Modeling and Performance Evaluation for Trajectory Tracking Control of a Wheeled Mobile Robot

Razvan SOLEA∗, Daniela Cristina CERNEGA∗

∗ Control Systems and Industrial Informatics Department, Computer Science Faculty, ”Dunarea de Jos” University from Galati,80021 Str. Domneasca 111, Galati, ROMANIA (e-mail: Razvan.Solea@ugal.ro,
Daniela.Cernega@ugal.ro)

Abstract: The wheeled mobile robot is a nonlinear system. The trajectory tracking control problem is solved using the sliding mode control. In this paper four control laws are modeled and the system performances are investigated. The sliding mode control laws for the trajectory tracking problem are simulated and then implemented on the PatrolBot Robot. The performances are analyzed in order to establish some rules. The analysis conclusions are based on the simulation results and on the implementation of the control laws on the PatrolBot Robot.

Keywords: Trajectory Tracking, Nonlinear Control, Sliding Mode Control, Performance Evaluation.

1. INTRODUCTION

In this paper trajectory tracking problem is solved using four Sliding Mode Control (SMC) laws. Sliding Mode control was chose because is known to possess merits such as the invariance to parametric uncertainties as well as the capacity to reject disturbances. However, this type of control suffers from the chattering phenomenon which is due to high frequency switching over discontinuity of the control signal (Perruquetti and Barbot (2002)).

In this paper for the trajectory tracking control problem it is used a nonlinear model (Slotine and Li (1991)):

\[ x^{(n)} = f(x, t) + b(x, t) \cdot u \]  (1)

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

The design of a variable structure control (VSC)(Gao and Hung (1993)) for a nonlinear system implies two steps:

(1) "reaching mode" or nonsliding mode;
(2) sliding mode.

For the reaching mode, the desired response usually is to reach the switching manifold, \( s \), described by:

\[ s(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} \cdot \dot{x} = 0 \]  (3)

where \( \dot{x} \) is the tracking error and \( \lambda \) is a strictly positive constant which determines the closed-loop bandwidth. For example, if \( n = 2 \),

\[ s = \dot{x} + \lambda \cdot \ddot{x} \]  (4)

Hence the corresponding switching manifold is:

\[ s(t) = 0 \]  (5)

For a system having \( m \) inputs, \( m \) switching functions are needed.

It is proved that the most important virtue of the VSC systems is robustness. Properly design of the switching functions for a VSC system ensures the asymptotic stability. A number of design criteria exist for this purpose (Utkin and Young (1978)), (Dorling and Zinober (1986)).

Sliding Mode is also known to posses merits such as the invariance to parametric uncertainties. Dynamic characteristics of the reaching mode are very important, and this type of control suffers from the chattering phenomenon which is due to high frequency switching over discontinuity of the control signal. This aspect of the sliding mode control is further investigated in this paper using four different control laws.

In order to handle the chattering problem two approaches are widely referred in literature. The first one is called the continuation method because the discontinuous relay type actuator is replaced by a high - gain device with saturation (Burton and Zinober (1989)). Although this method eliminates the high frequency chattering it also destroys the sliding mode. In addition, the resulting physical system,
often exhibits low frequency oscillations due to unmodeled dynamics.

The second approach deals directly with the reaching process since chattering is caused by the nonideal reaching at the end of the reaching phase. This approach establishes the reaching mode characteristics by the use of a reaching model. The resulting method is called the reaching law method (Gao and Hung (1993)).

In this paper four different reaching laws and reaching modes are investigated. The performances of the four laws in controlling the amplitude of the chattering are compared in order to establish an on-line parameters adjusting procedure.

This paper is organized as follows: Section II is dedicated to the presentation of the reaching mode and switching functions for nonlinear systems. The three stages considered for the sliding mode used for trajectory tracking problem are presented. The four reaching laws models are also listed. In Section III the control problem for the wheeled mobile robots is presented using the kinematic model. The trajectory-tracking problem is treated using the sliding mode control, and the sliding manifolds equations for the four reaching laws in Section II are presented. The commands for each controller are also obtained. Section IV is dedicated to experimental results of the implementation of the four reaching laws presented in Section II. Section V presents the conclusions of the implementation on PatrolBot Robot and future work directions.

