tc = 5 ms. Two power amplifiers drive the motors with PWM voltage.

1.4. Trajectory planner

A trajectory planner for passenger's transport vehicles must generate smooth velocity profiles (linear and angular) with low associated accelerations. The trajectory planning process can be divided into two separate parts. First, a continuous collision-free path is generated. In a second step, called trajectory generation, a velocity profile along the path is determined. A method to generate a velocity profile, with respect to human body comfort, for any two-dimensional path in static environments was is primarily proposed in (Solea and Nunes, 2007}.

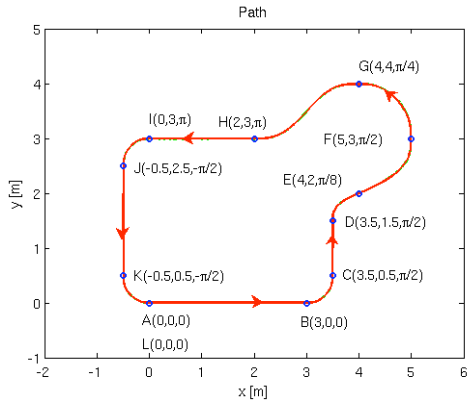

Fig.4. Path example composed by eleven path segments calculated by the trajectory planner, from the eleven waypoints A to L..

The ISO 2631-1 standard (ISO, 1997) relates comfort with the overall root mean square (r.m.s.) acceleration (a_w), acting on the human body (see Table 1):

$$(15)\quad a_w = \sqrt{\tau_x^2 \cdot a_{wx}^2 + \tau_y^2 \cdot a_{wy}^2 + \tau_z^2 \cdot a_{wz}^2}$$

where a_{wx}, a_{wy}, a_{wz}, are the r.m.s. accelerations on *x*, *y*, *z* axes respectively, and τ_x, τ_y, τ_z, are multiplying factors. For a seated person $\tau_x = \tau_y = 1.4$, $\tau_z = 1$. For motion on the *xy-plane*, $a_{wz} = 0$.

Table 1 ISO 2631-1 Standard.

Overall Acceleration	Consequence
$a_w < 0.315\ m/s^2$	Not uncomfortable
$0.315 < a_w < 0.63\ m/s^2$	A little uncomfortable
$0.5 < a_w < 1\ m/s^2$	Fairly uncomfortable
$0.8 < a_w < 1.6\ m/s^2$	Uncomfortable
$1.25 < a_w < 2.5\ m/s^2$	Very uncomfortable
$a_w > 2.5\ m/s^2$	Extremely uncomfortable

Figures 4 - 6 show an example of a planned trajectory using the algorithm, were the goal was to obtain an overall r.m.s. acceleration in the range of "not uncomfortable" ($a_w < 0.31\ m/s^2$). Figure 5 shows the generated path, where the larger circles represent the used twelve waypoints (A, B, ..., L). Each waypoint is defined by a position, in meters, and an orientation, in radians. Velocity and acceleration profiles are shown in Fig. 5.

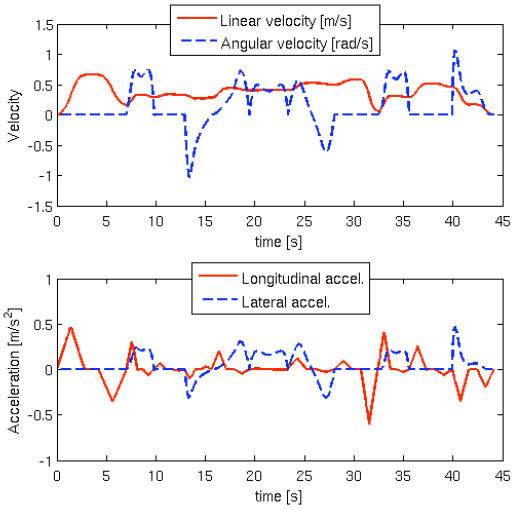

Fig.5. Velocities and accelerations for the path depicted in Fig. 4

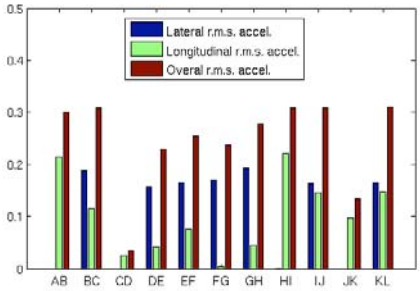

Fig.6. R.M.S. acceleration values for each path segment (AB, BC, ..., KL) of path depicted in Fig. 4

As can be observed in Fig. 6, the r.m.s overall accelerations, in each path segment, are below the imposed acceleration constraint of $a_w < 0.31\ m/s^2$.

