PARTICLE SWARM OPTIMIZATION USED FOR THE MOBILE ROBOT TRAJECTORY TRACKING CONTROL

Adrian Emanoil Serbencu, Adriana Serbencu and Daniela Cristina Cernega
Control Systems and Industrial Informatics Department, Computer Science Faculty
"Dunarea de Jos" University from Galati, 80021 Str. Domneasca 111, Galati, Romania

Keywords: Trajectory Tracking, Nonlinear Control, Sliding Mode Control, Particle Swarm Optimization.

Abstract: The wheeled mobile robot is a nonlinear system. The trajectory tracking problem is solved using the sliding mode control. In this paper an optimization technique is investigated in order to obtain the best values for the sliding mode control law parameters. The performances of the control law with the optimum parameters are analyzed in order to establish some rules. The conclusions are based on the simulation results.

1 INTRODUCTION

To solve the trajectory tracking problem for a Wheeled Mobile Robot (WMR) it is used a nonlinear model (Slotine and Li, 1991):

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

where x is the state variable; $x^{(n)} = [x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}]$; $x^{(n)}$ 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:

$$s(x) = c^T \cdot x = 0 \tag{2}$$

in finite time with small overshoot with respect to the switching manifold.

The distance between the state trajectory and the switching manifold, s is stated as:

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

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

$$s = \dot{\tilde{x}} + \lambda \cdot \tilde{x} \tag{4}$$

Hence the corresponding switching manifold is $s(t)=0$. 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 possess 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.

The parameters of the control laws have to be positive, and their values influence the reaching rate and the chattering. The values of these parameters are not specified in the literature. In this paper the optimal values for these parameters will be searched.

Many optimization methods were proposed in literature. Recently, the Particle Swarm Optimizer (PSO) proposed by Eberhart and Kennedy (Kennedy and Eberhart, 1995), gained a huge popularity due to its algorithmic simplicity and effectiveness. The PSO is presented in Section 2. Section 3 is dedicated to the Trajectory Tracking Problem for the WMR. This problem is solved within the Sliding Mode approach and the result is the sliding-mode trajectory-tracking controller. The parameters p_i and q_i of the control law are not specified in the literature. In Section 4 PSO is used to determine the optimal values of the control law parameters in order to ensure maximum possible reaching rate of the switching manifold and minimum chattering and the results obtained are presented. Section 5 is dedicated

Serbencu A., Serbencu A. and Cristina Cernega D. (2010).
PARTICLE SWARM OPTIMIZATION USED FOR THE MOBILE ROBOT TRAJECTORY TRACKING CONTROL.
In *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pages 240-245
DOI: 10.5220/0002945702400245

to experimental results. Section 6 is dedicated to the conclusion and future work directions.

2 PARTICLE SWARM OPTIMISATION ALGORITHM

Particle swarm optimization (PSO) is one of the evolutionary computation techniques introduced in 1995 (Kennedy and Eberhart, 1995). The algorithm is initialized with a population of random solutions and searches for optima by updating generations. One entity of the population is named particle. PSO makes use of a velocity vector to update the current position of each particle in the swarm. Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. This value is called *personal best*. The fitness value of *personal best* is also stored.

Another "best" position that is tracked by the PSO is the best one, obtained so far by any particle in the neighbours of the particle. This position is called *local best*. When a particle takes all the population as its topological neighbours, the best position is called global best. The PSO concept consists of, at each time step, changing the velocity of each particle toward its *personal best* and *local best* locations (local version of PSO). Every component of velocity is weighted by a random term, which assures the exploration of problem space.

This process is iterated a set number of times, or until a stop criterion is achieved, for example a threshold of distance (absolute or relative) between the two last positions, below which it is not necessary to go. Using a population of solutions allows PSO to avoid, in most cases, convergence to local optimum.

