Super Twisting Sliding Mode Controller Applied to a Nonholonomic Mobile Robot

Razvan Solea, Daniela Cernega
Faculty of Automatic Control,Computers, Electrical and Electronics Engineering
"Dunărea de Jos" University of Galati, Romania
Email: razvan.solea@ugal.ro, daniela.cernega@ugal.ro

***Abstract*—In this paper, a new approach of the super twisting sliding mode control of nonholonomic mobile robot system is proposed. Computer simulations and real time implementations are performed on a real mobile robot (Pioneer-3DX) with parameter uncertainties and external disturbances. The simulation results and real experiments have shown the improved performance of the proposed controller in terms of a decrease in the reaching and settling times and robustness to disturbances as compared to the conventional sliding mode controller. Moreover, the proposed sliding-mode controller is very simple and easy to implement in real nonholonomic mobile robot.**

***Index Terms*—Wheeled Mobile Robot, Sliding Mode Control, Super Twisting Sliding Mode Control.**

I. Introduction

In the past decades, wheeled mobile robots (WMRs) are widely applied in various industrial and service fields which include manipulation, transportation, inspection, security etc., and they became an object of interest. The applications for WMRs are designed for several mobility configurations (wheel number and type, their location and actuation, single or multibody vehicle structure). The most common for single-body robots are differential drive, tricycle or car-like drive, and omnidirectional steering [1].

One of the most challenging problems, with high relevance for applications, is the problem of autonomous motion planning and control of WMRs. In particular, WMRs are typical systems examples for nonholonomic mechanisms due to the perfect rolling constraints on the wheel motion (e.g. no longitudinal or lateral slipping).

The trajectory tracking control, which is one of the three basic navigation problems, means to track reference trajectories either predefined or given by path planners. The trajectory tracking control of vehicles research work, in the last ten years, lead to the development of various effective methods and tracking controllers.

The model-based tracking control approaches can be divided into kinematic and dynamic based methods. Sliding mode control (SMC) is a well-known control scheme which has been successfully and widely applied for systems with uncertainties [2]. The reason for the popularity of this method is one of its most attractive features: its robustness to external disturbances, parameter variations and uncertainties [3], [4].

The sliding mode control design is done in two steps. The first one is to choose a manifold in the state space that ensures the state trajectories to remain on it. The second one is to design a discontinuous state-feedback controller able to force the system to reach the state on the manifold in finite time. However, the drawback of the SMC is the presence of the chattering effect, caused by the switching frequency of the control. The high frequency components of the control law propagate through the system, excite the unmodeled fast dynamics and therefore cause undesired oscillations. In fact, this can degrade the system performances or may even lead to instability.

In the literature, three main approaches have been presented to reduce the chattering effects. The first class of methods consists in the use of the saturation control instead of the discontinuous one (signum function). The second class of methods is based on the use of a system observer. The use of the high order sliding mode controllers to reduce the chartering phenomenon and to keep the main advantages of the original approach of the SMC is the third class of methods used to eliminate chattering [5], [6].

The high order sliding mode uses differentiators and sliding mode manifold estimators (as shown in [7]) to maintain the robustness of the system. The second order sliding mode control (such as the super-twisting sliding mode control) is relative simple to implement, it provides good robustness to external disturbances. In recent years, super-twisting sliding mode control theory has become very popular and therefore, it has been widely used to control systems with uncertainties. The super-twisting sliding mode control allows finite time convergence of the sliding variable as well as its derivative to zero [6], [8], [9].

The sophisticated control law guarantees the robustness but for the price of increasing the transient time. To improve the transient time of this control approach, recently, as shown in [10], a Lyapunov function was constructed. This function is used to estimate the convergence time for super twisting algorithm [11]. In [12], a multivariable super twisting structure is presented, to analyse the stability using the ideas of Lyapunov function given in [10].

