Sliding Mode Control for Trajectory Tracking of a Non-holonomic Mobile Robot using Adaptive Neural Networks

Francisco G. Rossomando*, Carlos Soria,**
Ricardo Carelli **

** Subsecretaría de Promoción Científica y Tecnológica, Gobierno de San Juan,*
Centro Cívico – Av. España 50 (S) – 5º Piso CP J5400ARL San Juan, Argentina.
(e-mail: frossomando@sanjuan.gov.ar)
*** Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de San Juan,*
Av. Libertador 1109 Oeste, CP 5400, San Juan, Argentina
(e-mail: {csoria; rcarelli}@inaut.unsj.edu.ar)

Abstract: In this work a sliding mode control method for a nonholonomic mobile robot using adaptive neural network is proposed. Due to this property and restricted mobility, the trajectory tracking of this system has been one of the research topics for the last ten years. The proposed control structure combines a feedback linearization model, based on a kinematics nominal model, and a practical design that combines an indirect neural adaptation technique with sliding mode control to compensate the dynamics of the robot.

A neural sliding mode controller is used to approximate the equivalent control in the neighborhood of the sliding manifold, using an online adaptation scheme. A sliding control is appended to ensure that the neural sliding mode control can achieve a stable closed-loop system for the trajectory-tracking control of a mobile robot with unknown nonlinear dynamics. Also, the proposed control technique can reduce the steady-state error using the online adaptive neural network with sliding mode control; the design is based on Lyapunov's theory. Experimental results show that the proposed method is effective in controlling mobile robots with dynamics large uncertainties.

Keywords: mobile robot, nonlinear systems, adaptive neural control, sliding mode control.

1. INTRODUCTION

In the last years, methodology known as sliding mode control has been researched actively, and the sliding mode control has effectively used in the tracking control of mobile robot by many researchers (Oliveira *et al.,* 2003; Chwa, 2004; Hamerlain *et al.,* 2005, Park *et. al.,* 2009). The concept of sliding mode control has been studied in detail in (Slotine and Lee, 1991; Edwards and Spurgeon, 1998; Utkin, 1992) where it has been used to stabilize a class of non-linear systems.

For faster robot dynamics in the presence of model uncertainties such as parameter perturbations, unknown joint frictions and inertias, and external disturbances, various types of adaptive sliding mode controllers have been developed to alter the control signal to account the changes in robot dynamics and disturbances in the environment, so as to improve the overall performance of the conventional sliding mode control algorithms. The adaptive capability is also important whenever it is difficult to model the system exactly, even without dynamic changes from task to task. Some works present the design of controllers that compensate the robot dynamics. (Petre, 2002) propose the design of an adaptive trajectory tracking controller to generate torques based on a dynamic model whose parameters are unknown. In this work, only simulation results are shown. Other types of trajectory tracking controllers assuming uncertainty in the robot dynamics are developed by (Liu *et al.,* 2004; Dong and Guo, 2005; Dong and Huo, 1999), with the performance being shown just through simulations. (Das and Kar, 2006) show an adaptive fuzzy logic-based controller where the system uncertainty, which includes mobile robot parameter variation and unknown nonlinearities, is estimated by a fuzzy logic system and its parameters are tuned on-line.

Several studies have been published regarding the design of controllers to guide mobile robots during trajectory tracking. Most of the controllers designed so far are based only on the kinematics of the mobile robot, like the controllers presented in (Scaglia *et al*., 2009), in (Wu *et al*., 1999) and in (Künhe *et al*., 2005). To perform tasks requiring high speed movements and/or heavy load transportation, it is important to consider the robot dynamics, besides its kinematics. No matter the uncertainties or changes in its dynamics, the tasks must be performed with due precision. As an example, in the case of load transportation, the dynamic characteristics such as mass, mass center and inertia, change when the robot is loaded. Then, to keep a good performance, the controller should be capable of adapting itself to this kind of changes.

A global time-varying universal controller to achieve stabilization and tracking simultaneously for a mobile robot with saturated inputs is proposed by Do *et al*. (2002). The controller synthesis is based on Lyapunov's direct method and backstepping technique. The control results are based in numerical simulations to validate the effectiveness of the proposed controller. In (Oliveira *et al*., 2003) presents a kinematic control loop based on sliding modes technique and a dynamic control loop based on neural network technique. Simulation and experiments results are included to demonstrate the effectiveness of the proposed control approach.

(Bugeja and Fabri, 2009) present the use of a *RBF-NN* for mobile robot dynamics approximation, in which the weights are estimated stochastically in real-time. The authors show simulation results.