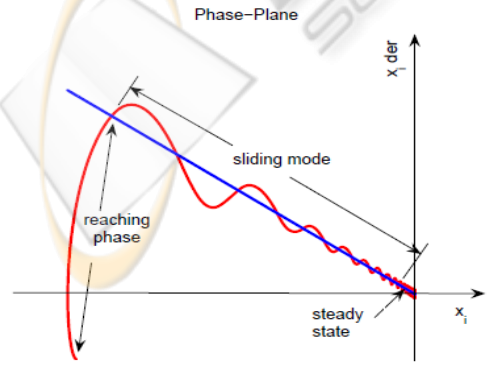

Figure 1: Diagram for the concept of 3-Stages approach.

3 TRAJECTORY TRACKING PROBLEM

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 and the control law used are:

$$\ddot{x} = f(x,\dot{x},t) + B(x,\dot{x},t)\cdot u \quad (5)$$

$$u = p(x,\dot{x},t) \quad (6)$$

where $x=[x_1,x_2,\cdots,x_n]$, $x\in\Re^n$, f is a vector of nonlinear functions, $f\in\mathcal{L}_2^n$, B is a matrix of gains, $B\in\Re^{n\times n}$; $\det(B)\neq 0$; u is the control vector, $u\in\Re^n$.

For the 2nd-order MIMO nonlinear system having the model shown in (6) efficient sliding mode control can be achieved via the following stages (see Figure 1):

1st reaching phase motion; during this stage the trajectory is attracted towards the switching manifold (if the reaching condition is satisfied); characterized by $s_i\neq 0, \tilde{x}_i\neq 0, \tilde{\dot{x}}_i\neq 0$

2nd sliding mode motion; during this stage the trajectory stays on the switching manifold, i.e. $s_i=0, \tilde{x}_i\neq 0, \tilde{\dot{x}}_i\neq 0$

3rd steady state; during this stage both the state variable and the state velocity will converge to the steady state value, therefore:

$$s_i=0, \text{ and } \begin{cases}\tilde{x}_i\to 0, \tilde{\dot{x}}_i\to 0 & or\\ \tilde{x}_i=0, \tilde{\dot{x}}_i=0\end{cases}$$

The reaching law is a differential equation which specifies the dynamics of a switching function *s(x)*. The differential equation of an asymptotically stable *s(x)*, is itself a reaching condition. In addition, by the choice of the parameters in the differential equation, the dynamic quality of the VSC system in the reaching mode can be controlled.

Gao and Hung (Gao and Hung, 1993) proposed a reaching law which directly specifies the dynamics of the switching surface by the differential equation

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

where $Q=diag[q_1,q_2,\cdots,q_n], q_i>0, i=1,2,\ldots,n$

$P=diag[p_1,p_2,\cdots,p_n], p_i>0, i=1,2,\ldots,n$

and $\text{sgn}(s)=[\text{sgn}(s_1),\text{sgn}(s_2),\cdots,\text{sgn}(s_n)^T]$

$h(s)=[h_1(s_1),h_2(s_2),\cdots,h_n(s_n)]^T$; $s_i\cdot h_i(s)>0,\ h_i(0)=0.$

In this paper, a constant plus proportional rate reaching law proposed in Gao and Hung is investigated:

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

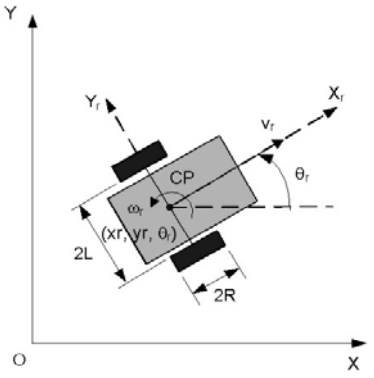

Figure 2: WMR model and symbols.

Clearly, by using the proportional rate term $-P \cdot s$, the state is forced to approach the switching manifolds faster when *s* is large.

The purpose of the trajectory tracking is to control the non-holonomic WMR to follow a desired trajectory, with a given orientation relatively to the path tangent, even when different disturbances exist. In the case of trajectory-tracking the path is to be followed under time constraints. Trajectory tracking is formulated as having the WMR following a virtual target 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. *Pr* is the origin of the robot coordinates system. *2L* 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.