This paper is organized as follows: Section II is dedicated to the analysis of a mathematical model for the trajectory tracking errors of a mobile robot. In section III, a Second Order Sliding Mode Controller is designed with the aim of providing robustness for parametric uncertainties. Simulation and experimental results are given in section IV, in order to illustrate the feasibility and the performance of the proposed

978-1-4799-8481-7/15/$31.00 ©2015 IEEE

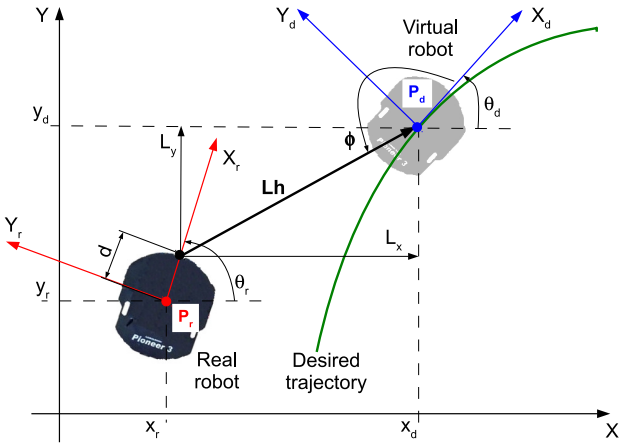

Fig. 1. Trajectory-tracking configuration

scheme. Finally, conclusions of this work are drawn.

II. Trajectory-Tracking Error Models

Figure 1 presents a wheeled mobile robots with two diametrically opposed drive wheels (with radius R). Pr is the origin of the mobile 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 of the mobile robot. 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 WMR 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 mobile robot, and ω_r the angular velocity around the vertical axis.

For an mobile robot rolling on a horizontal plane without slipping, the kinematic model can be expressed by:

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

which represents a nonlinear system.

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 WMR (see Fig. 1). The kinematic relationship between the virtual mobile robot configuration $q_d(t)$ and the corresponding desired velocity inputs $[v_d(t), \omega_d(t)]^T$ is analog with 1:

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

When a real mobile robot is controlled to move on a desired path it exhibits some tracking errors. This tracking errors, expressed in terms of the mobile robot coordinate system, as shown in Fig. 1, is given by

$$\begin{aligned} L_x &= x_d - x_r - d \cdot cos(\theta_r) \\ L_y &= y_d - y_r - d \cdot sin(\theta_r) \\ \theta_e &= \theta_d - \theta_r \end{aligned} \tag{3}$$

The distance Lh between the real robot and virtual robot is measured from the center of the two rear wheels of the virtual robot to the front of the real robot (offset by d from the center of the two rear wheels in the axis), and the bearing angle ϕ is measured from the line of orientation of the virtual robot to the distance line between the two robots (real and virtual).

$$\begin{aligned} Lh^2 &= L_x^2 + L_y^2 \\ \phi &= arctan2(L_y, L_x) - \theta_d + \pi \end{aligned} \tag{4}$$

Differentiating equations (3) we have:

$$\begin{aligned} \dot{L}_x &= \dot{x}_d - \dot{x}_r + d \cdot \dot{\theta}_r \cdot sin(\theta_r) \\ \dot{L}_y &= \dot{y}_d - \dot{y}_r - d \cdot \dot{\theta}_r \cdot cos(\theta_r) \end{aligned} \tag{5}$$

Substituting (1), (2) in (5) we have:

$$\begin{aligned} \dot{L}_x &= v_d \cdot cos(\theta_d) - v_r \cdot cos(\theta_r) + d \cdot \omega_r \cdot sin(\theta_r) \\ \dot{L}_y &= v_d \cdot sin(\theta_d) - v_r \cdot sin(\theta_r) - d \cdot \omega_r \cdot cos(\theta_r) \end{aligned} \tag{6}$$

