Integrated Computer-Aided Engineering 13 (2006) 1–15 1
IOS Press

Trajectory planning and sliding-mode control based trajectory-tracking for cybercars

Razvan Solea and Urbano Nunes*
Institute of Systems and Robotics, University of Coimbra, Polo II, 3030-290 Coimbra, Portugal
Tel.: +351 239 796200; Fax: +351 239 406672; E-mail: urbano@deec.uc.pt

Abstract. Fully automatic driving is emerging as the approach to dramatically improve efficiency (throughput per unit of space) while at the same time leading to the goal of zero accidents. This approach, based on fully automated vehicles, might improve the efficiency of road travel in terms of space and energy used, and in terms of service provided as well. For such automated operation, trajectory planning methods that produce smooth trajectories, with low level associated accelerations and jerk for providing human comfort, are required. This paper addresses this problem proposing a new approach that consists of introducing a velocity planning stage in the trajectory planner. Moreover, this paper presents the design and simulation evaluation of trajectory-tracking and path-following controllers for autonomous vehicles based on sliding mode control. A new design of sliding surface is proposed, such that lateral and angular errors are internally coupled with each other (in cartesian space) in a sliding surface leading to convergence of both variables.

1. Introduction

The number of accidents generated by road transport is a dramatic social problem requiring urgent and effective solutions. Improvements in vehicle quality have helped to increase safety and capacity, but as vehicle safety and traffic engineering has improved, the proportion of accidents due to driver error has increased with a result that automotive engineering has focused on accident mitigation rather than avoidance. The introduction of safer vehicles has sometimes been associated with greater accident frequency, particularly involving vehicle-pedestrian conflicts where drivers are encouraged to take greater risks as a result of their perceived invulnerability. It is now recognized that the limiting factor for further improvement is the poor performance of the human driver. Fully automatic driving is emerging as the approach to dramatically improve efficiency while at the same time leading to the goal of zero accidents. In this context, a new approach for mobility providing an al ternative to the private passenger car, by offering the same flexibility but with much less nuisances, is emerging, based on fully automated vehicles, named cybercars [4,9].

Although motion planning of mobile robots and autonomous vehicles has been thoroughly studied in the last decades, the requisite of producing trajectories with low associated accelerations and jerk is not easily traceable in the technical literature. This paper addresses this problem proposing an approach that consists of introducing a velocity planning stage to generate adequate time sequences to be used in the interpolating curve planners. In this context, we generate speed profiles (linear and angular) that lead to trajectories respecting human comfort. The need of having travel comfort in autonomous vehicle' applications, like in cybercars [4], motivated our research on the subject of this paper.

Additionally, trajectory tracking control strategies are here proposed using sliding mode control (SMC) techniques.

Trajectory tracking control means tracking reference trajectories predefined or given by path planners. It has been widely studied and various effective methods and tracking controllers have been developed (e.g. [1,2,5, 11,13]). The model-based tracking control approach-

*Corresponding author.

ISSN 1069-2509/06/$17.00

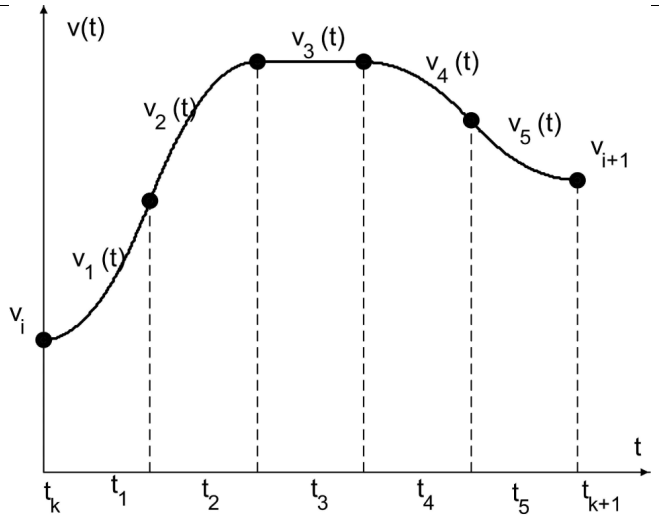

Fig. 1. A possible velocity planning for one path segment.

es can be divided into kinematic and dynamic based methods.

According to different control theories, a more elaborate category in the sense of kinematic methods can be grouped in five sets [13]: (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 kinematic-based methods, considering the stability of tracking control laws, the hardware computation load and the maneuverability in practice, tracking control law designed by sliding mode technology has been proved one of the best solutions.

Sliding mode control (SMC) is a special discontinuous control technique applicable to various practical systems [12]. The main advantages of using SMC include fast response, good transient and robustness with respect to system uncertainties and external disturbances. Therefore, it is attractive for many highly nonlinear uncertain systems [10].

Aguilar et al. [1] determined a variable structure control with sliding mode to stabilize the vehicle's motion around the reference path (path-following control). Yang and Kim [11] proposed a sliding mode control law for solving trajectory tracking problems of nonholonomic mobile robots in polar coordinates. A new SMC kinematic controller defined in polar coordinates, for trajectory-tracking in the context of wheeled mobile robots, is described in [2]. In the proposed method, two controllers are designed to asymptotically stabilize the tracking errors in position and heading direction, respectively.

2. Trajectory planning

Trajectory planning for passenger's transport vehicles must generate smooth trajectories with low associated accelerations and jerk. Lateral and longitudinal accelerations depend on the linear speed:

$$a_T = \frac{dv}{dt} \tag{1}$$

$$a_L = \frac{d\theta}{dt} \cdot v = k \cdot v^2 \tag{2}$$

thus, the trajectory planning must perform not only the planning of the curve (spatial dimension) but also the speed profile (temporal dimension).

The ISO 2631-1 standard [8] (Table 1) relates comfort with the overall r.m.s. acceleration, acting on the human body, defined as:

$$a_w = \sqrt{k_x^2 \cdot a_{wx}^2 + k_y^2 \cdot a_{wy}^2 + k_z^2 \cdot a_{wz}^2} \tag{3}$$

where a_{wx}, a_{wy}, a_{wz}, are the r.m.s. accelerations on x, y, z axes respectively, and k_x, k_y, k_z, are multiplying factors. For a seated person $k_x = k_y = 1.4$, $k_z = 1$. For motion on the xy-plane, $a_{wz} = 0$. The vehicle frame is chosen so that the $x-$axis and $y-$axis are aligned with the longitudinal and lateral directions of the trajectory, respectively.

Using Table 1 and Eq. (3), for "not uncomfortable" accelerations, the longitudinal and lateral r.m.s. accelerations must be less than 0.24 m/s^2. This value allows defining the reference speeds, and consequently the time values. The reference velocity at the final point

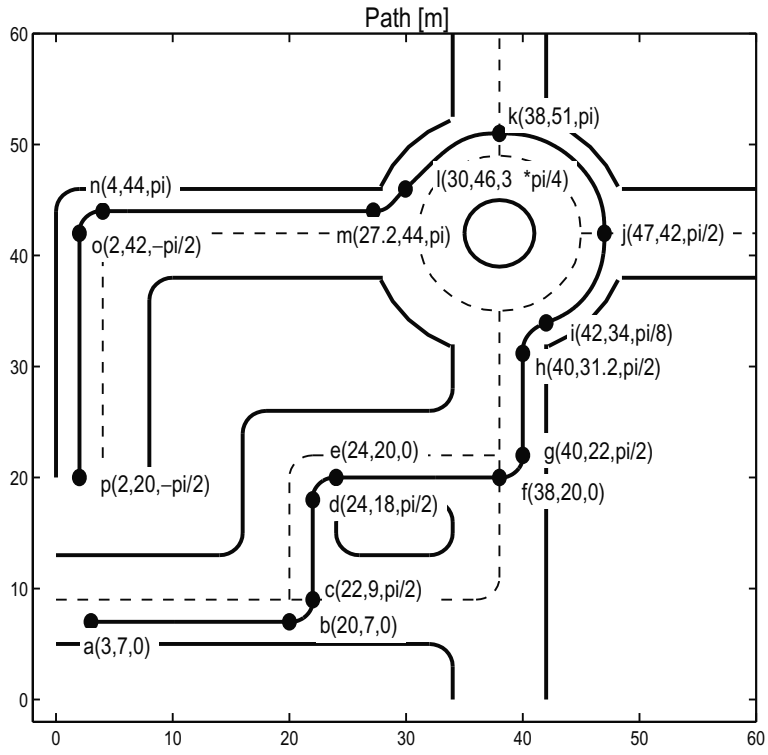

Fig. 2. Path planning example using quintic polynomial curves.

Table 1
ISO 2631-1 STANDARD

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

of a segment k (defined by two consecutive waypoints i and $i+1$), can be calculated as

$$v_{i+1} = v_i \pm a_{wx} \cdot t_k,\ a_{wx} \leqslant 0.24\ \mathrm{m/s^2} \tag{4}$$

$$t_k = t_{i+1} - t_i = \sqrt{\frac{2 \cdot s_k}{a_{wx}}} \tag{5}$$

where s_k is the length of the segment. A path between the initial and the final point is formed as a series of path segments (each segment connects two consecutive waypoints). We choose for Eq. (5) the maximum value of a_{wx} ($a_{wx} = 0.24$ m/s^2) to obtain a minimal value for t_k.