3. ROBCHAIR ARCHITECTURE

RobChair, shown in Fig. 7, has two differentially driven rear wheels and two passive castor front wheels. There is also a fifth rear wheel connected to the back of the wheelchair with a damper used for stability. It is powered by two 12 V batteries (60 Ah) and reaches a maximum speed of 7 km/h. It has been equipped, in ISR-UC, with several devices such as: two power driver modules, which provide an independent control of each motor, optical encoders, laser range finders, an inertial sensor and a magnetic sensing ruler (Lopes, et al., 2007). Figure 8 presents a

block diagram of the actual hardware control architecture. The current implementation of the framework is based on Linux as its underlying real-time operating system. The component-based software selected for the proposed software framework is GenoM (Generator of Modules) (Fleury, et al., 1997), which is an environment for description and implementation of software components.

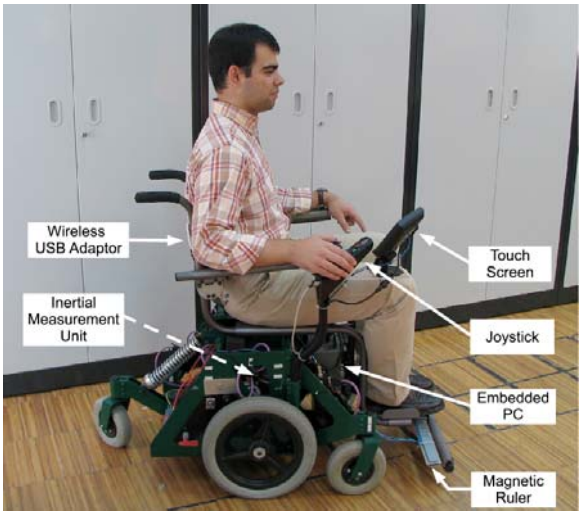

Fig.7. RobChair platform.

An embedded PC is responsible for giving some degree of intelligence to the robot. This computer is connected to distributed devices through fieldbuses. The platform is connected to external devices through a wireless link. This connection allows the implementation of a distributed architecture, which exhibits the possibility and capability to extend our single robot to other perspectives, like multi-robot cooperation, its integration in intelligent environments etc.

Each distributed device, connected through CAN, use a printed circuit board card, containing a microchip micro-controller (PIC boards in Fig. 8) (Maia, 2004). A custom communication protocol, based on the time-triggered protocol paradigm, was designed and implemented. All events are synchronized by a message, sent from a Synchronization Micro-Controller Unit (Trigger Node, in Fig. 8), that synchronizes the other Micro-Controller Units, and defines the control loop time reference.

The RobChair is equipped with a magnetic sensing ruler (MSR) developed at ISR-UC that is able to perform a robust detection of magnetic markers (Lopes, et al., 2007). The experimental results with Robchair primarily show that the detection system is robust, since it is able to detect true magnetic markers, and to eliminate noisy magnetic distortions and false markers.

The odometric data calculated based on the wheel encoders' is fused with the data from magnetic markers. The extended Kalman filter (EKF) was chosen for the fusion process. The vehicle's pose defined by the Cartesian coordinates (x_r, y_r) and heading (θ_r) are the state variables of the EKF. The magnetic ruler measures are treated as measurements in the fusion process. This navigation technology, based on sensing magnetic markers, is well suited when high precision navigation and robustness is required, and it can be used to complement other navigation systems, such as GPS.

The inertial sensor RGA300CA-100 (Crossbow) was used for measuring the wheelchair accelerations in three orthogonal directions.