2. REACHING MODE AND CONTROL LAW DESIGN

In this paper the model used for the controlled robot is a 2-order MIMO (Multiply Input Multiply Output) nonlinear system that is "linear in control". The model used is:

\[ \ddot{x} = f(x, \dot{x}, t) + B(x, \dot{x}, t) \cdot u \] (6)

where \( x = [x_1, x_2, \ldots, x_n] \), \( x \in \mathbb{R}^n \), \( f \) is a vector of nonlinear functions, \( f \in L_2^0 \), \( B \) is a matrix of gains, \( B \in \mathbb{R}^{n \times n}; \det(B) \neq 0 \); \( u \) is the control vector, \( u \in \mathbb{R}^m \). The control law is:

\[ u = p(x, \dot{x}, t) \] (7)

For the 2nd-order MIMO nonlinear system having the model shown in (7) efficient sliding mode control can be achieved via the following stages (see Fig. 1):

1. reaching phase motion: during this stage the trajectory is attracted towards the switching manifold (if the reaching condition is satisfied); characterised by
   \[ s_i \neq 0, \ \dot{x}_i \neq 0, \ \ddot{x}_i \neq 0 \] (8)
2. sliding mode motion: during this stage the trajectory stays on the switching manifold, i.e.,
   \[ s_i = 0, \ \dot{x}_i \neq 0, \ \ddot{x}_i \neq 0 \] (9)
3. steady state; during this stage both the state variable and the state velocity will converge to the steady state value, therefore:
   \[ s_i = 0, \text{ and } \begin{cases} \ddot{x}_i \to 0, \ \dot{x}_i \to 0 \\ \ddot{x}_i = 0, \ \dot{x}_i = 0 \end{cases} \] (10)

The four practical cases of the equation (11) used in this paper are given below.

A. Constant rate reaching (Gao and Hung (1993))

\[ \dot{s} = -Q \cdot sgn(s) - P \cdot h(s) \] (11)

where

\[ Q = diag[q_1, q_2, \ldots, q_n], \ q_i > 0, \ i = 1, 2, \ldots, n \]
\[ P = diag[p_1, p_2, \ldots, p_n], \ p_i > 0, \ i = 1, 2, \ldots, n \]

and

\[ sgn(s) = [sgn(s_1), sgn(s_2), \ldots, sgn(s_n)]^T \]
\[ h(s) = [h_1(s_1), h_2(s_2), \ldots, h_n(s_n)]^T \]
\[ s_i \cdot h_i(s) > 0, \ h_i(0) = 0 \]

2.1 Reaching Laws

The four practical cases of the equation (11) used in this paper are given below.

B. Constant plus proportional rate reaching (Gao and Hung (1993))

\[ \dot{s} = -Q \cdot sgn(s) - P \cdot \dot{s} \] (13)

Clearly, by adding the proportional rate term \(-P \cdot \dot{s}\), the state is forced to approach the switching manifolds faster when \( s \) is large. It can be shown that the reaching time
for $x$ to move from an initial state $x_0$ to the switching manifold $s_i$ is finite, and is given by:

$$T_i = \frac{1}{p_i \cdot t_n} \cdot \frac{p_i \cdot |s_i| + q_i}{q_i} \quad (14)$$

C. Power rate reaching (Gao and Hung (1993))

$$\dot{s}_i = -p_i \cdot |s_i|^{\alpha} \cdot sgn(s_i), \quad 0 < \alpha < 1, \quad i = 1, ..., m \quad (15)$$

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. The result is a fast reaching and low chattering reaching mode. Integrating (15) from $s_i = s_{i0}$ to $s_i = 0$ yields

$$T_i = \frac{|s_i(0)|^{1-\alpha}}{(1-\alpha) \cdot p_i} \quad (16)$$

showing that the reaching time $T_i$, is finite. Thus power rate reaching law gives a finite reaching time. In addition, because of the absence of the $-Q \cdot sgn(s)$ term on the right-hand side of (15), this reaching law eliminates the chattering.

D. Speed control rate reaching (Loh and Yeung (2004))

$$\dot{s}_i = -p_i \cdot \exp(\alpha \cdot |s_i|) \cdot sgn(s_i), \quad p_i > 0, \quad \alpha > 0, \quad i = 1, ..., m \quad (17)$$

and the reaching time $T_i$ becomes:

$$T_i = \frac{1}{\alpha \cdot p_i} \cdot (1 - \exp(-\alpha \cdot |s_i(0)|)) \quad (18)$$

The four reaching laws presented above are used in the implementation on the PatrolBot Robot system in order to analyze their performances.

3. CONTROL OF WHEELED MOBILE ROBOTS

The application of SMC strategies in nonlinear systems has received considerable attention in recent years (Chwa (2004), Yang and Kim (1999), Chwa et al. (2006), 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 is an objective to control the non-holonomic WMR to follow a desired trajectory, with a given orientation relatively to the path tangent, even when disturbances exist. In the case of trajectory-tracking the path is to be followed under time constraints. The path has an associated velocity profile, 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.