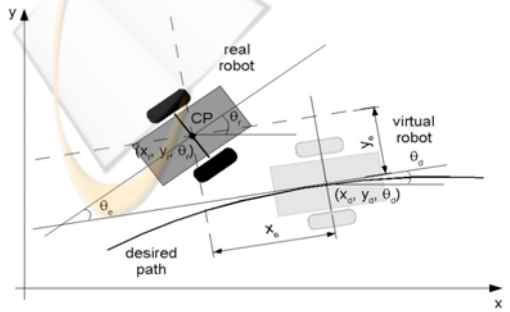

Figure 3: Lateral, longitudinal and orientation errors (trajectory-tracking).

For a unicycle WMR rolling on a horizontal plane without slipping, the kinematic model can be expressed by: which represents a nonlinear system.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} cos\theta_r & 0 \\ sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \tag{9}$$

3.2 Trajectory-tracking

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 Figure 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 analogue with (9):

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} cos\theta_d & 0 \\ sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} \tag{10}$$

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 Figure 3, is given by

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos\theta_d & sin\theta_d & 0 \\ -sin\theta_d & cos\theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix} \tag{11}$$

Consequently one gets the error dynamics for trajectory tracking as

$$\begin{cases} \dot{x}_e = -v_d + v_r \cdot cos\theta_e + \omega_d \cdot y_e \\ \dot{y}_e = v_r \cdot sin\theta_e - \omega_d \cdot x_e \\ \dot{\theta}_e = \omega_r - \omega_d \end{cases} \tag{12}$$

3.3 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 sliding-mode trajectory-tracking (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

$$\begin{aligned} s_1 &= \dot{x}_e + k_1 \cdot x_e \\ s_2 &= \dot{y}_e + k_2 \cdot y_e + k_0 \cdot sgn(y_e) \cdot \theta_e \end{aligned} \tag{13}$$

where k_0, k_1, k_2 are positive constant, x_e, y_e and θ_e are

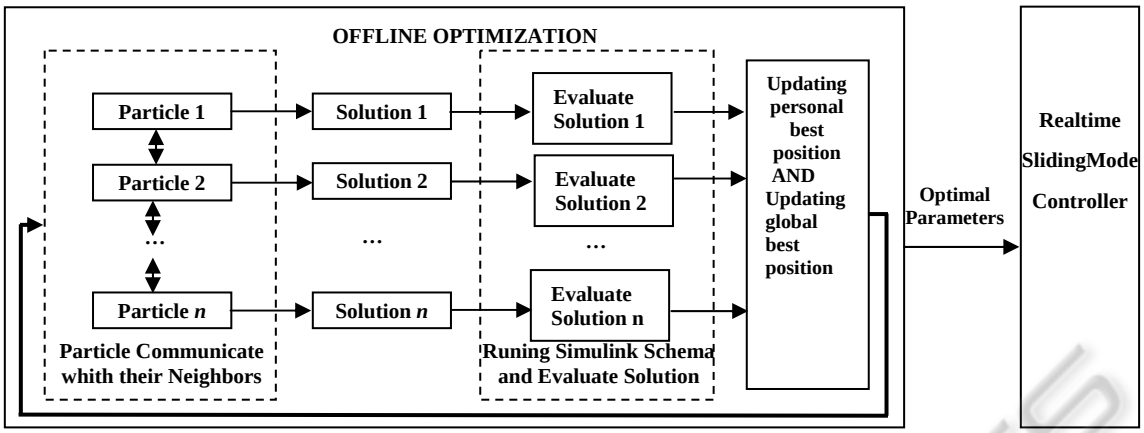

Figure 4: SM-TT controller parameters optimization with PSO.

the trajectory-tracking errors defined in (11).

If s_1 converges to zero, trivially x_e converges to zero. If s_2 converges to zero, in steady-state it becomes $\dot{y}_e = -k_e \cdot y_e - k_0 \cdot sign(y_e) \cdot \theta_e$.

For $y_e < 0 \Rightarrow \dot{y}_e > 0$ if only if $k_0 < k_2 \cdot |y_e| / |\theta_e|$.