The problem: Given a set of waypoints, find control inputs $v(t)$, $\omega(t)$ such that the vehicle starting from an arbitrary initial extended state:

$p_a = [x_a\ y_a]^T = [x(0)\ y(0)]^T;\ \theta_a = \theta(0);$
$v_a = v(0); \dot{v}_a = \dot{v}(0); \omega_a = \omega(0); \dot{\omega}_a = \dot{\omega}(0);$

reaches the arbitrary final extended state:

$p_w = [x_w\ y_w]^T = [x(t_{fin})\ y(t_{fin})]^T;\ \theta_w = \theta(t_{fin});$
$v_w = v(t_{fin}); \dot{v}_w = \dot{v}(t_{fin}); \omega_w = \omega(t_{fin}); \dot{\omega}_w = \dot{\omega}(t_{fin}).$

crossing all the given waypoints. The comfort of human body constraint

$$a_w < 0.4\ \mathrm{m/s^2} \tag{6}$$

is to be satisfied.

The algorithm:

– Step 1: Determine a path connecting p_a (start point) with p_w (final point)

$$p_k(u) = \begin{bmatrix} x_k(u) \\ y_k(u) \end{bmatrix} = \begin{bmatrix} c_{0k} + c_{1k} \cdot u + c_{2k} \cdot u^2 + \dots \\ d_{0k} + d_{1k} \cdot u + d_{2k} \cdot u^2 + \dots \end{bmatrix} \tag{7}$$

were $u \in [0,\ 1]$, and c_{ik} and d_{ik}, $i = 1, 2, \dots$ are constants to be found function of the type of the curve (e.g. cubic splines, trigonometric splines or quintic splines).

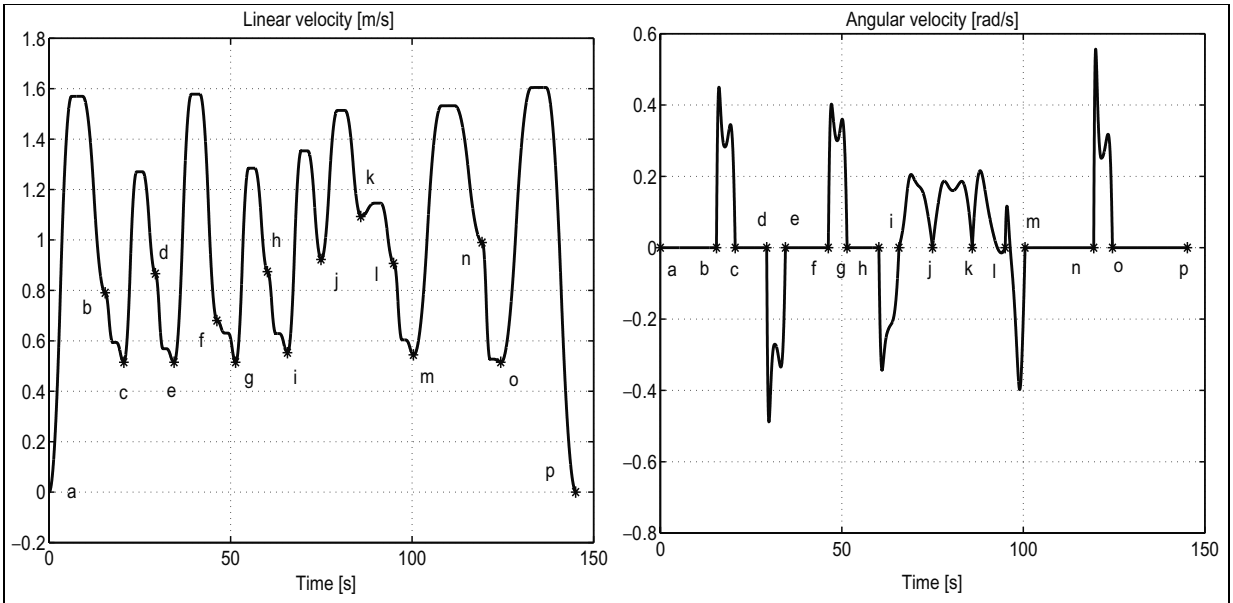

Fig. 3. Linear and angular velocities for path example.

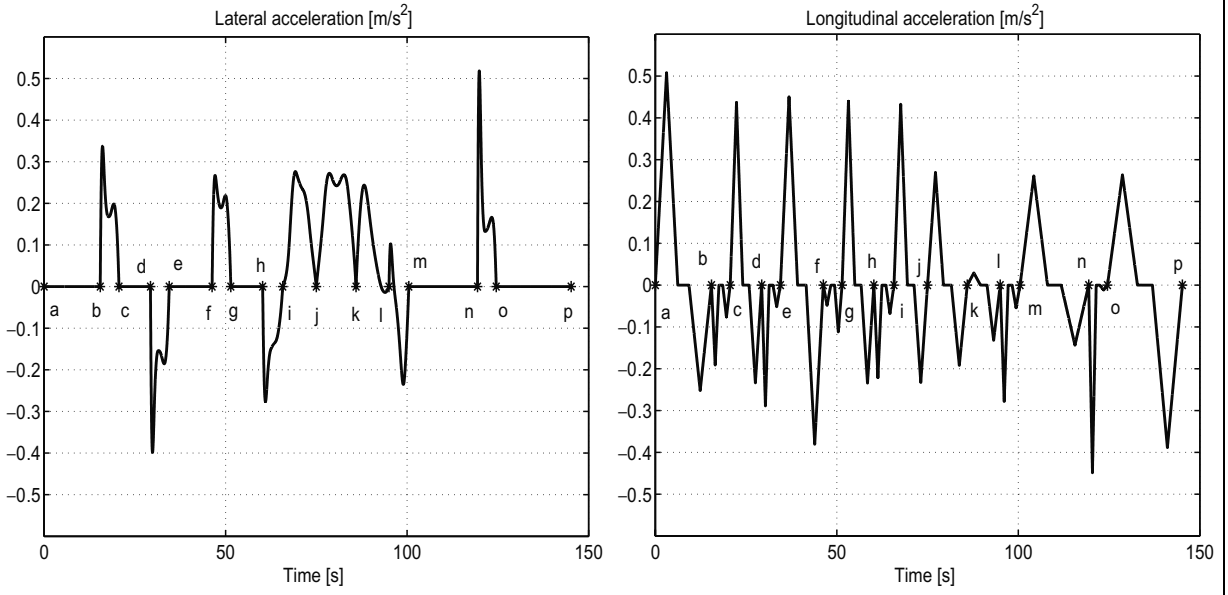

Fig. 4. Lateral and longitudinal accelerations for path example.

Determine the length s_k for each path segment,

$$s_k(u) = \int_0^1 \|\dot{p}_k(\xi)\| \cdot d\xi \tag{8}$$

and the curvature $k_{ck}(u)$ for each path segment,

$$k_{ck}(u) = \frac{\dot{x}_k(u) \cdot \ddot{y}_k(u) - \ddot{x}_k(u) \cdot \dot{y}_k(u)}{\sqrt{\dot{x}_k^2(u) + \dot{y}_k^2(u)}} \tag{9}$$

– Step 2: Each $v_k(t)$ are generated with five properly joined spline curves $(j = 1, 2, \ldots 5)$ like in [6]:

$$v_j(t) = a_{1j} + 2a_{2j} \cdot t + 3a_{3j} \cdot t^2, \quad t \in [0, t_j], \ \sum_{j=1}^{5} t_j = t_k \tag{10}$$

where coefficients a_{ij} are defined in [3]. The gen-

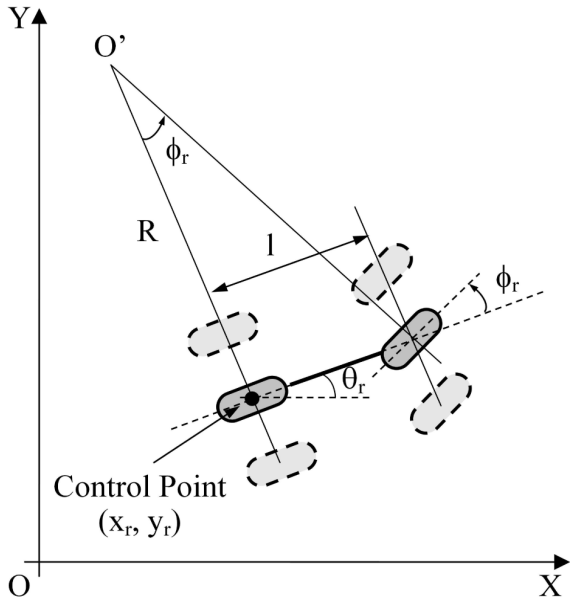

Fig. 5. Bicycle model Eq. (12).