(Park *et al*., 2009) presents an adaptive neural sliding mode controller for nonholonomic wheeled mobile robots with model uncertainties and external disturbances. This work considers a dynamic model with uncertainties and the kinematic model represented by polar coordinates. Self recurrent wavelet neural networks (SRWNNs) are used for approximating arbitrary model uncertainties and external disturbances in dynamics of the mobile robot. To demonstrate the robustness and performance of the proposed control system, it shows computer simulations.

(Kim *et al*., 2000) have proposed a robust adaptive controller for a mobile robot divided in two parts. The first one is based on robot kinematics and is responsible of generating references for the second one, which compensates for the modelled dynamics. However, the adapted parameters are not real parameters of the robot, and no experimental results are presented. Additionally, the control actions are given in terms of torques, while usual commercial robots accept velocity commands.

Similar studies are shown in (Yandong Li *et al*., 2010), where a hybrid control algorithm has been proposed based on PID sliding mode dynamic control and backstepping kinematics control. The algorithm was applied on nonholonomic mobile robot. The gain of sliding mode control is adjusted by using the RBF-NN control with adaptive tuning algorithm and the controller generate torques based on a dynamic model. The results shown are based on simulations.

In (Rossomando *et al*., 2010) presents an approach to adaptive trajectory tracking of mobile robots which combines a feedback linearization based on a nominal model and a RBF-NN adaptive dynamic compensation. The design of the controllers is based on discrete time and the stability analysis is based on linear parameterization of a unicycle-like mobile robot formulated by (De la Cruz and Carelli, 2006). This work shows experimental results.

In (Mehrjerdi *et al*., 2012) presents an adaptive exponential sliding mode control as a solution to reduce chattering, uncertainties and disturbances for the trajectory tracking of a mobile robot. The experimental results show the performance of the proposed technique against uncertainties and disturbances. But the control actions are given in terms of torques.

In this paper, the design of an adaptive trajectory tracking controller based on a nominal robot dynamics and neural controller is developed. The whole control system is designed in two parts: one including a kinematics controller and another one with a neural sliding mode dynamic control, similar to the one in (Kim *et al*., 2000).

Being a realistic consideration in this work, it is supposed that the most important uncertainties of the mobile robot model appear in the dynamic part of it.

The dynamic neuro-controller is designed based on the “sliding mode neuro-adaptive control” (SMNAC), where an online adaptation law is used to adjust the parameters of the radial basis functions (RBF). Such law is conditioned by the sliding surface which has been specified. The dynamic neuro controller uses a neuronal net based on the radial basis functions, being it the main controller in charge of the inverse dynamics of the mobile robot, and where the compensation by sliding surface is designed to delete the approximation error introduced by the neuronal controller.

Furthermore, the adaptation laws of the control system are obtained from the Lyapunov stability criterion and the Barbalat lemma; therefore, the control system stability can be ensured, being it an asymptotic stability. The proposal of a “sliding mode neuro-adaptive control” based controller has the following advantages:

(1) This control technique can be applied to a non-linear MIMO system, being the case of the mobile robot dynamics.

(2) It can control most of the mobile robot dynamic systems without knowing their exact mathematical models.

(3) The dynamic behavior of the controlled system can be dominated by a hybrid control system with a sliding surface.

(4) The focus of this technique has the advantage over the model-based control since the SMNAC does not require previous knowledge of the robot dynamics; moreover, it can be totally tuned on line adjusting the weights, centers, widths and profits.

(5) The proposed control scheme adjusts the main part of the robot dynamics effects, being a robust system. Besides, the controller integrates the Pi control with the SMNAC. With this control technique, the “chattering” effect can be reduced to really small values.

(6) The proposed control scheme in this article over other works mentioned, the control actions are based in terms of angular and linear velocity and not in terms of torque as the variables to be controlled are expressed in terms of velocity.

The paper is organized as follows: Section 2 presents a system overview and shows the mathematical representation of the complete unicycle-like robot model. The kinematics and dynamic controllers are discussed, respectively, in Sections 3 and 4, as well as the corresponding error analysis. Section 5 presents some experimental results to show the performance of the adaptive controller. Finally, conclusions are given in Section 6.

2. ROBOT MODEL

2.1 System overview

In this section, the dynamic model of the unicycle-like mobile robot presented in Fig. 1, is reviewed. Figure 1 depicts the mobile robot, with the parameters and variables of interest. There, x_1 and x_2 are, respectively, the linear and angular velocities developed by the robot, G is the center of mass of the robot, c is the position of the castor wheel, E is the tool location, h is the point of interest with coordinate r_x, r_y in the XY plane, ψ is the robot orientation, a is the distance between the point of interest and the central point of the virtual axis linking the traction wheels.