Define $\gamma = \theta_d - \theta_r + \phi$. Differentiating (4), substituting (6) and using trigonometric identities

$$\begin{aligned} \dot{Lh} &= -v_d \cdot cos(\phi) + v_r \cdot cos(\gamma) + d \cdot \omega_r \cdot sin(\gamma) \\ \dot{\phi} &= \tfrac{1}{Lh} \cdot (v_d \cdot sin(\phi) - v_r \cdot sin(\gamma) + \\ &\quad + d \cdot \omega_r \cdot cos(\gamma) - Lh \cdot \omega_d) \end{aligned} \tag{7}$$

The tracking errors are $Lh_e = Lh - Lh_d$ and $\phi_e = \phi - \phi_d$, respectively. The system (7) can be re-written as

$$\dot{e} = A(e,t) + B(e,t) \cdot u \tag{8}$$

where $u = [v_r, \omega_r]^T$ is the input, $e = [Lh_e, \phi_e]^T$ is the output and

$$\begin{aligned} A(e,t) &= \begin{bmatrix} -v_d \cdot cos(\phi_e + \phi_d) \\ \frac{v_d \cdot sin(\phi_e + \phi_d) - (Lh_e + Lh_d) \cdot \omega_d}{Lh_e + Lh_d} \end{bmatrix} \\ B(e,t) &= \begin{bmatrix} cos(\gamma_e) & d \cdot sin(\gamma_e) \\ \frac{-sin(\gamma_e)}{Lh_e + Lh_d} & \frac{d \cdot cos(\gamma_e)}{Lh_e + Lh_d} \end{bmatrix} \end{aligned} \tag{9}$$

where $\gamma_e = \theta_e + \phi_e + \phi_d$.

Because $Lh > 0$ and $d > 0$, the matrix B is nonsingular.

$$det(B) = det \begin{bmatrix} cos(\gamma_e) & d \cdot sin(\gamma_e) \\ \frac{-sin(\gamma_e)}{Lh_e + Lh_d} & \frac{d \cdot cos(\gamma_e)}{Lh_e + Lh_d} \end{bmatrix} = \frac{d}{Lh} \neq 0 \tag{10}$$

III. Second-order Sliding Mode Controller

The drawback of the first order sliding mode control is the chattering phenomenon. As a solution to resolve this problem, a second order sliding mode (SSMC) is proposed. In fact, the second order sliding mode appears as an effective application to reduce the chattering phenomenon and the switching control signals, with higher relative degrees in finite time.

SSMC controllers require the knowledge of values of the derivatives except for the super twisting algorithm (STW). The STW is a continuous SM algorithm ensuring main properties of the first order sliding mode control for systems with Lipschitz continuous matched uncertainties or disturbances with bounded gradients.

The design procedure of SMC technique can be divided into two steps. The first step is to design the sliding surface such that the system response in the sliding mode has the desired properties. The second step is to design the control law to fetch the trajectory of the system onto the surface for a sliding mode to be realized in finite time.

Trajectories on the two sliding planes are characterized by twisting around the origin, but the continuous control law $u_c(t) = [v_c(t), \omega_c(t)]^T$ is constituted by two terms. The first one is defined by the discontinuous time derivative and the second one is a continuous function of the available sliding variable [13].

The proposed controller is given by the following

$$u_c(t) = \frac{u_1(t) - k_e \cdot e(t) - A(e,t)}{B(e,t)} \tag{11}$$

where the super twisting controller

$$u_1(t) = -k_1 \cdot sign(s(t)) \cdot |s(t)|^{1/2} - k_2 \cdot s(t) + \sigma(t) \tag{12}$$