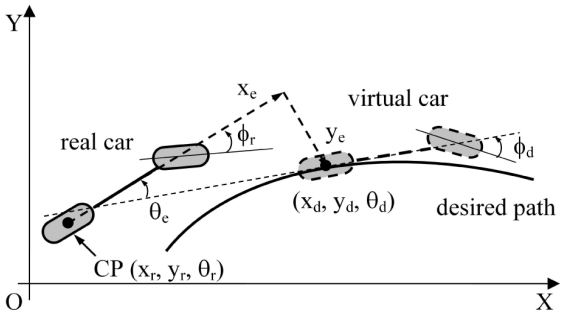

Fig. 6. Lateral, longitudinal and orientation error (trajectory- tracking).

erated velocity is C^1 and strictly positive for any $t \in [0, t_k]$. The number of curves is suggested by practical reasonings (the smoothness of the velocity profile increases with the number of curves). An example of the velocity profile (for one path segment) is shown in Fig. 1.

– Step 3: The angular velocity functions $\omega_k(t) \in C^1$ are defined according to:

$$\omega_k(t) := v_k(t) \cdot k_k(s_k)|_{s_k(t)=\int_0^t v_k(\xi)d\xi} \quad (11)$$

where $k_k(s_k)$ is the curvature expressed as a function of the arc length $s_k(t)$.

– Step 4: Calculate the r.m.s. acceleration a_{wk} for each path segment using Eqs (1), (2) and (3).

Table 2
Trajectory planning algorithm

1. calculate $x_k(u)$ and $y_k(u)$ for each path segment (Eq. 7).
2. calculate the length $s_k(u)$ for each path segment (Eq. 8).
3. calculate the curvature for each path segment (Eq. 9).
4. calculate the time t_k for each path segment (Eq. 5).
5. calculate velocity at each waypoint v_i (Eq. 4).
6. repeat (for each path segment)
   (a) calculate the curvature, $k_k(s_k(t))$
   (b) calculate $v_k(t)$
   (b1) if $\min(v_k(t)) < 0$ then decrease the final velocity v_{i+1}
   and go to (b).
   (c) calculate the angular velocity $\omega_k(t)$ (Eq. 10).
   (d) find the overall rms acceleration a_{wk} (Eq. 3).
   (e) if $a_{wk} > 0.4$ m/s^2 then increase the time (t_k).
7. until $a_{wk} < 0.4$ m/s^2.

Table 3
Length, time and r.m.s. acceleration values for each path segment

no	s_k $[m]$	t_k $[s]$	a_{wx_k} $[m/s^2]$	a_{wy_k} $[m/s^2]$	a_{w_k} $[m/s^2]$
1	17.0000	15.4500	0.2071	0.0000	0.2899
2	3.19000	5.1600	0.0750	0.2025	0.3024
3	9.0000	8.6600	0.1806	0.0000	0.2528
4	3.1900	5.1600	0.1070	0.2106	0.3307
5	14.0000	11.8000	0.2152	0.0000	0.3013
6	3.1900	5.1600	0.0444	0.1978	0.2838
7	9.1800	8.7500	0.1818	0.0000	0.2545
8	3.6700	5.5300	0.0846	0.1593	0.2525
9	10.2200	9.2300	0.1795	0.1816	0.3575
10	14.3500	10.9400	0.1210	0.2126	0.3424
11	9.8700	9.0700	0.0493	0.1344	0.2004
12	3.5600	5.4500	0.1034	0.1320	0.2348
13	23.1800	18.9000	0.1089	0.0000	0.1524
14	3.1900	5.1600	0.1636	0.2326	0.3982
15	22.0000	20.6400	0.1715	0.0000	0.2401
Total	r.m.s	acceler.	0.1537	0.1209	0.2737

$s_{fin} = 148.7900$ m.
$t_{fin} = 145.0600$ s.

IF $a_{wk} > 0.4$ m/s^2 THEN change the time for that segment and go to step 2.

If the r.m.s. acceleration is not under 0.4 m/s^2 for a given path segment the time for that segment is increased (an increase of 10% has been used). Using the proposed algorithm (summarized in Table 2) a path, respecting comfort of human body, is obtained.

Consider the example depicted in Fig. 2 where the larger circles represent waypoints $(a, b, \ldots, p)$. Each waypoint is defined by a position, in meters, and an orientation, in radians.

Figures 2–4 show results of the application of the proposed trajectory planning algorithm satisfying the comfort condition Eq. (6). From Fig. 4 we observe that the maximum absolute values for lateral and longitudinal accelerations are both below 0.51 m/s^2.

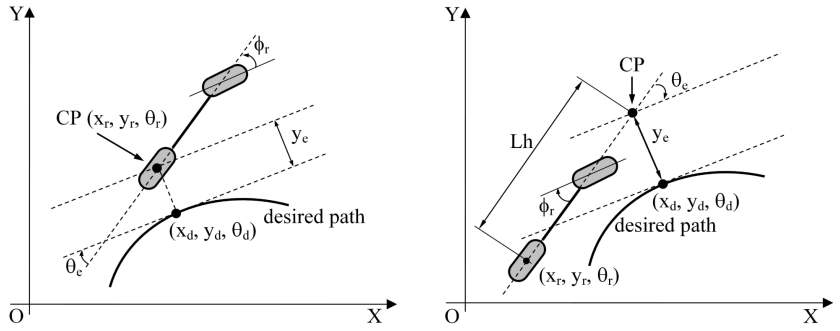

Fig. 7. Lateral and orientation errors for path-following: left) without look-ahead distance; rigth) with look-ahead distance.

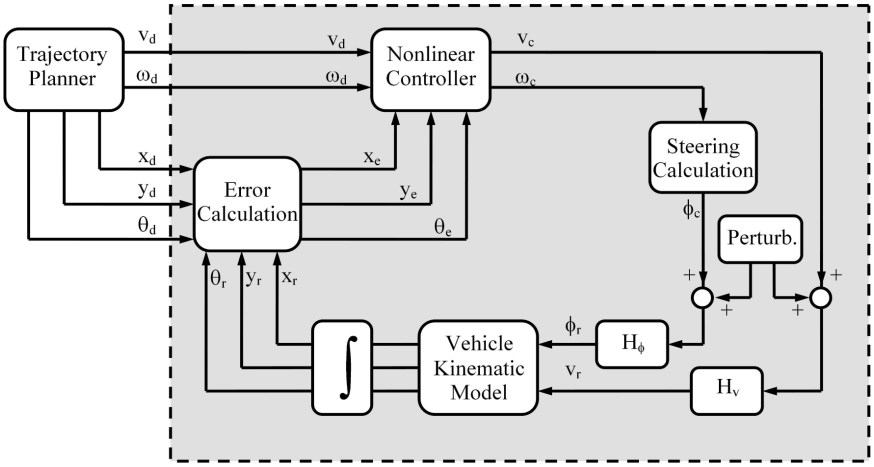

Fig. 8. Simulation model block diagram.

Table 3 summarizes the results of the applied trajectory planning algorithm: length, time and r.m.s. acceleration values for each path segment. Three acceleration components are considered: lateral, longitudinal and overall acceleration. In accordance with the 6th column of Table 3, a_{wk} satisfies Eq. (6), i.e. $a_w < 0.4\ \mathrm{m/s}^2$. This value is in the range of "A little uncomfortable" (see Table 1).

3. Sliding mode control design

The research in the area of Sliding Mode Control (SMC) was initiated about 50 years ago. In the last two decades this control methodology has been receiving great attention from the international control community with widespread use in many applications.

The scope of sliding mode control studies embraces:

– mathematical methods of discontinuous differential equations;
– design of manifolds in the state space and discontinuous control functions enforcing motions along the manifolds;
– implementation of sliding mode controllers and their applications to control of dynamic plants.

The first problem concerns development of the tools to derive the equations governing sliding modes and the conditions for this motion to exist. Formally motion equations of SMC do not satisfy the conventional uniqueness-existence theorems of the ordinary differential equations theory. The sliding mode existence problem is studied in terms of the stability theory. Sliding mode control algorithms are efficient when controlling nonlinear dynamic plants of high dimension operating under uncertainty conditions.

The main requirement in the design is that the control should satisfy the reaching condition, which in turn guarantees the existence of sliding mode on the switching manifold. Additional requirements include fast reaching and low chattering.