The mathematical representation of the complete model (De la Cruz and Carelli, 2006), is given by.

Kinematics model

$$\begin{pmatrix}\dot r_x(t)\\ \dot r_y(t)\\ \dot\psi(t)\end{pmatrix}=\begin{pmatrix}\cos\psi(t) & -a\sin\psi(t)\\ \sin\psi(t) & a\cos\psi(t)\\ 0 & 1\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix}+\begin{pmatrix}\delta_{r1}(t)\\ \delta_{r2}(t)\\ 0\end{pmatrix}. \tag{1}$$

Dynamic model

$$\begin{pmatrix}\dot x_1(t)\\ \dot x_2(t)\end{pmatrix}=\begin{pmatrix}\frac{\vartheta_3}{\vartheta_1}x_2^2(t)-\frac{\vartheta_4}{\vartheta_1}x_1(t)\\ -\frac{\vartheta_5}{\vartheta_2}x_1(t)x_2(t)-\frac{\vartheta_6}{\vartheta_2}x_2(t)\end{pmatrix}+\dots \tag{2}$$

$$+\begin{pmatrix}\frac{1}{\vartheta_1} & 0\\ 0 & \frac{1}{\vartheta_2}\end{pmatrix}\begin{pmatrix}u_1(t)\\ u_2(t)\end{pmatrix}+\begin{pmatrix}\delta_{x1}(t)\\ \delta_{x2}(t)\end{pmatrix}.$$

The vector of identified parameters and the vector of uncertainties parameters associated to the mobile robot are, respectively,

$$\vartheta=\begin{pmatrix}\vartheta_1 & \vartheta_2 & \vartheta_3 & \vartheta_4 & \vartheta_5 & \vartheta_6\end{pmatrix}^T,$$
$$\boldsymbol{\delta}=\begin{pmatrix}\delta_{r1} & \delta_{r2} & 0 & \delta_{x1} & \delta_{x2}\end{pmatrix}^T, \tag{3}$$

where δ_{r1} y δ_{r2} are functions of slip velocities and robot orientation, δ_{x1} y δ_{x2} are functions of physical parameters as mass, inertia, wheel and tires diameters, motor and its servos parameters, forces on the wheels, and others. These are considered as disturbances.

The robot's model presented in (1) and (2) is split in a kinematics and a dynamics part respectively, as shown in Fig. 2. Therefore, two controllers are implemented, based on feedback linearization, for both the kinematics and dynamic models of the robot.

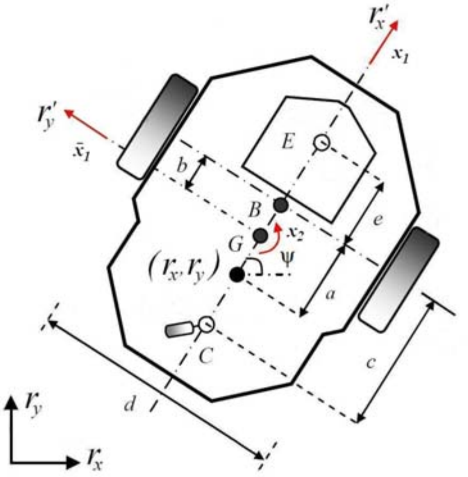

Fig. 1. Mobile Robot parameters.

3. KINEMATICS CONTROLLER

The design of the kinematics controller is based on the robot's kinematics model. The proposed kinematics controller is given by:

$$\begin{pmatrix}x_{1ref}^k(t)\\ x_{2ref}^k(t)\end{pmatrix}=\begin{pmatrix}\cos\psi(t) & \sin\psi(t)\\ -\frac{1}{a}\sin\psi(t) & \frac{1}{a}\cos\psi(t)\end{pmatrix}\dots \tag{4}$$
$$\begin{pmatrix}\dot r_{xref}(t)+l_x\tanh\left(\frac{k_x}{l_x}\tilde r_x(t)\right)\\ \dot r_{yref}(t)+l_y\tanh\left(\frac{k_y}{l_y}\tilde r_y(t)\right)\end{pmatrix},$$

where r_{xref}, r_{yref}, are the coordinates of position reference, $\tilde r_x, \tilde r_y$ are trajectory error, l_x, l_y, and k_x, k_y are adjust gains of *tanh(.)* function. Defining the output error vector $\tilde{\mathbf{h}}=\left(\tilde r_x, \tilde r_y\right)^T$, and expressing eq.(4) in compact form

$$\mathbf{x}_{ref}=\mathbf{H}^{-1}\left(\dot{\mathbf{h}}_{ref}+L\left(\tilde{\mathbf{h}}\right)\right), \tag{5}$$