Variations of the term σ are described by:

$$\dot{\sigma}(t) = -k_3 \cdot sign(s(t)) - k_4 \cdot s(t) \tag{13}$$

where $k_1, ..., k_4$ are positive scalars, and

$$s(t) = e(t) + k_e \cdot \int_0^t e(\tau) \cdot d\tau \tag{14}$$

For the stability proof, the Lyapunov function candidate given in [12] is used:

$$V = 2 \cdot k_3 \cdot |s| + k_4 \cdot s^T \cdot s + \frac{1}{2}\sigma^T \cdot \sigma + \zeta^T \cdot \zeta \tag{15}$$

where

$$\zeta = -k_1 \cdot sign(s) \cdot |s|^{1/2} + k_2 \cdot s - \sigma \tag{16}$$

As shown in papers [12] and [14] $\dot{V} \leq 0$ is true if only if the following conditions are satisfied:

$$\begin{aligned} k_2^2 &= 2 \cdot k_4 \\ 4 \cdot k_1^2 &= \tfrac{5}{2} \cdot k_3 \end{aligned} \tag{17}$$

IV. Simulation and Experimental Results

Nowadays, the path a WMR has to follow is known a priori. More precisely, the GPS coordinates of intermediate points of the path are the input data for the path. These points are usually situated on the median axis of the road (represented with the dashed line in Fig. 2). The WMR has to keep the distance and the angle to road axis represented as the desired trajectory in Fig. 1.

There are some other applications where the WMR has to follow the vehicle before it (e.g. formation control). For these applications, a distance sensor (laser or optical-video) has to monitor the distance to the before vehicle.

In this section, some simulations and experimental tests are presented. The parameters used in this section are: $k_1 = 0.05$, $k_2 = 0.1$, $k_3 = 0.004$, $k_4 = 0.005$, $k_e(1) = 1.25$, $k_e(2) = 0.35$ and $d = 0.2$.

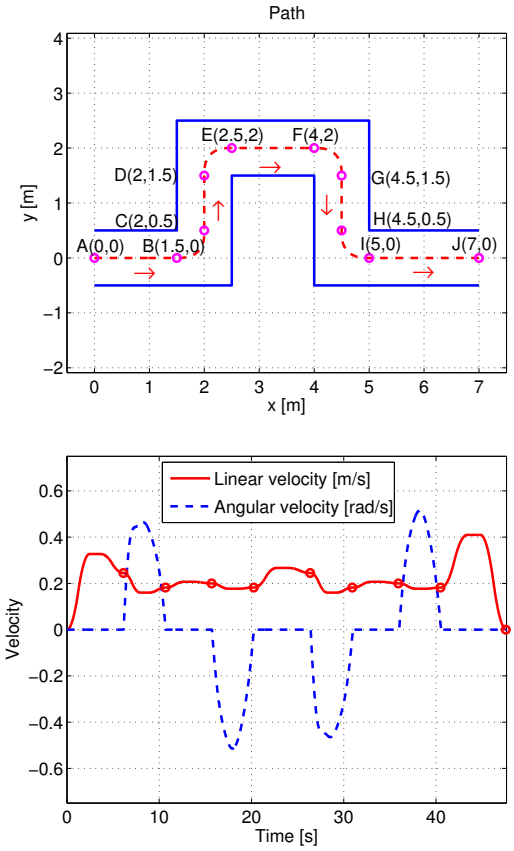

Fig. 2. Path example and desired velocities (linear and angular) calculated by the trajectory planner

In Fig. 3 is represented the control arhitecture used for the implementation of the SSMC with super-twisting control algorithm.

The trajectory planning process can be divided into two separate steps. First, a continuous collision-free path is generated. In a second step, called trajectory generation, a velocity profile along the path is determined. The method used to generate a velocity profile for any two-dimensional path in static environments is similar with the one presented in [15].