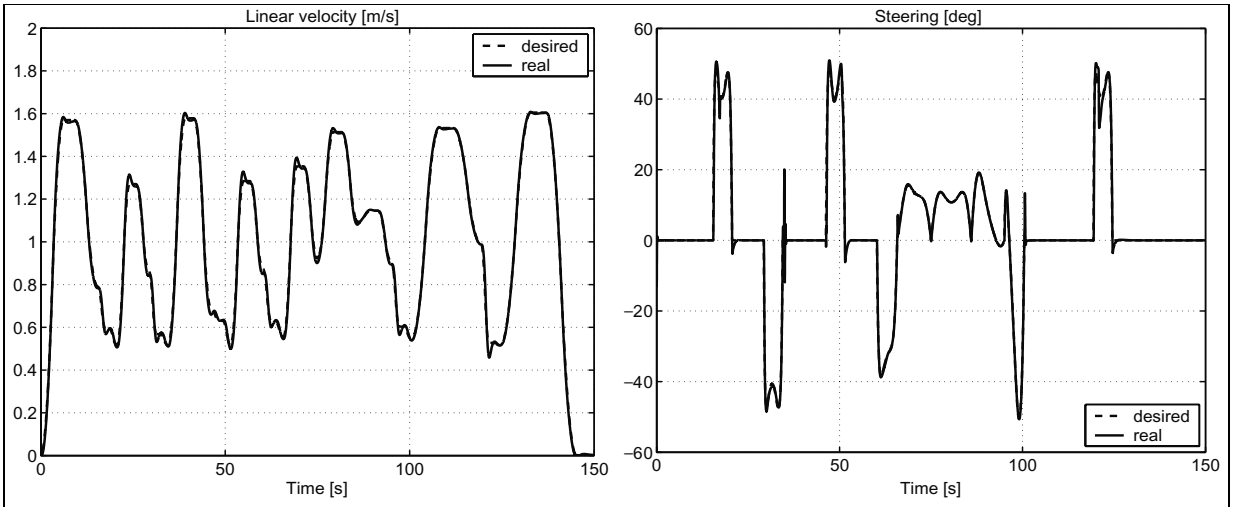

Fig. 9. Linear velocity and steering angle for trajectory-tracking controller without initial pose error.

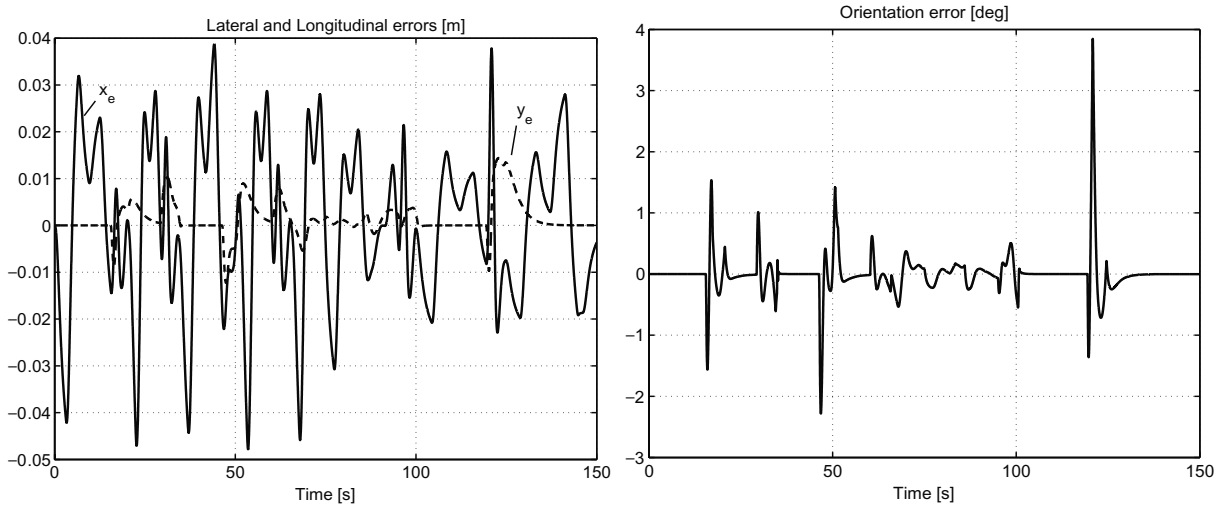

Fig. 10. Lateral, longitudinal and orientation errors for trajectory-tracking controller without initial pose error.

Lyapunov method is usually used to determine the stability properties of an equilibrium point without solving the state equation. Let $V(x)$ be a continuously differentiable scalar function defined in a domain D that contains the origin. A function $V(x)$ is said to be positive definite if $V(0) = 0$ and $V(x) > 0$ for $x \in D$ and $x \neq 0$. A generalized Lyapunov function, that characterizes the motion of the state trajectory to the sliding surface, is defined in terms of the surface. For each chosen switched control structure, one chooses the "gains" so that the derivative of this Lyapunov function is negative definite, thus guaranteeing motion of the state trajectory to the surface.

3.1. Sliding mode trajectory-tracking control

It is supposed that a feasible desired trajectory for the vehicle is pre-specified by a trajectory planner. The problem is to design a robust controller so that the vehicle will correctly track the desired trajectory under a large class of disturbances.

We consider as a motion model of a vehicle the following simplified nonholonomic system:

$$\begin{cases} \dot{x}_r(t) = v_r(t) \cdot \cos\theta_r(t) \\ \dot{y}_r(t) = v_r(t) \cdot \sin\theta_r(t) \\ \dot{\theta}_r(t) = \frac{v_r}{l} \cdot \tan\phi_r(t) \end{cases} \tag{12}$$

where (see Fig. 5) x_r and y_r are the Cartesian coordinates of the rear axle midpoint, v_r is the velocity of

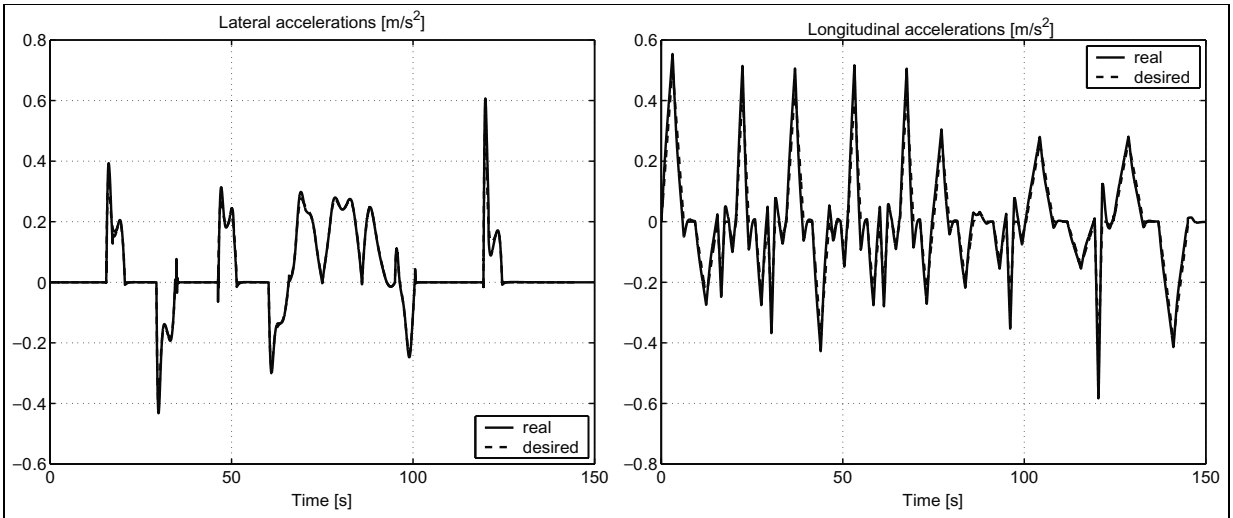

Fig. 11. Lateral and longitudinal accelerations for trajectory-tracking controller without initial pose error.

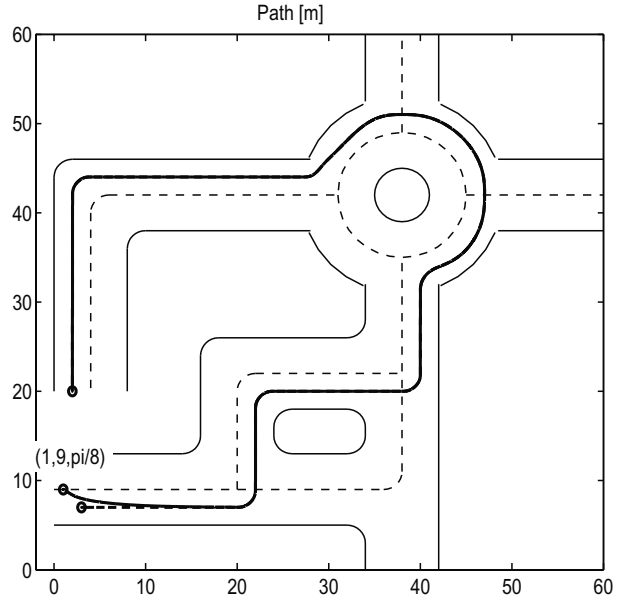

Fig. 12. xy-plot of the vehicle path for trajectory-tracking controller with initial pose error ($x_r(0) = 1, y_r(0) = 9, \theta_r(0) = \pi/8$).

this midpoint, θ_r is the vehicle's heading angle, l is the inter-axle distance, and ϕ_r the front wheel angle, which is the control variable to steer the vehicle. In this paper, the kinematic bicycle model is considered (see Figs 5–7). The trajectory tracking errors can be described by (x_e, y_e, θ_e). The aim is to design a stable controller with commands (v_c, ϕ_c).