where

$$\dot{\mathbf{h}}_{ref}=\begin{pmatrix}\dot r_{xref} & \dot r_{yref}\end{pmatrix}^T,$$
$$L\left(\tilde{\mathbf{h}}\right)=\begin{pmatrix}l_x\tanh\left(\frac{k_x}{l_x}\tilde r_x(t)\right) & l_y\tanh\left(\frac{k_y}{l_y}\tilde r_y(t)\right)\end{pmatrix}^T, \tag{6}$$
$$\mathbf{H}^{-1}=\begin{pmatrix}\cos\psi(t) & \sin\psi(t)\\ -\frac{1}{a}\sin\psi(t) & \frac{1}{a}\cos\psi(t)\end{pmatrix}.$$

By replacing (4) in the upper part of (1) under the assumption of perfect velocity tracking, the closed-loop equation is,

$$\begin{pmatrix} \dot{\tilde{r}}_x(t) \\ \dot{\tilde{r}}_y(t) \end{pmatrix} + \begin{pmatrix} l_x & 0 \\ 0 & l_y \end{pmatrix} \begin{pmatrix} \tanh\left(\frac{k_x}{l_x}\tilde{r}_x(t)\right) \\ \tanh\left(\frac{k_y}{l_y}\tilde{r}_y(t)\right) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{7}$$

Equation (6) can be written as

$$\dot{\tilde{\mathbf{h}}}(t) = -L(\tilde{\mathbf{h}}) = \dots$$
$$-\left(l_x \tanh\left(\frac{k_x}{l_x}\tilde{r}_x(t)\right) \quad l_y \tanh\left(\frac{k_y}{l_y}\tilde{r}_y(t)\right)\right)^T . \tag{8}$$

The stability analysis will be showed in section 4.3

4. DYNAMIC CONTROLLER

4.1 Problem formulation

The dynamic controller, implemented as a NN, receives the references of linear and angular velocities which are generated by the kinematics controller $\mathbf{x}_{ref} = \left(x^k_{1ref}, x^k_{2ref}\right)^T$, and produces another pair of linear and angular velocities commands to be sent to the robot servos $\mathbf{u} = \left(u_1, u_2\right)^T$, as shown in Fig. 2.

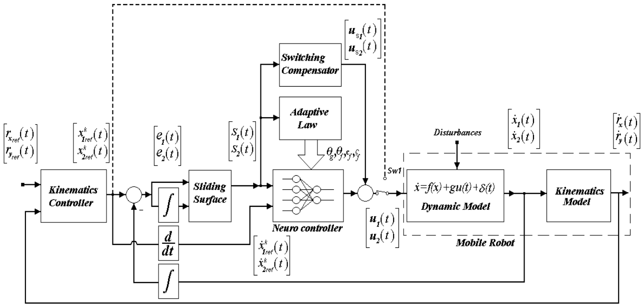
$\begin{bmatrix} r_{x_{ref}}(t) \\ r_{y_{ref}}(t) \end{bmatrix}$
$\begin{bmatrix} x^k_{1ref}(t) \\ x^k_{2ref}(t) \end{bmatrix}$
$\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}$
$\begin{bmatrix} S_1(t) \\ S_2(t) \end{bmatrix}$
$\begin{bmatrix} u_{s1}(t) \\ u_{s2}(t) \end{bmatrix}$
Switching Compensator
Adaptive Law
$\theta_g, \theta_f, c_f, \varsigma_f$
Kinematics Controller
Sliding Surface
Neuro controller
Sw1
Disturbances
$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}$
$\begin{bmatrix} \dot{r}_x(t) \\ \dot{r}_y(t) \end{bmatrix}$
$\dot{x}=f(x)+gu(t)+\delta(t)$
Dynamic Model
Kinematics Model
Mobile Robot
$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$
$\frac{d}{dt}$
$\begin{bmatrix} \dot{x}^k_{1ref}(t) \\ \dot{x}^k_{2ref}(t) \end{bmatrix}$

Fig. 2. Structure Control of NN-SMC.