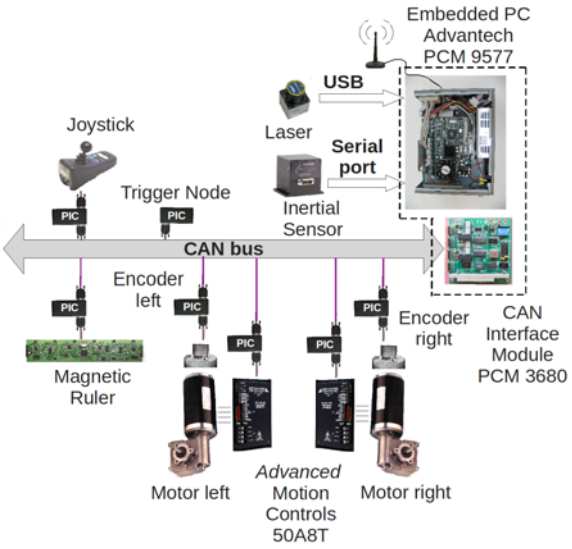

Fig.8. RobChair hardware architecture.

4. EXPERIMENTAL RESULTS

A set of experimental tests, using RobChair (Pires and Nunes, 2002), (Lopes, et al., 2007), has been performed to evaluate the performance of the sliding mode controllers (SM-TT and SM-PF) and the trajectory planning algorithm with comfort constraint.

According to the notation shown in Fig. 1, the RobChair parameter values are $L = 0.614\ m$ and $R = 0.175\ m$. For all the experiments, the following values of the SM-TT controller (13) were used: $k_0=4$, $k_1=0.75$, $k_2=5$, $p_1=0.05$, $p_2=0.05$, $q_1=0.75$, $q_2=1.75$. The signum functions in the command signals (13) were replaced by saturation functions with ±0.15 thresholds, to reduce the chattering phenomenon.

1.5. Experimental results under odometry navigation

As shown in Fig. 9, the first experiment reported in the paper concerns the implementation of the SM-TT (using (13)) in the following conditions: 1) only odometry feedback is used; 2) initial pose error (xe=-*1*, ye = *-1* and θ_e = *0*). The same figure shows that the RobChair retrieved quickly ($\Delta t \approx 10$ *s*) and smoothly from its initial state error, and the tracking errors converge on average to zero with acceptable reduced values along the path.

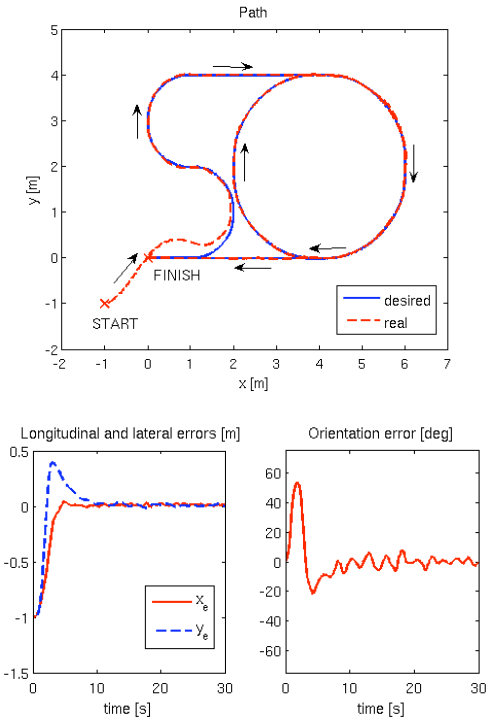

Fig.9. Experimental SM-TT control starting from an initial error state ($x_e(0)$=*-1*, $y_e(0)$ = *-1*, $\theta_e(0)$ = *0*), under odometry navigation.

Experimental results for SM-TT controller under odometry navigation (without initial pose errors) for two different trajectories (as is shown in Fig. 10) are presented in Table 2. Three experimental trials were made for each path. The table shows the maximum (Max), root mean square (RMS), RMS accelerations a_{wx}, a_{wy} and a_{wz} (on *x*, *y* and *z* axes), and the overall RMS acceleration a_w values (see eq. (15)).

The maximum absolute values of lateral and longitudinal errors are smaller than 0.065 m and the overall r.m.s. acceleration values (a_w) are below 0.5 m/s^2. This value is in the range of *"a little uncomfortable"* (see Table 1).