Figure 2 shows an example of a planned trajectory using the method described in [15] were the goal was to obtain a

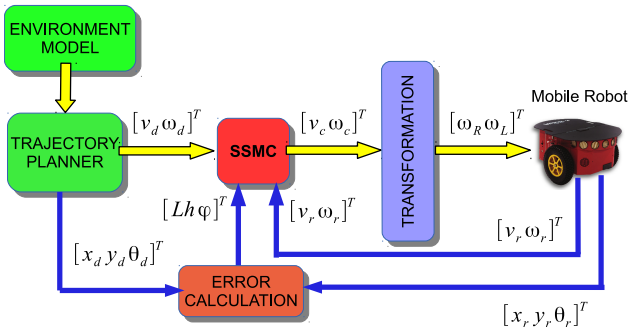

Fig. 3. Control arhitecture

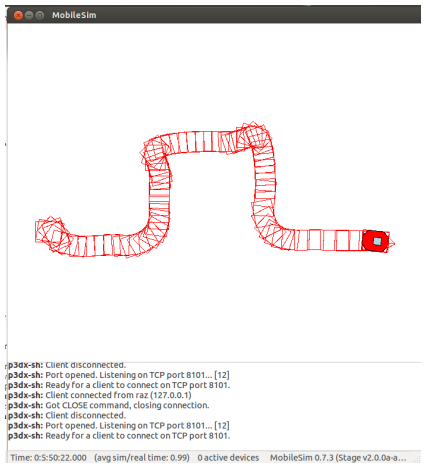

Fig. 4. The MobileSim simulator

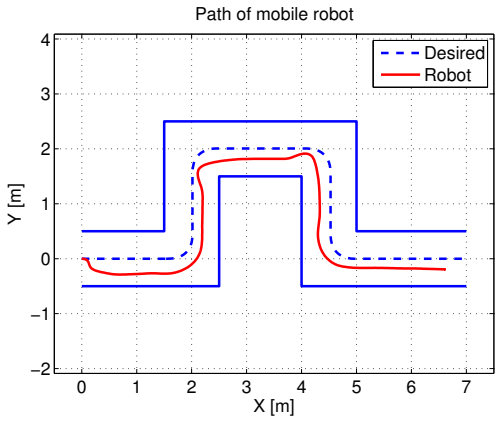

Fig. 5. Simulation results using the path example

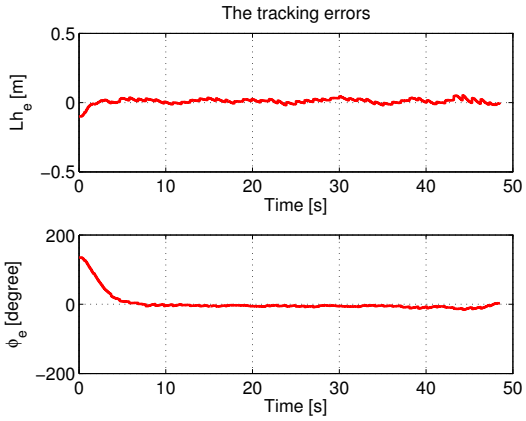

Fig. 6. The traking errors for the path example

smooth path of mobile robot.

In Fig. 3 the SSMC block delivers the velocities command $[v_c, \omega_c]^T$ (the linear and angular commands). These commands are transformed in wheel velocities command for the right and

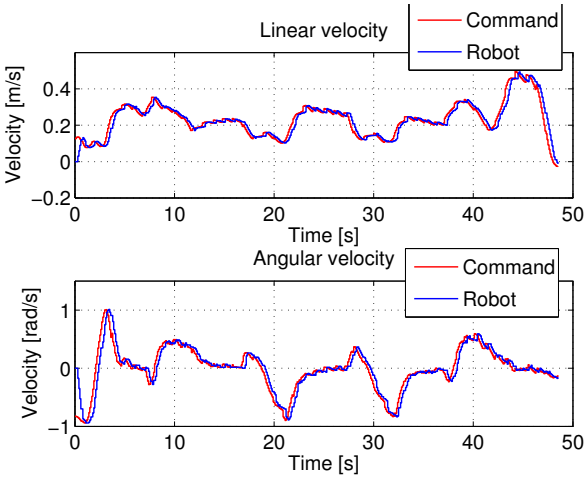

Fig. 7. Linear and angular velocities (command and robot) for the path example

the left wheel:

$$\begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{1}{R} & \frac{L}{R} \\ \frac{1}{R} & \frac{-L}{R} \end{bmatrix} \cdot \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (18)$$

The right and the left wheel commands are sent to the mobile robot. The encoder measures are used in the mobile robot. The odometric computations are then as entries for the Error Calculation block (see Fig. 2). The differences between the measured and reference poses are passed to the tracking controller (SSMC block) after Error Calculation block.