For $y_e > 0 \Rightarrow \dot{y}_e < 0$ 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 $\dot{y}_e$ leads to convergence of θ_e to zero.

From the time derivative of (13) and using the reaching laws defined in (8), yields:

$$\begin{aligned} \dot{s}_1 &= \ddot{x}_e + k_1 \cdot \dot{x}_e = -q_1 \cdot sgn(s_1) - p_1 \cdot s_1 \\ \dot{s}_2 &= \ddot{y}_e + k_2 \cdot \dot{y}_e + k_0 \cdot sgn(y_e) \cdot \dot{\theta}_e = \\ &= -q_2 \cdot sgn(s_2) - p_2 \cdot s_2 \end{aligned} \quad (14)$$

From (11), (12) and (14), and after some mathematical manipulation, the output commands of the sliding-mode trajectory-tracking controller result:

$$\dot{v}_c = \frac{-p_1 \cdot s_1 - q_1 \cdot sgn(s_1) - k_1 \cdot \dot{x}_e - y_e \cdot \dot{\omega}_d}{cos(\theta_e)} + \frac{-\dot{y}_e \cdot \omega_d + v_r \cdot \dot{\theta}_e \cdot sin(\theta_e) + \dot{v}_d}{cos(\theta_e)} \quad (15)$$

$$\omega_c = \frac{-p_2 \cdot s_2 - q_2 \cdot sgn(s_2) - k_2 \cdot \dot{y}_e + x_e \cdot \dot{\omega}_d + \dot{x}_e \cdot \omega_d - \dot{v}_r \cdot sin(\theta_e)}{v_r \cdot cos(\theta_e) + k_0 \cdot sgn(y_e)} + \omega_d$$

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

$$\begin{aligned} \dot{V} &= s_1 \cdot \dot{s}_1 + s_2 \cdot \dot{s}_2 = s_1 \cdot (-q_1 \cdot s_1 - p_1 \cdot sgn(s_1)) + \\ &+ s_2 \cdot (-q_2 \cdot s_2 - p_2 \cdot sgn(s_2)) = \\ &= s^T \cdot Q \cdot s - p_1 \cdot |s_1| - p_2 \cdot |s2| \end{aligned}$$

For $\dot{V}$ to be negative semi-definite, it is sufficient to choose q_i and p_i such that q_i, $p_i \geq 0$. But the optima values for q_i, $p_i \geq 0$ will be determined in the next section.

The *signum* functions in the control laws were replaced by *saturation* functions, to reduce the chattering phenomenon (Slotine and Li, 1991).

4 SLIDING MODE CONTROLLER PARAMETERS EVALUATED WITH PSO

Solving the Trajectory Tracking Problem with a SMC, leads to the reaching laws (15). In literature, the parameters *q1, q2*, *p1* and *p2* are usual determined through experiments (Solea and Cernega, 2009) and have great impact on the performance of the controller. *q1, q2* influence the rate at which the switching variable *s(x)* reach the switching manifold *S.* Parameters *p1, p2* force the state *x* to approach the switching manifolds faster when *s* is large.

Choosing parameters through experiments only depends on experience or repeated debugging. In this paper is presented a method of choosing the parameters of the Sliding Mode Controller using the PSO algorithm. The advantages of PSO are: simplicity and efficiency, proven in many other parameters training problems (Mendes *et al.*, 2002; Kim *et al.*, 2008). The optimisation algorithm is working off-line. The *P* and *Q* parameters found by PSO can be used in real-time implementation of SM-TT controller on PatrolBot Robot (see Figure 4).

PSO algorithm is generating, at each step, a number of solutions equal to the number of particles. The quality (fitness) of the solutions is evaluated with the objective function. PSO algorithm requires current solutions fitness to calculate new solutions at the next iteration. The objective function used in PSO takes into account booth the speed of reaching

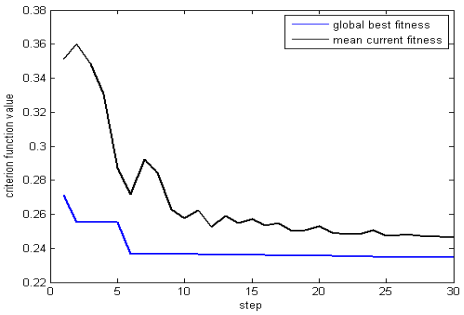

Figure 5: The evolution of best value found by PSO and of the mean criterion function over all particle.

manifolds and the amplitude of the chattering. This is accomplished using the sum of root mean square of the two errors x_e and y_e (11). The evaluation of every set of parameters is achieved after running a numerical simulation of the SM-TT control structure implemented in a Matlab Simulink schema that contains the model of the robot.