1.6. Experimental results under magnetic-markers navigation

Another set of experiments using SM-TT controller (Fig. 10) was made for two type of trajectories without initial pose errors when the odometric date is fused with absolute position data from magnetic markers.

Figures 11 and 12 shows desired, command and real linear and angular velocities for SM-TT control in case of Path-2. Corrections in the pose after each magnetic marker detection provokes an error signal that is efficiently dealt by the SM-TT controller, and rapidly the tracking errors converge to zero (see Fig. 13). All the experimental results for SM-TT controllers under magnetic-markers navigation are shown in Table 3.

The maximum absolute values of lateral and longitudinal errors are under *0.28 m* and the overall r.m.s. acceleration values (a_w) are below *0.55 m/s*^{2}.

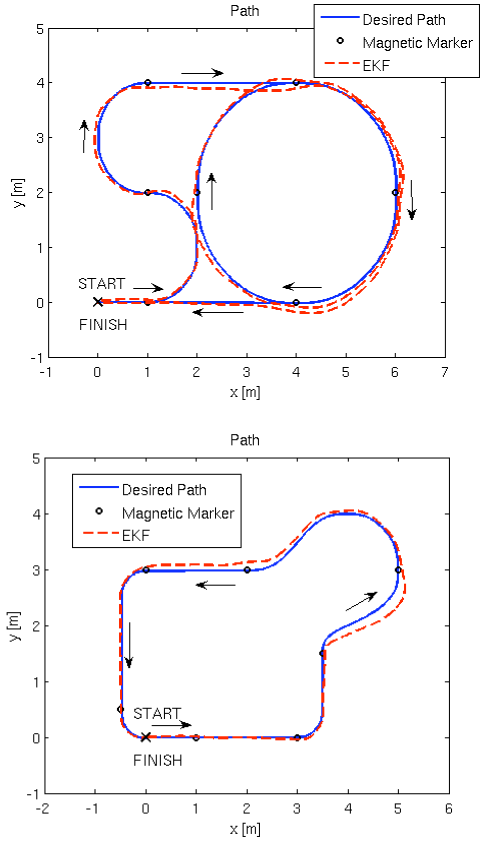

Fig.10. Experimental SM-TT control under magnetic-markers navigation: Top) *Path-1*; Bottom) *Path-2*. In both paths, the circles represent the positions of the seven magnetic markers used in the experiment.

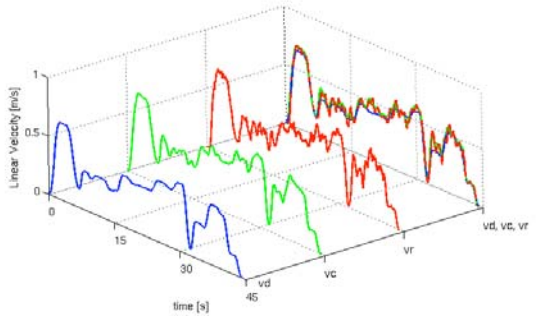

Fig.11. Desired (v_d), command (v_c) and real (v_r) linear velocities for SM-TT control using an EKF-based fusion in the on-line pose estimation - *Path-2*.

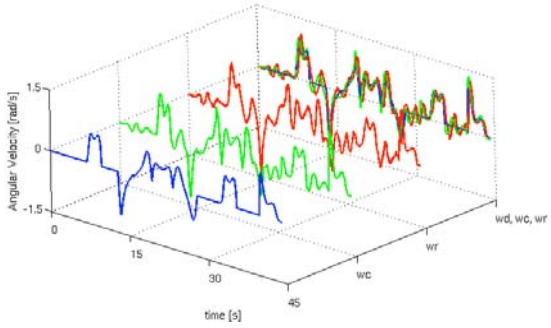

Fig.12. Desired (ω_d), command (ω_c) and real (ω_r) angular velocities for SM-TT control using an EKF-based fusion in the on-line pose estimation - *Path-2*.