A. Simulation Results

To simulate the proposed algorithm, the MobileSim software [16] was used. In Fig. 4 the simulated trajectory of the robot is presented. MobileSim is software for simulating real mobile robots and their environments, for debugging and experimentation with ARIA.

Written in the C++ language, ARIA (ActivMedia Robotics Interface for Application) is a client-side software for easy, high-performance access and management of the robot server, as well as other robot accessories like sensors and effectors. Its versatility and flexibility makes ARIA an excellent foundation for high-level robotics applications.

The trajectory generated by the Trajectory Planner block, from the control arhitecture used for the simulation is presented in Fig. 3.

In Fig. 5 is easy to observe that the mobile robot moves along the desired trajectory. During this movement the WMR keeps the desired distance ($Lh_d = 0.3m$) and desired angle ($\phi_d = -135°$) to the median axis of the path.

The tracking errors presented in Figure 6 are situated in small neighbourhood of zero. Figures 7 show the evolution of the linear and angular velocities, both for the command and the robot. One can observe that the largest differences between the command velocities and the robot velocities are in the begining of the simulation. Simulation shows small errors for both linear and angular velocities.

B. Experimental Results

Good simulation results encouraged real time implementation. A Pioneer 3-DX differential-drive mobile robot manufactured by MobileRobots Inc was used for experimental implementation. Optical encoders are installed on the driving motor axes. The encoder reading is used to measure the robot velocities, the robot position and orientation (by fusing the encoder reading with the gyro measurement).

The proposed algorithm is written in C++ language and runs in real-time with a sample time $T_s = 100ms$ on an embedded PC.

In Fig. 8 is easy to observe that the mobile robot moves along the desired trajectory and the real trajectory is almost equal with the trajectory obtained in simulation. During this movement the WMR keeps the desired distance ($Lh_d = 0.3m$) and desired angle ($\phi_d = -135°$) to the median axis of the path.

The tracking errors presented in Figure 9 are situated in small neighbourhood of zero. Figures 10 show the evolution of the linear and angular velocities, both for the command and the robot. One can observe that the largest differences between the command velocities and the robot velocities are in the begining of the real-time implementation. The experimental test shows small errors for both linear and angular velocities.

Figures 11 show the comparison between the commands obtained in real-time experiment versus simulation.

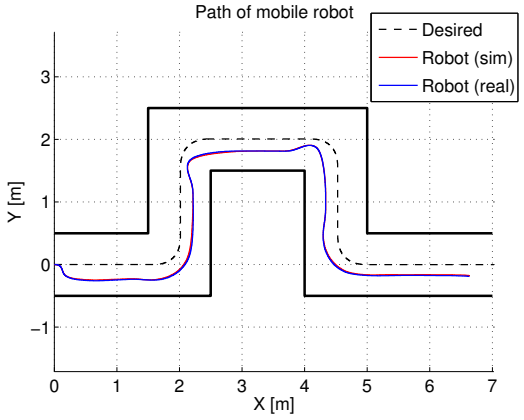

Fig. 8. Experimental and simulation results using the path example

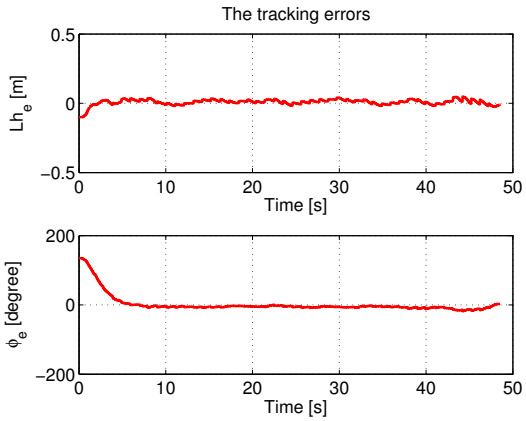

Fig. 9. The traking errors for the path example (using real robot)