The error vector for trajectory-tracking is easily obtained from Figs 5 and 6,

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_d & \sin\theta_d & 0 \\ -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \quad (13)$$

where (x_d, y_d, θ_d) denotes the virtual car pose. The corresponding error derivatives are

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot \cos\theta_e + y_e \cdot \frac{v_d}{l} \cdot \tan\phi_d \\ \dot{y}_e = v_r \cdot \sin\theta_e - x_e \cdot \frac{v_d}{l} \cdot \tan\phi_d \\ \dot{\theta}_e = \frac{v_r}{l} \cdot \tan\phi_r - \frac{v_d}{l} \cdot \tan\phi_d \end{cases} \quad (14)$$

where v_d and ϕ_d are the heading speed and desired front wheel angle, respectively.

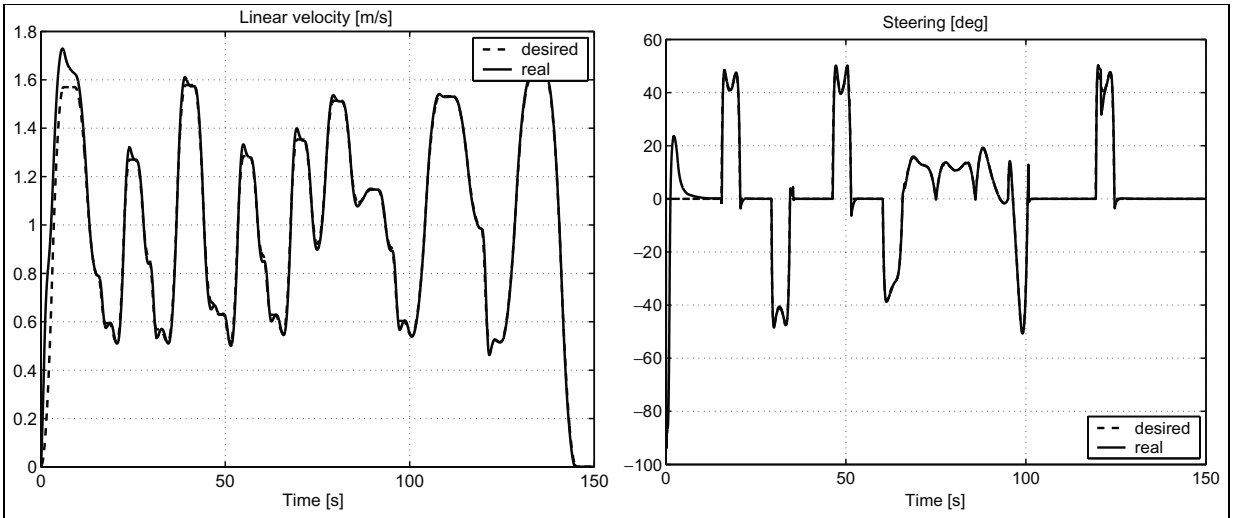

Fig. 13. Linear velocity and steering angle for trajectory-tracking controller with initial pose error.

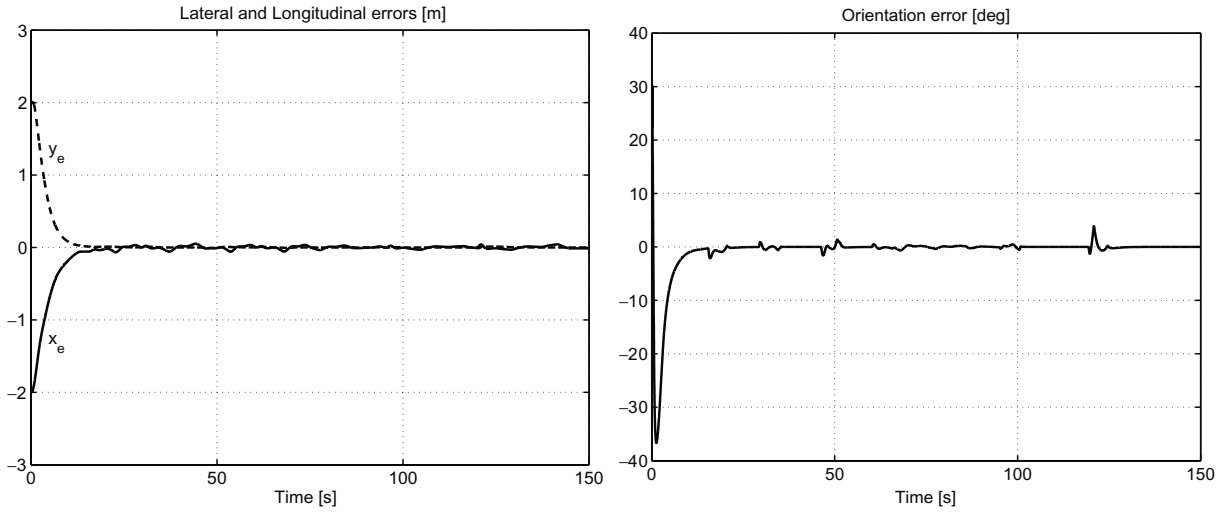

Fig. 14. Lateral, longitudinal and orientation errors for trajectory-tracking controller with initial pose error.

In this paper it is assumed that $|\theta_e| < \pi/2$, which means that the vehicle orientation must not be perpendicular to the desired trajectory.

The typical sliding surface (s) is [10]:

$$s = \dot{\tilde{x}} + k \cdot \tilde{x} \tag{15}$$

where

$$\tilde{x} = x - x_{\text{desired}} \tag{16}$$

In case of ideal sliding mode:

$$s = \dot{\tilde{x}} + k \cdot \tilde{x} = 0, \quad \dot{s} = \ddot{\tilde{x}} + k \cdot \dot{\tilde{x}} = 0 \tag{17}$$

In other words it means that the state trajectories of the controlled plant satisfies the equation $s = 0$. For all practical purposes the existence condition is mathematically expressed as:

$$s \cdot \dot{s} < 0 \tag{18}$$

In physical sense it means that sliding mode exists if in the vicinity of the switching line $s = 0$, the tangent or the velocity vector of the state trajectory points towards the switching line.

A new design of sliding surface is proposed, such that lateral error, y_e, and angular error, θ_e, are internally coupled with each other in a sliding surface leading to convergence of both variables. For that purpose the following sliding surfaces are proposed:

$$s_1 = \dot{x}_e + k_1 \cdot x_e \tag{19}$$

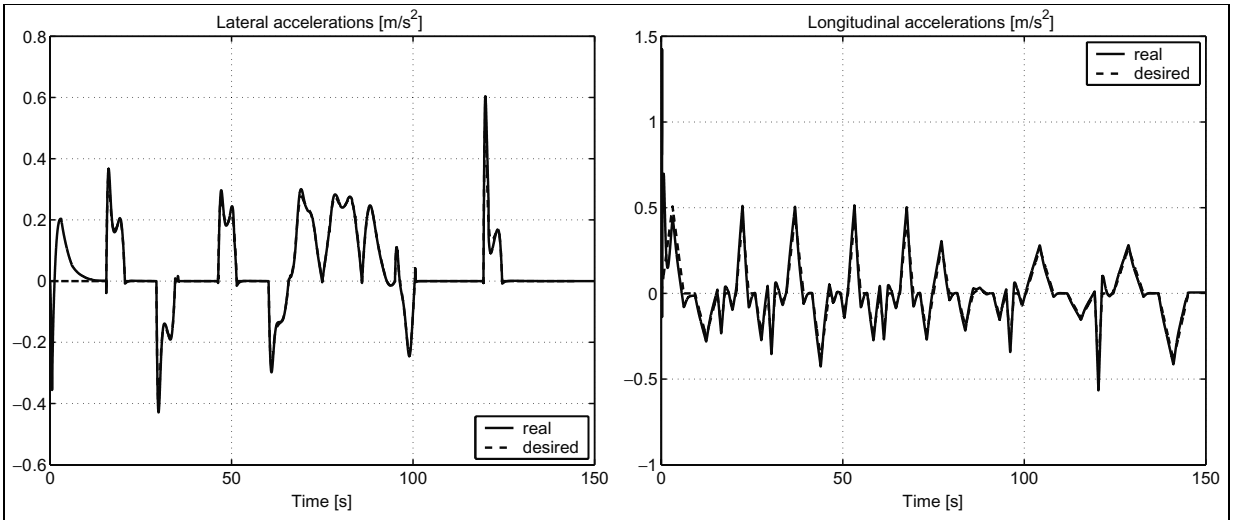

Fig. 15. Lateral and longitudinal accelerations for trajectory-tracking controller with initial pose error.