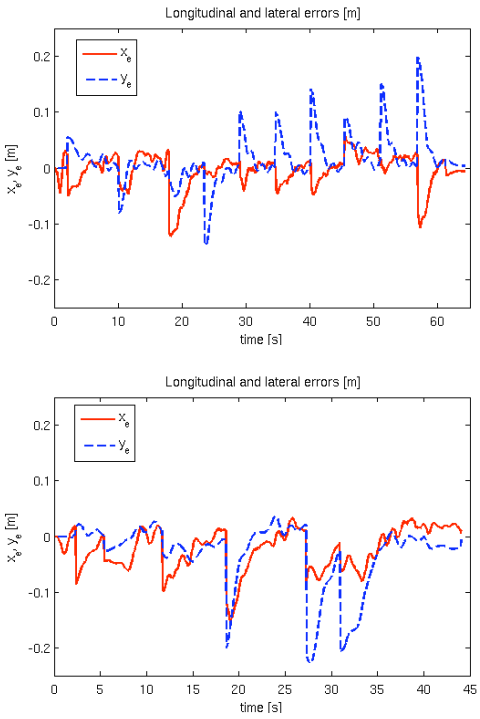

Fig.13. Longitudinal and lateral errors of SM-TT control under magnetic-markers navigation: Top) *Path-1*; Bottom) *Path-2*.

Table 2 Experimental results for SM-TT controller under odometry navigation.

Case	No	Max x_e [m]	RMS x_e [m]	Max y_e [m]	RMS y_e [m]	Max. θ_e [rad]	RMS θ_e [rad]	a_{wx} $[m/s^2]$	a_{wy} $[m/s^2]$	a_{wz} $[m/s^2]$	a_w $[m/s^2]$
	I	0.0419	0.0152	0.0612	0.0140	0.2263	0.0461	0.2407	0.2029	0.0618	0.4452
$Path-1$	II	0.0450	0.0159	0.0448	0.0113	0.1689	0.0426	0.2417	0.2052	0.0600	0.4480
	III	0.0493	0.0182	0.0864	0.0195	0.2787	0.0514	0.2354	0.1891	0.0570	0.4266
average		**0.0454**	**0.0164**	**0.0641**	**0.0149**	**0.2246**	**0.0467**	**0.2393**	**0.1991**	**0.0596**	**0.4399**
	I	0.0581	0.0245	0.0614	0.0190	0.2935	0.0769	0.2696	0.2356	0.0564	0.5045
$Path-2$	II	0.0498	0.0232	0.0560	0.0181	0.3775	0.0783	0.2710	0.2149	0.0531	0.4872
	III	0.0492	0.0211	0.0406	0.0137	0.2171	0.0649	0.2777	0.2246	0.0572	0.5034
average		**0.0524**	**0.0229**	**0.0527**	**0.0169**	**0.2960**	**0.0734**	**0.2728**	**0.2250**	**0.0556**	**0.4984**

Table 3 Experimental results for SM-TT controller under magnetic-markers navigation.

Case	No	Max x_e [m]	RMS x_e [m]	Max y_e [m]	RMS y_e [m]	Max. θ_e [rad]	RMS θ_e [rad]	a_{wx} $[m/s^2]$	a_{wy} $[m/s^2]$	a_{wz} $[m/s^2]$	a_w $[m/s^2]$
	I	0.2528	0.0923	0.2932	0.0768	0.3779	0.0950	0.2275	0.2138	0.0584	0.4410
$Path-1$	II	0.2210	0.0517	0.1982	0.0424	0.2572	0.0721	0.2502	0.2247	0.0709	0.4761
	III	0.2039	0.0748	0.1945	0.0452	0.2527	0.0642	0.2222	0.1977	0.0652	0.4215
average		**0.2259**	**0.0729**	**0.2286**	**0.0548**	**0.2959**	**0.0771**	**0.2333**	**0.2121**	**0.0648**	**0.4462**
	I	0.1626	0.0586	0.2222	0.0626	0.2299	0.0698	0.3071	0.2234	0.0518	0.5342
$Path-2$	II	0.1647	0.0511	0.2390	0.0687	0.2585	0.0758	0.2892	0.2447	0.0565	0.5334
	III	0.1849	0.0515	0.2739	0.0685	0.2982	0.0814	0.3023	0.2416	0.0753	0.5470
average		**0.1707**	**0.0537**	**0.2450**	**0.0666**	**0.2622**	**0.0757**	**0.2995**	**0.2366**	**0.0612**	**0.5382**

5. CONCLUSION

This paper describes a trajectory-tracking controller based on the sliding-mode theory for WMRs. The proposed control structure is based on two nonlinear sliding surfaces ensuring the tracking of the three output variables, exploiting the nonholonomic constraint. The control law has been thoroughly evaluated in terms of tracking performance either by simulation and real experiments.