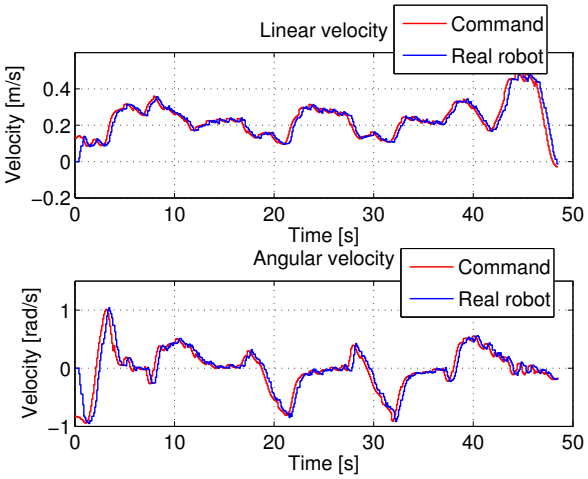

Fig. 10. Linear and angular velocities (command and robot) for the path example (using real robot)

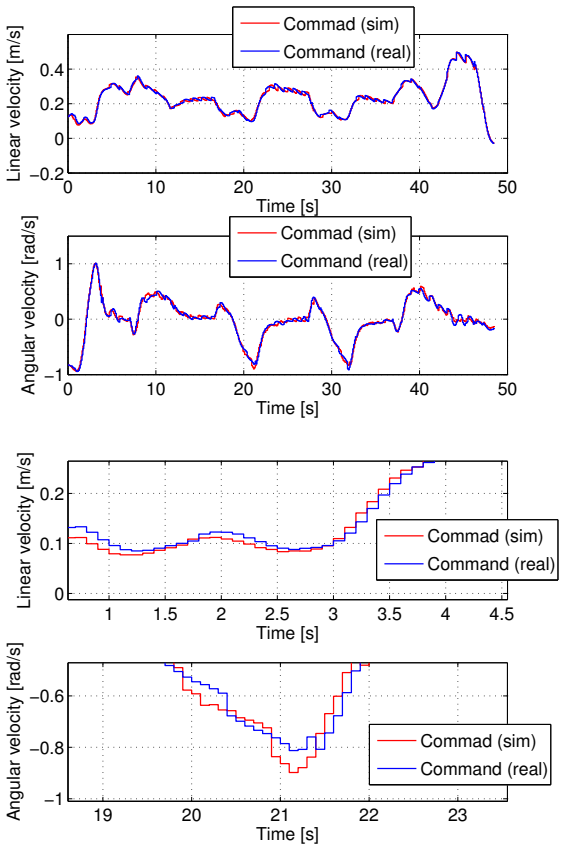

Fig. 11. Comparison between the commands (simulations vs real-time experiments) and zoom in

V. Conclusion

This work presented a super-twisting sliding mode structure for a multivariable system. The strategy using super-twisting control algorithm proved to able to eliminate the chattering problem, thus improving the performance of the trajectory-tracking control. The maximum linear speed equal to $0.5m/s$ is not exceeded either in simulation or in real time implementation. The differences between the linear and angular velocities are greater in case of the real time robot control. These differences are due to the robot dynamics (inertia and friction) when executing a sharp turn.

The simulation is done for a planned trajectory with more than one sharp turn. It is known that classical trajectory tracking control algorithms generate high tracking errors for sharp turns. The simulation results for the proposed algorithm proved the performances in eliminating the chattering and in reducing the tracking errors which are situated in the vicinity of zero.