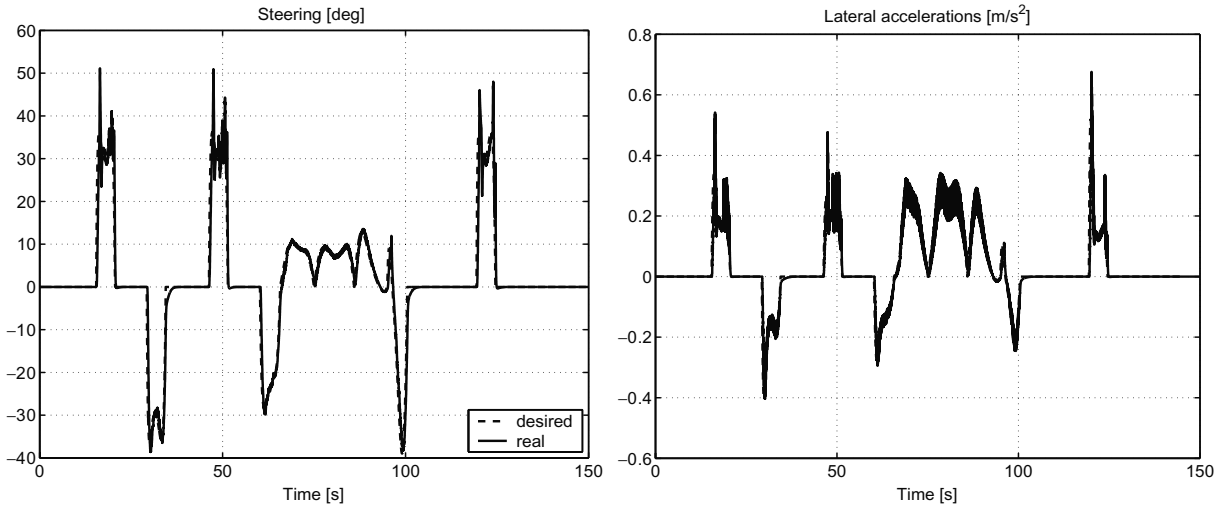

Fig. 16. Steering angle and lateral acceleration for path-following without initial pose error.

$$s_2 = \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \theta_e \tag{20}$$

In trajectory-tracking exist three variables (x_e, y_e, θ_e) and just two control variables, which implies that we have only two sliding surfaces. We chose to couple y_e and θ_e in one sliding surface. The condition under which the state will move toward and reach a sliding surface is called a reaching condition. A practical general form of the reaching law is

$$\dot{s} = -Q \cdot g(s) - P \cdot \mathrm{sgn}(s) \tag{21}$$

where

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \quad q_i > 0, \quad P = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix},$$

$$p_i > 0, \quad i = 1, 2.$$

$$\mathrm{sgn}(s) = \begin{bmatrix} \mathrm{sgn}(s_1) \\ \mathrm{sgn}(s_2) \end{bmatrix}, \quad g(s) = \begin{bmatrix} g_1(s_1) \\ g_2(s_2) \end{bmatrix},$$

$$s_i \cdot g_i(s_i) > 0, \;\; g_i(0) = 0$$

Various choices of Q and P specify different rates for s and yield different structures in the reaching law.

Three practical special cases of Eq. (21) are given below:

1. *constant rate reaching*

$$\dot{s} = -P \cdot \mathrm{sgn}(s) \tag{22}$$

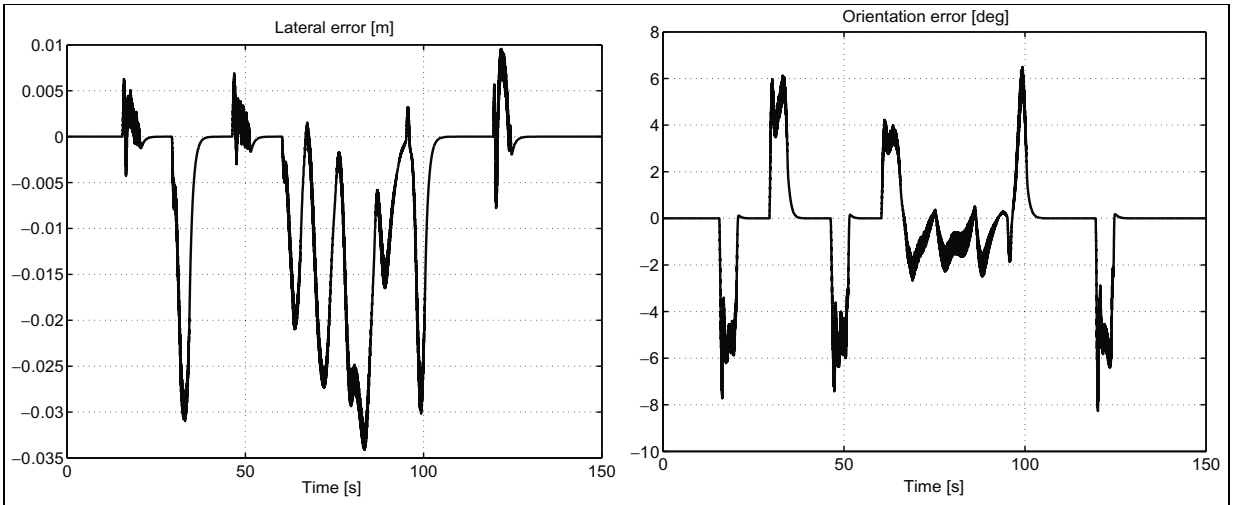

Fig. 17. Lateral and orientation errors for path-following without initial pose error.

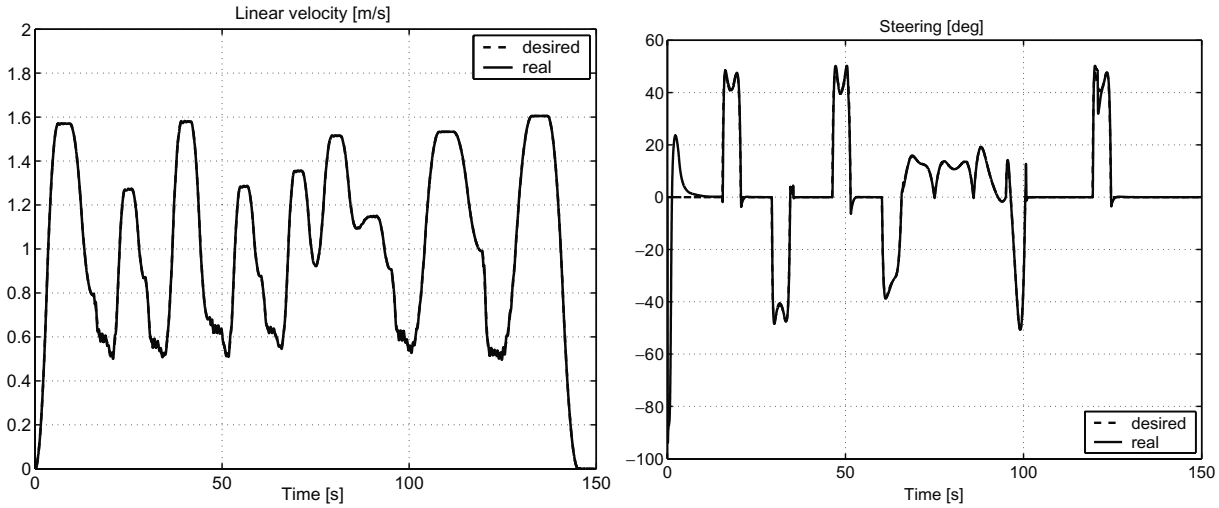

Fig. 18. Linear velocity and steering angle for Chwa trajectory-tracking controller (see [2]).

This law forces the switching variable s to reach the switching manifold at a constant rate $|\dot{s}_i| = -p_i$, $i = 1, 2$.

2. *constant plus proportional rate reaching*

$$\dot{s} = -Q \cdot s - P \cdot \mathrm{sgn}(s) \tag{23}$$

By adding the proportional rate term $-Q \cdot s$, the state is forced to approach the switching manifolds faster when s is large.

3. *power rate reaching*

$$\dot{s} = -Q \cdot |s|^{\alpha} \cdot \mathrm{sgn}(s), 0 < \alpha < 1. \tag{24}$$

This reaching law 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.