The robot dynamics system (2) can be written in the following form

$$\dot{\mathbf{x}}\ (t) = \mathbf{f}(\mathbf{x}) + \mathbf{g}\mathbf{u} + \boldsymbol{\delta}(t) =$$
$$= \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \mathbf{u}(t) + \begin{pmatrix} \delta_1(t) \\ \delta_2(t) \end{pmatrix}. \tag{9}$$

In order for (7) to be controllable, it is required that ***g****(t)≠0*, without loss the generality, it is assume that the robot dynamic is a BIBO system. Being $g_{\min}$<***g***(t)<g_{Max} where $g_{\min}$ and g_{Max} are the lower and upper bounds respectively of ***g***(t); f_{Max} represents, the upper bound of ***f***(***x***) and ***δ*** (t) is the unknown time dependent uncertainties, where δ_{Max}, is its upper bound.

$$\delta_{Max} = \sup_{t\in\Re^+} \left|\boldsymbol{\delta}(t)\right|, \tag{10}$$

where

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) & x_2(t) \end{pmatrix}^T,$$
$$\mathbf{u}(t) = \begin{pmatrix} u_1(t) & u_2(t) \end{pmatrix}^T, \tag{11}$$
$$\boldsymbol{\delta}(t) = \begin{pmatrix} \delta_{x1}(t) & \delta_{x2}(t) \end{pmatrix}^T,$$

are the state variables output vector, input vector and uncertainties parameters vector respectively, and the state tracking error is defined as:

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_{ref}(t) =$$
$$= \left(x_1(t) - x^k_{1ref}(t), x_2(t) - x^k_{2ref}(t)\right)^T . \tag{12}$$

4.2 Sliding Mode Control

The control objective is to design an adaptive neural controller which guarantee boundedness of all variables for the closed-loop system and tracking of a given bounded reference signal $\boldsymbol{x}_{ref}$.

The neural feedback linearization method which is based on NN-RBF model can solve this kind of control problem (Slotine and Lee, 1991).

The state tracking error is defined as ***e(t)=x(t)- x***$_{\mathrm{ref}}$***(t)*** and the control objective is to find a control law such that the state ***x*** of the closed-loop system will follow the desired state ***x***$_{\mathrm{ref}}$, in other words, the tracking error should converge to zero.

The purpose of the control law obtained by sliding mode technique is to track the nonlinear trajectory of the system to a pre-specified surface (defined by the designer) in the state space and maintain it in this surface for all subsequent time. This surface is also known as *sliding manifold* because, at least in theory, once this surface is intercepted by the system trajectory, the control law would impose to the system trajectory to track the surface for all the future time (the trajectory will slide over the surface). The dynamics of the process limited to this surface denotes the behavior of the controlled system. The first step is to design the sliding surface according to the desired behavior of the closed-loop system, such as convergence to the origin, parametric variation robustness (Slotine and Lee, 1991; Edwards and Spurgeon, 1998; Utkin, 1992).

For instance, it is shown a simple example to illustrate such technique

A sliding surface for MIMO system can be defined in the error state ***S(x)*** from (12).

$$\mathbf{S}(\mathbf{x})=\begin{pmatrix}\left(\frac{d}{dt}+\lambda_1\right) & 0\\ 0 & \left(\frac{d}{dt}+\lambda_2\right)\end{pmatrix}\int_0^t \mathbf{e}(\tau)d\tau = \tag{13}$$

$$=\begin{pmatrix} e_1(t)+\lambda_1\int_0^t e_1(\tau)d\tau \\ e_2(t)+\lambda_2\int_0^t e_2(\tau)d\tau \end{pmatrix}.$$

The derivate of sliding surface ***S(x)*** *is:*

$$\dot{\mathbf{S}}(\mathbf{x})=\frac{d}{dt}\begin{pmatrix} e_1(t)+\lambda_1\int_0^t e_1(\tau)d\tau \\ e_2(t)+\lambda_2\int_0^t e_2(\tau)d\tau \end{pmatrix}=\begin{pmatrix}\dot{e}_1(t)+\lambda_1 e_1(t)\\ \dot{e}_2(t)+\lambda_2 e_2(t)\end{pmatrix}, \tag{14}$$

where λ_i is a strictly positive constant. Then equivalent problem control is to construct ***u****(t) such that:*

-***S(x)*** is reached in finite time t_S
-***S(x)*** =0; for all $t>t_S$

The solution for the equivalent problem is to choose a control action that:

$$\frac{1}{2}\frac{d}{dt}\left(S_i^2\right)\le -\eta_i\left|S_i\right|;\forall t\ge 0,\eta_i>0 \qquad i=1,2, \tag{15}$$

where S_i is a row of ***S*** matrix. or:

$$\frac{1}{2}\frac{d}{dt}\left(\mathbf{S}^T\mathbf{S}\right)\le -\eta_{\max}\left\|\mathbf{S}\right\| \quad \forall t\ge 0,\eta_{\max}>0, \tag{16}$$

$$\eta_{\max}=\max\left(\eta_1,\eta_2\right),$$

In the design of the sliding-mode control system, first, the ideal equivalent control law $\mathbf{u}^*$, which determines the dynamic of the system on the sliding surface, is derived. The ideal equivalent control law is derived by recognizing