The simulation showed better performance and the experimental results proved the practical use of the proposed controller. The proposed algorithm proved to be easy to implement on the real robot. Real time implementation of this algorithm demonstrated the simulation performances were correct, only the differences between the command velocities and the real velocities of the robot slightly increased.

Acknowledgment

The work of Razvan Solea has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132397.

The work was supported by the Romanian Executive Unit of Funding Higher Education, Research, Development and Innovation (UEFISCDI), project number PN-II-PT-PCCA-2013-4-0686.

References

[1] R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, *Introduction to Autonomous Mobile Robots, second edition*, The MIT Press Cambridge, Massachusetts London, England, 2011.

[2] H. K. Khalil, *Nonlinear Systems - Third Edition*, Prentice Hall, 2002.

[3] T. Yu and H. Sun *Variable Structure Control of Spherical Robots with Exponential Reaching Law*, Studies in System Science, Vol. 2, pp. 63-67, 2014.

[4] A. Polyakov and A. Poznyak, *Reaching Time Estimation for Super-Twisting Second Order Sliding Mode Controller via Lyapunov Function Designing*, IEEE Transactions on Automatic Control, Vol. 54(8), pp. 1951-1955, 2009.

[5] A. Levant, *Higher-Order Sliding Modes, Differentiation and Output-Feedback Control*, International Journal of Control, Vol. 76(9-10), pp. 924941, 2003.

[6] T. Gonzalez, J. A. Moreno and L. Fridman, *Variable Gain Super-Twisting Sliding Mode Control*, IEEE Transactions on Automatic Control, Vol. 57(8), pp. 2100-2105, 2012.

[7] L. Luque-Vegan, B.Castillo-Toledo, A.G.Loukianov, *Robust block second order sliding mode control for a quadrotor*, Journal of the Franklin Institute, Vol. 349, pp.719739, 2012.

[8] A. Kareem and M. F. Azeem, *Novel Adaptive Super-Twisting Sliding Mode Controller with a Single Input-Single Output Fuzzy Logic Control based Moving Sliding Surface*, International Journal of Control and Automation, Vol. 6(3), pp.183-198, 2013.

[9] J.P. Barbot, M. Djemai, T. Floquet and W. Perruquetti, *Stabilization of a unicycle-type mobile robot using higher order sliding mode control*, European Control Conference (ECC), pp. 519-523, 2003.

[10] J. A. Moreno and M. Osorio. *Strict Lyapunov Functions for the Super-Twisting Algorithm*, IEEE Transactions on Automatic Control, Vol. 57(4), pp. 10351040, 2012.

[11] J.A. Moreno, M. Osorio, *A Lyapunov approach to second-order sliding mode controllers and observers*, Proceedings of 47th IEEE Conference on Decision and Control (CDC 2008), pp. 2856-2861, 2008.

[12] I. Nagesha and C. Edwardsb, *Technical Communique: A Multivariable Super-twisting Sliding Mode Approach*, Journal Automatica (Journal of IFAC), Vol. 50(3), pp. 984-988, 2014.

[13] Y. Shtessel, C. Edwards, L. Fridman and A. Levant, *Sliding Mode Control and Observation*, Publisher Springer New York Heidelberg Dordrecht London, ISBN 978-0-8176-4892-3, 2010.

[14] S. Mahjoub, F. Mnif and N. Derbel, *Second-order sliding mode control applied to inverted pendulum*, 14th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), pp. 269 - 273, 2013.

[15] R. Solea and U. Nunes, *Trajectory planning and sliding-mode control based trajectory-tracking for cybercars*, Integrated Computer-Aided Engineering, IOS Press, Vol. 14(1), pp. 3347, 2007.

[16] MobileRobots/ActivMedia (2005). MobileSim. URL: http://robots.mobilerobots.com/wiki/MobileSim