In this work the second reaching law Eq. (23) was selected. From the time derivative of Eqs (19) and (20) and using (23), results

$$\dot{s}_1 = \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot s_1 - p_1 \cdot \mathrm{sgn}(s_1) \tag{25}$$

$$\begin{aligned} \dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \dot{\theta}_e \\ &= -q_2 \cdot s_2 - p_2 \cdot \mathrm{sgn}(s_2) \end{aligned} \tag{26}$$

From Eqs (14), (25) and (26), and after some mathematical manipulation, we get the commands:

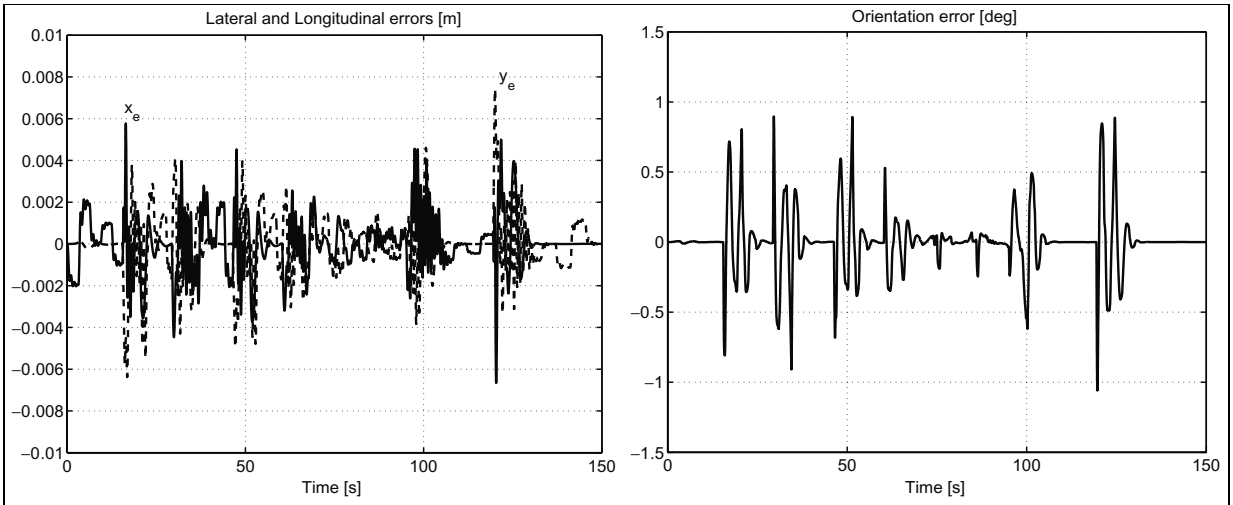

Fig. 19. Lateral and orientation errors for Chwa trajectory-tracking controller (see [2]).

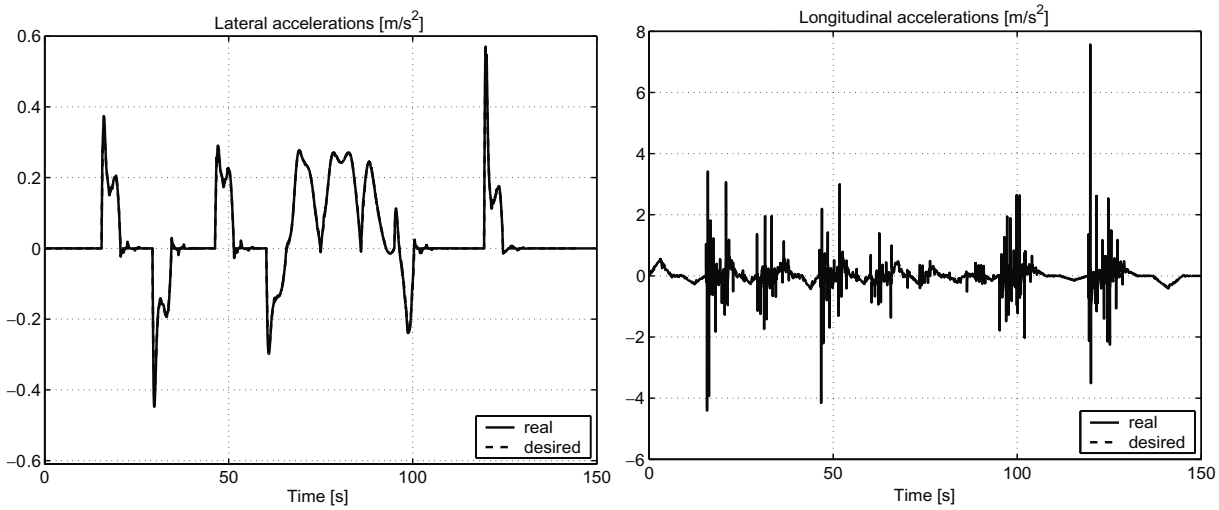

Fig. 20. Lateral and longitudinal accelerations for Chwa trajectory-tracking controller (see [2]).

$$\begin{aligned}\dot{v}_c = \frac{1}{\cos\theta_e}\cdot(&-q_1\cdot s_1 - p_1\cdot\mathrm{sgn}(s_1)\\ &-k_1\cdot\dot{x}_e - \dot{\omega}_d\cdot y_e - \omega_d\cdot\dot{y}_e\\ &+v_r\cdot\dot{\theta}_e\cdot\sin\theta_e + \dot{v}_d)\end{aligned} \tag{27}$$

$$\begin{aligned}\phi_c = \arctan\Bigg(&\frac{l}{v_r}\cdot\omega_d + \frac{l}{v_r\cdot(v_r\cdot\cos\theta_e + k_0\cdot\mathrm{sgn}(y_e))}\\ &\cdot(-q_2 s_2 - p_2\mathrm{sgn}(s_2) - k_2\cdot\dot{y}_e\\ &-\dot{v}_r\cdot\sin\theta_e + \dot{\omega}_d\cdot x_e + \omega_d\cdot\dot{x}_e)\Bigg)\end{aligned} \tag{28}$$

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

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

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

3.2. Sliding mode path-following control

In path-following, the control objective is to ensure that the vehicle will correctly follow the reference path. For this purpose, both the lateral error, y_e, and the orien-

tation error, θ_e, must be minimized. It is supposed that a feasible desired path for the vehicle is pre-specified by a trajectory planner.

For the path-following without look-ahead ($Lh = 0$) (see Fig. 7) the error vector is:

$$\begin{bmatrix} y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \quad (29)$$

The lateral error, y_e, is defined as the distance between the vehicle control point (CP) and the closest point along the desired trajectory. The corresponding error derivatives are:

$$\begin{cases} \dot{y}_e = v_r \cdot \sin\theta_e \\ \dot{\theta}_e = \dot{\theta}_r = \frac{v_r}{l} \cdot \tan\phi_r \end{cases} \quad (30)$$

Defining the control point CP (see Fig. 7) at a distance $Lh \neq 0$ in front of the vehicle (called look-ahead distance), Eq. (29) becomes:

$$\begin{bmatrix} y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} -\sin\theta_d & \cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d + Lh \cdot \cos\theta_r \\ y_r - y_d + Lh \cdot \sin\theta_r \\ \theta_r - \theta_d \end{bmatrix}$$

and

$$\begin{cases} \dot{y}_e = v_r \cdot \sin\theta_e + Lh \cdot \frac{v_r}{l} \cdot \tan\phi_r \cdot \cos\theta_e \\ \dot{\theta}_e = \dot{\theta}_r = \frac{v_r}{l} \cdot \tan\phi_r \end{cases} \quad (31)$$

In this paper it is assumed that $|\theta_e| < \pi/2$, which means that the vehicle orientation must not be perpendicular to the desired trajectory.

We propose a new design of sliding surface such that lateral error, y_e, and angular variable, θ_e are internally coupled with each other in a sliding surface leading to convergence of both variables. For that purpose the following sliding surface is proposed:

$$s = \dot{y}_e + k \cdot y_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \theta_e \quad (32)$$

In path-following exist two variables (y_e, θ_e) and just one control variable, which implies that we have only one sliding surface. We chose to couple y_e and θ_e in one sliding surface.

By choosing the second reaching law Eq. (23),

$$\begin{aligned} \dot{s} &= \ddot{y}_e + k \cdot \dot{y}_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \\ \dot{\theta}_e &= -Q \cdot s - P \cdot \mathrm{sgn}(s) \end{aligned} \quad (33)$$

From Eqs (31) and (33), the steering control law is obtained as

$$\phi_c = \arctan\left(\frac{l}{v_r} \cdot \frac{-Qs - P\mathrm{sgn}(s) - kv_r \sin\theta_e}{v_r \cdot \cos\theta_e + k_0 \cdot \mathrm{sgn}(y_e)}\right) \quad (34)$$

For the case of using look-ahead Lh:

$$\begin{aligned} \dot{\phi}_c = \frac{l \cdot \cos^2(\phi_r)}{v_r \cdot Lh \cdot \cos\theta_e} \cdot (&-Q \cdot s - P \cdot \mathrm{sgn}(s) \\ &-k \cdot \dot{y}_e - v_r \cdot \dot{\theta}_e \cdot \cos\theta_e + +Lh \cdot \dot{\theta}_e^2 \\ &\cdot \sin\theta_e - k_0 \cdot \mathrm{sgn}(y_e) \cdot \dot{\theta}_e) \end{aligned} \quad (35)$$

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

$$\begin{aligned} \dot{V} = &= s \cdot \dot{s} \\ &= s \cdot (\ddot{y}_e + k \cdot \dot{y}_e + k_0 \cdot \mathrm{sgn}(y_e) \cdot \dot{\theta}_e) \\ &= s \cdot (-Q \cdot s - P \cdot \mathrm{sgn}(s)) \\ &= -Q \cdot s^2 - P \cdot \frac{s^2}{|s|} \end{aligned}$$

For $\dot{V}$ to be negative semi-definite, it is sufficient to choose Q and P such that $Q, P \geqslant 0$.