3.1 Kinematic model of a WMR

Figure 2 presents a WMR with two diametrically opposed drive wheels (radius R) and free-wheeling castors (not considered in the kinematic models). $P_r$ is the origin of the robot coordinates system. $L$ is the length of the axis between the drive wheels. $\omega_R$ and $\omega_L$ 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 $\theta_r$ the heading angle with respect to the $x$-axis. In addition, $v_r$ denotes the linear velocity of the robot, and $\omega_r$ 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:

$$\begin{bmatrix}
\dot{x}_r \\
\dot{y}_r \\
\dot{\theta}_r
\end{bmatrix} = \begin{bmatrix}
cos\theta_r & 0 & v_r \\
\sin\theta_r & 0 & 0 \\
0 & 1 & \omega_r
\end{bmatrix} \cdot \begin{bmatrix}
v_r \\
\omega_r
\end{bmatrix} \quad (19)$$

which represents a nonlinear system.

Controllability of the system (19) 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.

3.2 Trajectory-tracking

The first case to be considered is the trajectory-tracking control. 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. 3). The kinematic relationship between the virtual configuration $q_d(t)$ and the corresponding desired velocity inputs $[v_d(t), \omega_d(t)]^T$ is analog with (19):
When a real robot is controlled to move on a desired path it exhibits some tracking error. This tracking error, expressed in terms of the robot coordinate system, as shown in Fig. 3, is given by

\[
\begin{bmatrix}
    \dot{x}_c \\ \\ \dot{y}_c \\ \\ \dot{\theta}_c
\end{bmatrix} = \begin{bmatrix}
    \cos \theta_d & \sin \theta_d & 0 \\
    -\sin \theta_d & \cos \theta_d & 0 \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    v_d \\ \omega_d
\end{bmatrix}
\] (20)

Consequently one gets the error dynamics for trajectory tracking as

\[
\begin{aligned}
\dot{x}_e &= -v_e + v_r \cdot \cos \theta_e + \omega_d \cdot y_e \\
\dot{y}_e &= v_r \cdot \sin \theta_e - \omega_d \cdot x_e \\
\dot{\theta}_e &= \omega_r - \omega_d
\end{aligned}
\] (21)

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, is proposed 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

\[
s_1 = \dot{x}_e + k_1 \cdot x_e,
\]

\[
s_2 = \dot{y}_e + k_2 \cdot y_e + k_0 \cdot \text{sgn}(y_e) \cdot \theta_e.
\] (23)

where \( k_0, k_1, k_2 \) are positive constant parameters, \( x_e, y_e \) and \( \theta_e \) are the trajectory-tracking errors defined in (21).

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

From the time derivative of (23) and using the reaching laws defined in (12), (13), (15) and (17) yields:

\[
\begin{aligned}
\dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot \text{sgn}(s_1) \\
\dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \text{sgn}(y_e) \cdot \dot{\theta}_e = -q_2 \cdot \text{sgn}(s_2)
\end{aligned}
\] (24)

\[
\begin{aligned}
\dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot \text{sgn}(s_1) - p_1 \cdot s_1 \\
\dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \text{sgn}(y_e) \cdot \dot{\theta}_e = -q_2 \cdot \text{sgn}(s_2) - p_2 \cdot s_2
\end{aligned}
\] (25)

\[
\begin{aligned}
\dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -p_1 \cdot |s_1|^\alpha \cdot \text{sgn}(s_1) \\
\dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \text{sgn}(y_e) \cdot \dot{\theta}_e = -p_2 \cdot |s_2|^\alpha \cdot \text{sgn}(s_2)
\end{aligned}
\] (26)

\[
\begin{aligned}
\dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -p_1 \cdot \exp(\alpha \cdot |s_1|) \cdot \text{sgn}(s_1) \\
\dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot \text{sgn}(y_e) \cdot \dot{\theta}_e = -p_2 \cdot \exp(\alpha \cdot |s_2|) \cdot \text{sgn}(s_2)
\end{aligned}
\] (27)

From (21), (22) and (24)-(27), and after some mathematical manipulation, we get the output commands of the sliding-mode trajectory-tracking controller:

\[
\begin{bmatrix}
    \dot{v}_A \\ \dot{\omega}_A \\ \dot{v}_B \\ \dot{\omega}_B \\ \dot{v}_C \\ \dot{\omega}_C \\ \dot{v}_D \\ \dot{\omega}_D
\end{bmatrix} = \begin{bmatrix}
    \frac{s}{\phi} \\ \text{sgn} \left( \frac{s}{\phi} \right)
\end{bmatrix}
\] (32)

where constant factor \( \phi \) defines the thickness of the boundary layer.
Fig. 5. The experimental mobile robot - PatrolBot.