$$\dot{\mathbf{S}}(\mathbf{x})\Big|_{u(t)=u^*(t)}=0, \tag{17}$$

substituting (14) into (17), is obtained

$$0=\dot{\mathbf{e}}(t)+diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)= \tag{18}$$

$$=\left(\mathbf{f}(\mathbf{x})+\mathbf{g}\mathbf{u}^*(t)+\boldsymbol{\delta}(t)-\dot{\mathbf{x}}_{ref}(t)\right)+\dots$$

$$+diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t).$$

Now, let the problem of controlling the uncertain nonlinear system (9) treated in (Slotine and Lee, 1991), defining a control law u* that guarantees the sliding condition of (16) , is composed by an equivalent control $\mathbf{u}(t)=\mathbf{g}^{-1}\left[-\mathbf{f}(\mathbf{x},t)-diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)-\boldsymbol{\delta}(t)+\dot{\mathbf{x}}_{ref}(t)\right]$ and a discontinuous term $\mathbf{u}_S$=-$\boldsymbol{\eta}$sign(***S***) defined by:

$$\mathbf{u}^*(t)=\mathbf{g}(t)^{-1}\left[-\mathbf{f}(\mathbf{x},t)-diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)-\dots\right. \tag{19}$$

$$\left.-\boldsymbol{\delta}(t)-\eta\,\mathrm{sgn}(\mathbf{S})+\dot{\mathbf{x}}_{ref}(t)\right],$$

Where

$$sgn\left(S_i\right)=\begin{cases}1 & \text{for } S_i>0\\ 0 & \text{for } S_i=0\,.\\ -1 & \text{for } S_i<0\end{cases} \tag{20}$$

Let the Lyapunov function candidate defined as

$$V=\sum_{i=1}^{2}\frac{1}{2}\left(S_i^2\right), \tag{21}$$

Differentiating (21) with respect to time

$$\dot{V}=\sum_{i=1}^{2}S_i\dot{S}_i=\sum_{i=1}^{2}S_i\left(\dot{e}_i(t)+\lambda_i e_i(t)\right)=\dots \tag{22}$$

$$=\sum_{i=1}^{2}S_i\left(\dot{x}_i(t)-\dot{x}_{iref}^{k}(t)+\lambda_i e_i(t)\right)=\dots$$

$$=\sum_{i=1}^{2}S_i\left(f_i(\mathbf{x})+g_i u_i(t)+\delta_i(t)-x_{iref}^{k}(t)+\lambda_i e_i(t)\right),$$

replacing (19) in (22).

$$\dot{V}=\sum_{i=1}^{2}S_i\dot{S}_i=\sum_{i=1}^{2}S_i\left(-\eta_i\,\mathrm{sgn}\left(S_i\right)\right)\le-\sum_{i=1}^{2}\eta_i\left|S_i\right|. \tag{23}$$

Then, dividing every term in (23) by $\left|S_i\right|$ and integrating both sides over the interval $0 \le t \le t_S$, where t_S is the time required to hit ***S***, gives

$$\int_0^{t_S}\left(\frac{S_i}{\left|S_i\right|}\dot{S}_i\right)dt\le-\int_0^{t_S}\eta_i\,dt \;\Rightarrow\; \left|S_i\left(t_S\right)\right|-\left|S_i(0)\right|\le-\eta_i t_S\,. \tag{24}$$

In this way, noting that $S_i(t_S)=0$, one has

$$t_S\le\frac{\left|S_i(0)\right|}{\eta_i}. \tag{25}$$

And consequently, the finite time convergence to sliding surface ***S.***

4.3 Neural system adjustment laws

In real systems, ***f(x), g, δ(t)*** and ***η*** are unknown and the function *sgn(**S**)* is not continuous. Thus, it is impossible to generate the control law (19). To overcome these difficulties, is used neural systems for $\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)$, $\hat{\mathbf{g}}\left(\boldsymbol{\theta}_g^*\right)$ and PI for $\hat{\boldsymbol{\delta}}\left(\mathbf{S},\boldsymbol{\theta}_\delta^*\right)$ to approximate respectively ***f(x), g, δ(t)*** and the stabilized term (***η*** sgn(***S***)). The Gaussian function is used as the activation function of each neuron in the hidden layer (26).

$$\hat{\xi}\left(\mathbf{x},\hat{\mathbf{c}}_i,\hat{\varsigma}_i\right)=\exp\left(-\hat{\varsigma}_i^2\left(\mathbf{x}-\hat{\mathbf{c}}_i\right)^T\left(\mathbf{x}-\hat{\mathbf{c}}_i\right)\right), \tag{26}$$

where *i* is the i^{th} neuron of the hidden layer, $\boldsymbol{c}_i$ is the central position of the i^{th} neuron, and ς_i is the spread factor of the Gaussian function.