4. Simulation results

In this section, simulation results of the proposed methods are presented. The simulation model block diagram is shown in Fig. 8. The following transfer functions were employed:

$$\begin{aligned} H_\phi &= \frac{\phi_r}{\phi_c} = \frac{\omega_n^2}{s^2 + 2 \cdot D \cdot \omega_n \cdot s + \omega_n^2}; \\ H_v &= \frac{v_r}{v_c} = \frac{1}{0.25 \cdot s + 1} \end{aligned} \quad (36)$$

with $D = 0.7$ and $\omega_n = 2\pi \cdot 5$ Hz.

The reference trajectory was generated using the trajectory planner, described in Section 1 (we considered the example illustrated in Fig. 2). Design parameters of Eqs (19), (20), (25), (26), (32) and (33) are $k_0 = 0.05$, $k_1 = 0.25$, $k = k_2 = 0.5$, $p_1 = 1$, $p_2 = 1$, $q_1 = 1$ and $q_2 = 1$. The signum functions in the control inputs Eqs (25), (26) and (33) and sliding surfaces Eqs (25), (26) and (33) were replaced by saturation functions with ± 0.5 thresholds, to reduce the chattering phenomenon. Also, input disturbances were chosen to be Gaussian random noise with zero mean and variance 0.05. The look-ahead distance for path-following control is $Lh = 1.5$ m.

Figures 9–11 show the results for trajectory-tracking control without initial pose error where it is shown that the achieved linear velocity follows the reference velocity, both accelerations (lateral and longitudinal) follow the reference accelerations, and the tracking errors converge to zero.

Table 4
Simulation results for the trajectory-tracking and path-following controllers

		A	B	C	D	E	F	G
Max. Longit. Error	$[m]$	0.0478	2.0000	0.0030	–	–	–	0.0067
r.m.s. Longit. Error	$[m]$	0.0173	0.2694	0.0009	–	–	–	0.0013
Max. Lateral Error	$[m]$	0.0144	2.0063	0.0628	0.0341	0.0771	0.0435	0.0074
r.m.s. Lateral Error	$[m]$	0.0039	0.2656	0.0206	0.0099	0.0240	0.0954	0.0014
Max. Angular Error	$[rad]$	0.0671	0.6407	0.0954	0.1443	0.2032	0.1378	0.0185
r.m.s. Angular Error	$[rad]$	0.0063	0.0763	0.0152	0.0360	0.0619	0.0350	0.0037
Max. Lateral Accel.	[m/s^2]	0.6071	0.6044	0.6071	0.6758	0.6873	0.3493	0.5704
r.m.s. Lateral Accel.	[m/s^2]	0.1242	0.1274	0.1289	0.1185	0.1545	0.1197	0.1219
Max. Longit. Accel.	[m/s^2]	0.5541	1.4226	0.5095	0.5081	0.5081	0.5081	7.5701
r.m.s. Longit. Accel.	[m/s^2]	0.1662	0.1694	0.1514	0.1537	0.1537	0.1537	0.2303
Overal r.m.s. Accel.	[m/s^2]	0.2904	0.2968	0.2784	0.2716	0.3050	0.2727	0.3648

Figures 12–15, for trajectory-tracking control with initial pose error ($x_r(0) = 1, y_r(0) = 9, \theta_r(0) = \pi/8$), show that the car retrieve the initial difference ($x_e = -2, y_e = 2$) quickly ($\Delta t \approx 20$ s), and that the tracking errors converge to zero. The trajectory-tracking task must perform not only the tracking of the path segment (spatial dimension) but also doing it following a specified speed profile (temporal dimension).

Figures 16–17 show that path-following performance for reference trajectory (see Fig. 2) is satisfactory. The lateral and orientation errors also converge to zero. In the path-following task, the lateral error, y_e, is defined as the distance between the vehicle control point and the closest point along the vehicle desired trajectory as illustrated in Fig. 7.

Table 4 summarizes results of simulations for the following cases:

1. *Trajectory-tracking* A. without initial pose error; B. with initial pose error ($x_r(0) = 1$, $y_r(0) = 9$, $\theta_r(0) = \pi/8$); C. with disturbance – Gaussian random noise with zero mean and variance 0.05.
2. *Path-following*
   D. without initial pose error;
   E. with disturbance – Gaussian random noise with zero mean and variance 0.05;
   F. with look-ahead distance ($Lh = 1.5$ m) and without initial pose error.
3. *Trajectory-tracking for Chwa controller* (*see Section 4.1*)
   G. without initial pose error.

For trajectory-tracking the lateral and orientation errors are less than the errors for path-following, but the r.m.s. accelerations (lateral and longitudinal) are worse (the trajectory-tracking controller is more "nervous"). The overall r.m.s accelerations, for all our cases (A to F), are under 0.315 m/s^2 ("not uncomfortable", see Table 1). Moreover, in all our cases r.m.s lateral and longitudinal accelerations are less than 0.24 m/s^2 (see Section 2).

Sliding mode control algorithms are efficient when controlling nonlinear dynamic plants operating under perturbations (see case C and E). Moreover, the simulation results show that good performances are still obtained even when a large initial pose error is applied.

4.1. Experimental comparison

The Chwa SMC controller [2] was simulated, for the sake of comparison with our trajectory-tracking controller, using the same simulation model block diagram shown in Fig. 8. In order to have the data of both controllers in the same coordinates, the polar tracking errors of the Chwa controller were transformed into Cartesian tracking errors.

In Figures 18–20, performance of the Chwa trajectory-tracking controller is shown. The lateral and orientation errors are, in this case, lower than in our controller (Figs 9–11), but the longitudinal acceleration is worse. The r.m.s. values concerning Chwa controller are shown in column G of Table 4, and the corresponding values, for our trajectory-tracking controller, applied in similar situation, are shown in column A. The overall r.m.s. acceleration of Chwa controller is higher than for our SMC controller (25% more, approximately). We also notice some high values (peak values) of the maximum longitudinal acceleration in the case of Chwa controller.

5. Conclusion

For autonomous vehicle, trajectory planning methods that produce smooth trajectories, with low level associated accelerations and jerk for providing human comfort, are required. This paper addresses this prob-

lem proposing a new approach that consists of introducing a velocity planning stage in the trajectory planner. Moreover, this paper describes a strategy for trajectory-tracking and path-following control of vehicles. A new design of sliding surface is proposed, such that lateral and angular errors are internally coupled with each other in a sliding surface leading to convergence of both variables. The main advantages of using SMC include fast response, good transient and robustness with respect to system uncertainties and external disturbances. Simulation results showed the effectiveness of the proposed trajectory planning algorithm.

Acknowledgements

This work was supported in part by ISR-Coimbra and FCT (Portuguese Science and Technology Foundation), under contract NCT04: POSC/EEA/SRI/58016/2004. The first author benefited a research fellowship also from FCT, contract: SFRH/BD/18211/2004.

References

[1] L.E. Aguilar, T. Hamel and P. Soueres, Robust path following control for wheeled robots via sliding mode techniques, *Intelligent Robots and Systems – IROS '97* **3** (1997), 1389–1395.

[2] D. Chwa, Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates, *IEEE Transactions on Control Systems Technology* **12** (2004), (637–644).

[3] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Reading, Massachusetts, 1986.

[4] Cybercars. Cybernetic techologies for the car in the city, [online], www.cybercars.org.

[5] R. Fierro and F.L. Lewis, *Control of a nonholonomic mobile robot: backstepping kinematics into dynamics*, 34th Conference on Decision and Control, 2005, New Orleans, 252–261.

[6] C. Guriano, Lo Bianco, A. Piazzi and M. Romano, Velocity planning for autonomous vehicles, *IEEE Intelligent Vehicles Symposium* (2004), Parma, 413–418.

[7] C. Guriano, Lo Bianco, A. Piazzi and M. Romano, Smooth motion generation for unicycle mobile robots via dynamic path inversion, *IEEE Transactions on Robotics* **15** (2004), 884–891.

[8] ISO, Mechanical vibration and shock – Evaluation of human exposure to whole body vibrations – Part 1: General requirements, *ISO 2631-1* (1997).

[9] M. Parent, G. Gallais, A. Alessandrini and T. Chanard, Cybercars: review of first projects, *Ninth Int Conf on Automated People Movers* (2003), Singapore.

[10] J.J.E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall Inter. Ed., 1991.

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

[12] K.D. Young, V.I. Utkin and U. Ozguner, A control engineer's guide to sliding mode control, *IEEE Transactions on Control Systems Technology* **7** (1999), 328–342.

[13] G.F. Yuan, Tracking control of a mobile robot using neural dynamics based approaches, *Masters Abstracts International* **39** (2001), 1437.