The horizon of simulation and initial conditions are chosen to allow a correct comparison between sets of parameters. The step of simulation is selected according to the one used to control the PatrolBot.

Let us suppose that the swarm is composed by n particles. Each particle i is recorded as a structure that transfer from the current iteration t to the next iteration the four elements specified below:

- The current position of i^{th} particle in the search space at the moment t is given through a vector with 4 components

$$x_i(t)=(x_{q1i}(t), x_{q2i}(t), x_{p1i}(t), x_{p2i}(t)).$$

- The current velocity $v_i(t)$ of i particle is a vector with components for each direction of the search space too.
- The best position found up to now by this particle is given by a vector $l_i(t)$ with the same meaning as $x_i(t)$.
- The quality of personal best position $l_i(t)$.

In order to compute the next position where to move, every particle of the swarm needs one more information: the best position found by its neighbours stored in the vector $g_i(t)$. Generally, this is written simply x, v, l, and g. The d^{th} component of one of these vectors is indicated by the index d, for example x_d. With these notations, the motion equations of a particle are, for each dimension $d \in \{ q1i, q2i, p1i, p2i\}$:

$$\begin{cases} v_d \leftarrow c_1 v_d + c_2(l_d - x_d) + c_3(g_d - x_d) \\ x_d \leftarrow x_d + v_d \end{cases} \quad (16)$$

The confidence coefficients are defined as

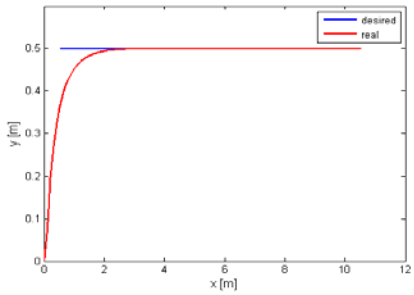

Figure 6: Simulated trajectory with the parameters value found by PSO. Experimental SM-TT control starting from an initial error state ($x_e(0)$ = -0.5, $y_e(0)$ = -0.5, $\theta_e(0)$ = 0).

follows:

- c_1 is (confidence in its own movement) the
- inertia weight, with linear variance from 0.9 to 0.4.
- c_2, c_3 (respectively confidence in its best performance and that of its best informant) are randomly selected at each step according to an uniform distribution in the interval [0, c_{max}].

A circular neighbourhood is used as graph of influence.

5 EXPERIMENTAL RESULTS

Based on the above analysis, mathematical simulation software MATLAB was used to accomplish the experiment simulation study.

The mobile robot PatrolBot used in simulation is assumed to have the same structure as in Figure 2. Parameter values of the PatrolBot are: mass of the robot body 46 [Kg], radius of the drive wheel 0.095 [m], and distance between wheels 0.48 [m]. 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 = 0.5 [m/s], ω_d = 0 [rad/s].

The experiments were done on the robot with the initial error (x_e = -0.5 [m], y_e= -0.5 [m], θ_e = 0 [deg]) and used the reaching law (8).

Settings, used in Matlab implementation of PSO algorithm, are: particle number n = 20; maximal number of iteration = 30, c_{max} =1.9. The criterion function used for solution evaluation are the sum of root mean square (RMS) of the two errors longitudinal - x_e and lateral - y_e. RMS error is an old, proven measure of control and quality. Taking into account that parameters must be positive and value too large can causes chattering, the search interval for each SM parameters was selected to be [0.01 5].