The structure of RBF-NN is shown in Fig. 3.

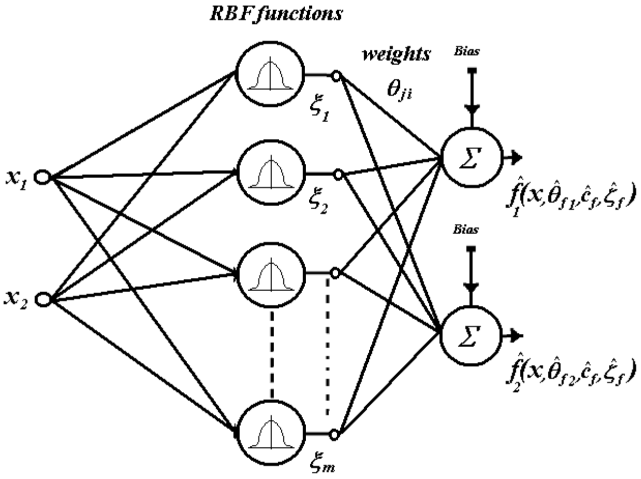
RBF functions
weights
θ_{ji}
Bias
Bias
x_1
x_2
ξ_1
ξ_2
ξ_m
Σ
Σ
$\hat{f}_1(x,\hat{\theta}_{f1},\hat{c}_f,\hat{\varsigma}_f)$
$\hat{f}_2(x,\hat{\theta}_{f2},\hat{c}_f,\hat{\varsigma}_f)$

Fig. 3. Radial Basis Function Network.

Thus, the control law will be written as:
u can be approximated by a RBF-NN through on-line learning,

$$\mathbf{u}=\hat{\mathbf{g}}^{-1}\left(\boldsymbol{\theta}_g^{T*}\right)\Big[-\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)-diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)-\ldots \qquad (27)$$

$$\ldots-\hat{\boldsymbol{\delta}}^*\left(\mathbf{S},\boldsymbol{\theta}_\delta^*\right)+\dot{\mathbf{x}}_{ref}+\mathbf{u}_S\Big],$$

where θ_f^* *(m x j)* and ξ^* *(m x 1)* are optimal parameter vectors of weights θ_f and radial basis functions ξ, respectively; $c^*$and ς^* are optimal parameter vectors of centres ***c*** and spread factor ς, respectively; and $\mathbf{u}_S$ is the stabilized term. However, the optimal parameter vectors are unknown, so it is necessary to estimate the values. Define an estimative function.

$$\mathbf{u}=\hat{\mathbf{g}}^{-1}\left(\boldsymbol{\theta}_g^{T*}\right)\Big[-\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)-diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)-\ldots \qquad (28)$$

$$\ldots-\hat{\boldsymbol{\delta}}^*\left(\mathbf{S},\boldsymbol{\theta}_\delta^*\right)+\dot{\mathbf{x}}_{ref}+\mathbf{u}_S\Big],$$

Where $\hat{\boldsymbol{\theta}}_f$ is a estimated parameter vector of $\boldsymbol{\theta}_f$ and $\hat{\mathbf{c}}$ and $\hat{\varsigma}$ are estimated parameter vectors of ***c*** and ς, respectively of the function vector ξ.

Where $\boldsymbol{\theta}_f$ $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_\delta$ are neural weights and gains of the approximating adaptive system $\hat{\mathbf{f}}\left(\mathbf{x},\hat{\boldsymbol{\theta}}_f,\hat{\mathbf{c}}_f,\hat{\varsigma}_f\right)$, $\hat{\mathbf{g}}\left(\hat{\boldsymbol{\theta}}_g\right)$, $\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)$ respectively, and can be expressed by:

$$\hat{\mathbf{f}}\left(\mathbf{x},\hat{\boldsymbol{\theta}}_f,\hat{\mathbf{c}}_f,\hat{\varsigma}_f\right)=\hat{\boldsymbol{\theta}}_f^T\hat{\boldsymbol{\xi}}(\mathbf{x})=\ldots$$

$$=\begin{pmatrix}\sum\limits_{i=1}^{m}\hat{\theta}_{f1i}^T\exp\left(-\hat{\varsigma}_{fi}^2\left(\mathbf{x}-\hat{\mathbf{c}}_{fi}\right)^T\left(\mathbf{x}-\hat{\mathbf{c}}_{fi}\right)\right)\\ \sum\limits_{i=1}^{m}\hat{\theta}_{f2i}^T\exp\left(-\hat{\varsigma}_{fi}^2\left(\mathbf{x}-\hat{\mathbf{c}}_{fi}\right)^T\left(\mathbf{x}-\hat{\mathbf{c}}_{fi}\right)\right)\end{pmatrix}, \qquad (29)$$

$$\hat{\mathbf{g}}\left(\hat{\boldsymbol{\theta}}_g\right)=\hat{\boldsymbol{\theta}}_g^T, \qquad (30)$$