The experimental tests presented in this paper are representative of the average performance of the controllers. We summarize our acquired experience in general observations that can be useful guidelines for implementation of the same control strategies in other type of mobile robots.

6. REFERENCES

Canudas de Wit C., H. Khennouf, C. Samson and O. Sordalen (1993), Nonlinear control design for mobile robots. *Recent Trends in Mobile Robotics*, 11, Word Scientific Series in Robotics and Automated Systems, pp. 121–156.

Chwa, D. (2004). Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Transactions on ControlSystems Technology*, 12(4), pp. 637–644.

Chwa D., S. K. Hong and B. Song. (2006). Robust posture stabilization of wheeled mobile robots in polar coordinates, *The 17th International Symposium on Mathematical Theory of Networks and Systems*, pp. 343-348.

Dixon W. E., Z. P. Jiang and D. M. Dawson (2000). Global exponential setpoint control of wheeled mobile robots: a Lyapunov approach, *Automatica*, 36, pp. 1741–1746.

Fierro R. and F. L. Lewis. (1997). Control of a nonholonomic mobile robot: backstepping kinematics into dynamics, *Journal of Robotic Systems*, 14(3), 149-163.

Fleury, S., M. Herrb, and R. Chatila, (1997). GenoM: A tool for the specification and the implementation of operating modules in a distributed robot architectures, *IEEE International Conference on Intelligent Robots and Systems*, Grenoble, pp. 842-848.

Floquet, T., J. P. Barbot and W. Perruquetti.(2003). Higher-order sliding mode stabilization for a class of nonholonomic perturbed systems, *Automatica*, 39, pp. 1077–1083.

Gao W. and J. C. Hung, (1993). Variable structure control of nonlinear systems: A new approach, *IEEE Transactions on Industrial Electronics*, 40(1), pp. 45–55.

ISO-2631 (1997). Mechanical vibration and shock - Evaluation of human exposure to whole body vibration - Part 1: General requirements, International Organization for Standardization.

Klancar G., I. Skrjanc. (2007). Tracking-error model-based predictive control for mobile robots in real time, *Journal of Robotics and Autonomous Systems*, 55, 460–469.

LaValle, S. M. (2006). *Planning Algorithms*, Cambridge University Press, Cambridge, U.K.

Laumond, J. (1998). *Robot Motion Planning and Control*. Lectures Notes in Control and Information Sciences, Springer.

Lopes, A.C., F. Moita, U. Nunes, and R. Solea (2007). An outdoor guidepath navigation system for AMRs based on robust detection of magnetic markers. *12th IEEE International Conference on Emerging Technologies and Factory Automation*, Patras, pp. 989–996.

Maia, R. (2004). Movimento de robots moveis com rodas de tracao diferencial:modelacao e controlo do sistema motriz, MSc thesis, Universidade de Coimbra (in portuguese).

Oriolo G., A. De Luca and M. Vendittelli. (2002) WMR Control via dynamic feedback linearization: design, implementation, and experimental validation, *IEEE Transactions on Control Systems Technology*, 10(6), pp. 835–852.

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

Slotine J. J. E. and W. Li, (1991). *Applied Nonlinear Control*. Prentice-Hall.

Slotine, J. J. E. (1984). Sliding Controller Design for Nonlinear Systems, *International Journal of Control*, 40(2), pp.421–434.

Solea R. and U. Nunes (2007). Trajectory planning and sliding-mode control based trajectory-tracking for cybercars, *Integrated Computer-Aided Engineering*, IOS Press, 14(1), pp. 33–47.

Utkin V. I. (1992). *Sliding modes in optimization and control*, Springer-Verlag, New York.

Utkin, V. I., J. Guldner, and J. Shi. (1999) *Sliding mode control in electromechanical systems*, Taylor & Francis, London.

Yang J. M. and J. H. Kim. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots, *IEEE Transactions on Robotics and Automation*, 15(3), pp. 578–587.