In Figure 5 the evolution of criterion function

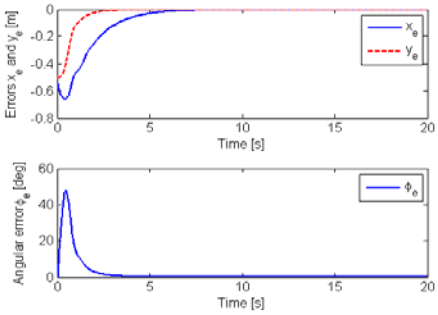

Figure 7: Longitudinal, lateral and orientation errors for experimental SM-TT control.

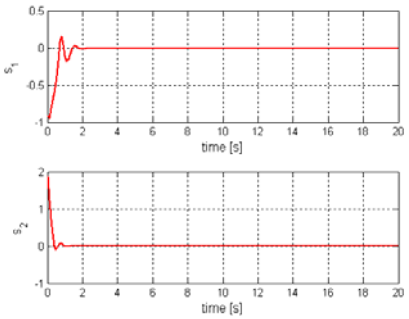

Figure 8: Sliding surface for SM-TT controller.

best value found by PSO is presented. Average value of criterion function for all particles is also recorded. Note that a number of 15 iterations are sufficient to find a good set of values of parameters.

The parameters values for the considered PatrolBot, found by PSO are $q1$=0.4984, $q2$=1.9075, $p1$=1.0230 and $p2$=2.4872.

In Figures 6 and 7, the simulation results for the case of optimised parameters are presented.

In Figure 8 the two sliding manifolds are represented. In Figure 8 one can also see the value of the reaching time.

Figure 9 presents the response of the robot corresponding to the situation of a poor choice of the control law parameters without any optimisation.
It is easy to see the difference between the performances between Figures 6 and 9.

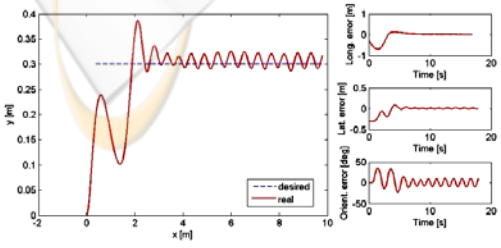

Figure 9: An unfavourable case of experimental SM-TT control.

6 CONCLUSIONS

The paper proposed an efficient method to determine the optimum set of parameters for the sliding mode controller. The PSO algorithm proved to be adequate for this problem because it eliminates the need for repeated simulations in order to find a satisfactory set of parameters. The tests have proven that this optimization technique is efficient for the problem to be solved. A very good solution without chattering was found in a quite acceptable time interval and number of iterations.

The search of the optimum values for the sliding mode trajectory tracking control laws parameters was done in order to use, in the future, such optimum parameters into a supervised control structure having the ability to switch between different controllers.

ACKNOWLEDGEMENTS

This work was supported by the CNCSIS - UEFISCSU, project number IDEI-506/2008.

REFERENCES

Slotine, J., Li, W., 1991. *Applied Nonliner Control*, Prentice Hall, New Jersey.

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

Utkin, V., Young, K., 1978. Methods for constructing discontinuity planes in multidimensional variable structure systems. *Automation and Remote Control, 39*, pp. 1466–1470.

Dorling, C., Zinober, A., 1986. Two approaches to hyperplane design in multivariable variable structure control systems. *Int Journal Control*, Vol. 44, p.65–82

Kennedy, J., Eberhart, R. C., 1995. Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1948.

Solea, R., Cernega, D. C., 2009. Sliding Mode Control for Trajectory Tracking Problem - Performance Evaluation. *Artificial Neural Networks – ICANN 2009, Lecture Notes in Computer Science*, Vol. 5769, pp. 865-874.

Mendes, R., Cortez, P., Rocha, MR., Firns, J., 2002. Particle Swarms for Feedforward Networks Trening. *International Conference on Neural Networks*, Honolulu (Hawaii), USA, pp. 1895-1889.

Kim, T. H., Maruta, I., Sugie, T., 2008. Robust PID controller tuning based on the constrained particle swarm optimization. *Automatica*, Vol. 44, pp. 1104–1110.