Table 1. Parameter values of the mobile robot - PatrolBot

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass of the robot body</td>
<td>46 kg</td>
</tr>
<tr>
<td>radius of the drive wheel</td>
<td>0.095 m</td>
</tr>
<tr>
<td>distance between wheels</td>
<td>0.480 m</td>
</tr>
</tbody>
</table>

4. EXPERIMENTAL RESULTS

In this section, experimental results of the proposed method are presented.

To show the effectiveness of the proposed sliding mode control law numerically, real experiments were carried out on the trajectory-tracking problem of a nonholonomic wheeled mobile robot. The mobile robot is assumed to have the same structure as in Fig. 2. Parameter values of the mobile robot are given in Table 1. The parameters of sliding modes were held constant during the experiments: $k_1 = 0.75$, $k_2 = 3.75$, and $k_0 = 2.5$; and the desired trajectory is given by $v_d = 0.4 \text{ [m/s]}$, $w_d = 0 \text{ [rad/s]}$.

The robot has two-level control architecture (see Fig. 4). High-level control algorithms (including desired motion generation) are written in C++ and run with a sampling time of $T_s = 100 \text{ ms}$ on an embedded PC, which also provides a user interface with real-time visualization and a simulation environment. Wheel velocity commands,

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

are sent to the PI controllers, and encoder measures $N_R$ and $N_L$ are received in the robots pose estimator for odometric computations.

The real-time experiments are carried out on PatrolBot, a general purpose mobile robot acquired from MobileRobots Inc (see Fig. 5).

4.1 Mobile Robot Setup

PatrolBot is a programmable autonomous general purpose Service robot rover built by MobileRobots Inc.

Technical Specifications

PatrolBot has a 59cm x 48cm x 38cm, CNC aluminum body. Its 19 cm diameter tires handle nearly any indoor surface. The two motor shafts hold 1000-tick encoders. This differential drive platform is holonomic so it can turn

![Fig. 6. Experimental SM-TT control starting from an initial error state $(x_e(0) = -0.3, y_e(0) = -0.3, \theta_e(0) = 0)$.](image)

Fig. 6. Experimental SM-TT control starting from an initial error state $(x_e(0) = -0.3, y_e(0) = -0.3, \theta_e(0) = 0)$.

![Fig. 7. Longitudinal and lateral errors for experimental SM-TT control.](image)

Fig. 7. Longitudinal and lateral errors for experimental SM-TT control.

in place. Moving wheels on one side only, it forms a circle of 29 cm radius. The robot is equipped with 1.6 GHz Intel Pentium processor and 500 MB of RAM.

Software Specifications

A small proprietary μARCS transfers sonar readings, motor encoder information and other I/O via packets from the micro controller server to the PC client and returns control commands. PatrolBot can be operated from the client or users can design their own programs under Linux or under WIN32 using C/C++ compiler. ARIA and ARNL software supply library functions to handle navigation, path planning, obstacle avoidance and many other robotic tasks.

4.2 Real-time Experiments Results

The real-time experiments were made for all types of reaching laws presented in (12), (13), (15) and (17).

In Table 2 are represented 36 experiments using sliding-mode trajectory-tracking controller for PatrolBot robot. Three experimental trials were executed for each parameters of reaching low. The table shows the maximum (Max) and root mean square (RMS) of errors (longitudinal - $x_e$, lateral - $y_e$ and orientation - $\theta_e$). Root mean square error is an old, proven measure of control and quality. RMS can be expressed as $\text{RMS} = \left( \frac{1}{T_s} \sum_{i=1}^{T_s} x^2(i) \right)^{\frac{1}{2}}$.

In order to compare all the four reaching laws there was analyzed the real-time implementation on the PatrolBot Robot. All the 36 real time experiments were realized on a single mobile robot with the same initial error $(x_e = -0.3 \text{ m}, y_e = -0.3 \text{ m}, \theta_e = 0)$.

In Figures 6 and 7 the experimental results for the most favorable cases are presented. These results offer the opportunity to distinguish the performances of the four analyzed reaching laws.
Case is the reaching law

Table 2. Experimental Results

<table>
<thead>
<tr>
<th>Q, P, α</th>
<th>x_e Max [m]</th>
<th>y_e RMS [m]</th>
<th>θ_e RMS [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaching law A.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q2 = 0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q1 = 0.05</td>
<td>0.6810</td>
<td>0.3762</td>
<td>0.3000</td>
</tr>
<tr>
<td>q1 = 0.50</td>
<td>0.6670</td>
<td>0.1823</td>
<td>0.3000</td>
</tr>
<tr>
<td>q1 = 0.95</td>
<td>0.6670</td>
<td>0.2081</td>
<td>0.3000</td>
</tr>
<tr>
<td>q2 = 0.05</td>
<td>0.6670</td>
<td>0.1820</td>
<td>0.3000</td>
</tr>
<tr>
<td>q2 = 1.50</td>
<td>0.6790</td>
<td>0.1880</td>
<td>0.3000</td>
</tr>
<tr>
<td>q2 = 2.50</td>
<td>0.6730</td>
<td>0.1866</td>
<td>0.3000</td>
</tr>
<tr>
<td>Reaching law B.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q2 = 0.5, p1 = 0.75, p2 = 1.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q1 = 0.05, p1 = 0.75, p2 = 1.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1 = 0.05</td>
<td>0.6790</td>
<td>0.3133</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.75</td>
<td>0.6790</td>
<td>0.3180</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 1.00</td>
<td>0.6790</td>
<td>0.2020</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 0.05</td>
<td>0.6670</td>
<td>0.1828</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 1.75</td>
<td>0.6790</td>
<td>0.1880</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 2.50</td>
<td>0.6860</td>
<td>0.1930</td>
<td>0.3000</td>
</tr>
<tr>
<td>Reaching law C.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2 = 1.75, α = 0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1 = 0.5, α = 0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1 = 0.05</td>
<td>0.6840</td>
<td>0.3761</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.50</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.90</td>
<td>0.6690</td>
<td>0.2961</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 0.05</td>
<td>0.6670</td>
<td>0.1820</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 1.75</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 3.00</td>
<td>0.6750</td>
<td>0.1882</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.5, p2 = 1.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>α = 0.05</td>
<td>0.6690</td>
<td>0.1850</td>
<td>0.3000</td>
</tr>
<tr>
<td>α = 0.50</td>
<td>0.6690</td>
<td>0.1850</td>
<td>0.3000</td>
</tr>
<tr>
<td>α = 0.95</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
<tr>
<td>Reaching law D.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p2 = 1.75, α = 0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1 = 0.5, α = 0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p1 = 0.05</td>
<td>0.6840</td>
<td>0.3759</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.50</td>
<td>0.6690</td>
<td>0.1850</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.90</td>
<td>0.6690</td>
<td>0.2961</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 0.05</td>
<td>0.6670</td>
<td>0.1820</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 1.75</td>
<td>0.6690</td>
<td>0.1850</td>
<td>0.3000</td>
</tr>
<tr>
<td>p2 = 3.00</td>
<td>0.6750</td>
<td>0.1882</td>
<td>0.3000</td>
</tr>
<tr>
<td>p1 = 0.5, p2 = 1.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>α = 0.05</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
<tr>
<td>α = 0.50</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
<tr>
<td>α = 0.95</td>
<td>0.6690</td>
<td>0.1851</td>
<td>0.3000</td>
</tr>
</tbody>
</table>

It is easy to observe in figure that the most unfavorable case is the reaching law A. The other laws, cases B, C, and D have similar characteristics with small differences. In Table 2 one can observe the performances for laws C and D are equal. The differences between law B and laws C and D (which are identical) can be observed in Figs. 6 and 7. The same observation can also be extracted from Table 2, where one can observe differences between RMS of x_e and θ_e (in case of reaching law B these values are smaller than for cases C and D).

In Figure 8 the two sliding manifolds for each of the four laws are represented. In figure is also indicated the reaching time for each of the analyzed reaching laws.

![Fig. 8. Sliding surface for experimental SM-TT control.](image1)

![Fig. 9. The most unfavorable case of experimental SM-TT control.](image2)
Figure 9 presents the most unfavorable case with $q_1 = 0.05$, $q_2 = 2.00$, $p_1 = 0.75$ and $p_2 = 1.75$. In this case a too large value of $q_2$ cause severe chattering.

5. CONCLUSIONS

The paper was focused on the performances analysis of the four laws presented in Section II. This performance analysis is based on real-time implementation on PatrolBot Robot. All the experimental results have been presented in a table where the position errors and their mean root square were considered.

Analyzing the performances of the four laws it is easy to see that the most adequate laws for the trajectory tracking problem of the robot are laws $C$ and $D$ (equations (15) and (17)). The most unfavorable of the four laws is law $A$ having the longest reaching time.

ACKNOWLEDGEMENTS

This work was supported in part by the Romanian High Education Scientific Research National Council, under project IDEI-506.

REFERENCES