Where $\hat{\boldsymbol{\theta}}_{f1,2}^T\in\mathbb{R}^{1xm}$ (*m*=*5*) and $\hat{\boldsymbol{\theta}}_g^T\in\mathbb{R}^{2x2}$

It's employed a PI control term in order to attenuate the external disturbances. The MIMO PI control term is in the form of:

$$\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)=\left\{\begin{pmatrix}k_{P1} & k_{I1}\end{pmatrix}\begin{pmatrix}S_1\\ \int_0^t S_1d\tau\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right\}+\ldots$$

$$+\left\{\begin{pmatrix}k_{P2} & k_{I2}\end{pmatrix}\begin{pmatrix}S_2\\ \int_0^t S_2d\tau\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right\}=\ldots \qquad (31)$$

$$\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)=\begin{pmatrix}k_{P1}S_1+k_{I1}\int_0^t S_1d\tau\\ k_{P2}S_2+k_{I2}\int_0^t S_2d\tau\end{pmatrix}=\ldots$$

$$=\begin{pmatrix}\hat{\boldsymbol{\theta}}_{\delta1}^T\boldsymbol{\chi}\left(S_1\right)\\ \hat{\boldsymbol{\theta}}_{\delta2}^T\boldsymbol{\chi}\left(S_2\right)\end{pmatrix},$$

where $\hat{\boldsymbol{\theta}}_{\delta1}^T=\left[k_{P1},k_{I1}\right]$ and $\hat{\boldsymbol{\theta}}_{\delta2}^T=\left[k_{P2},k_{I2}\right]$are an adjustable parameters, and the vectors $\chi\left(S_1\right)=\left[S_1,\int_0^t S_1d\tau\right]^T$ and $\chi\left(S_2\right)=\left[S_2,\int_0^t S_2d\tau\right]^T$ are regressive vectors.

The global control law is given by the equation (27), and defining the minimum approximation error as:

$$\boldsymbol{\varepsilon}=\mathbf{f}\left(\mathbf{x}\right)-\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)+\ldots \qquad (32)$$

$$+\left[\mathbf{g}-\hat{\mathbf{g}}\left(\boldsymbol{\theta}_g^*\right)\right]\mathbf{u}\;+\boldsymbol{\delta}(t)-\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta^*\right).$$

Now, using (14) and considering the dynamic robot model (9) can be written as:

$$\dot{\mathbf{S}}=diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)+\dot{\mathbf{x}}-\dot{\mathbf{x}}_{ref} \qquad (33)$$

$$=diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)+\mathbf{f}\left(\mathbf{x}\right)+\mathbf{gu}\;\;(t)+\boldsymbol{\delta}(t)-\dot{\mathbf{x}}_{ref}$$

Replacing the control action proposed in (27) in (32)

$$\dot{\mathbf{S}}=diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)+\left(\mathbf{f}\left(\mathbf{x}\right)-\hat{\mathbf{f}}\left(\mathbf{x},\hat{\boldsymbol{\theta}}_f,\hat{\mathbf{c}}_f,\hat{\varsigma}_f\right)\right)+\ldots$$

$$+\left(\mathbf{g}-\hat{\mathbf{g}}\left(\hat{\boldsymbol{\theta}}_g\right)\right)\mathbf{u}(t)-diag\left(\lambda_1,\lambda_2\right)\mathbf{e}(t)+\dot{\mathbf{x}}_{ref}+\ldots$$

$$+\mathbf{u}_S+\boldsymbol{\delta}(t)-\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)-\dot{\mathbf{x}}_{ref}=\ldots \qquad (34)$$

$$=\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)-\hat{\mathbf{f}}\left(\mathbf{x},\hat{\boldsymbol{\theta}}_f,\hat{\mathbf{c}}_f,\hat{\varsigma}_f\right)+\ldots$$

$$+\left(\hat{\mathbf{g}}\left(\boldsymbol{\theta}_g^*\right)-\hat{\mathbf{g}}\left(\hat{\boldsymbol{\theta}}_g\right)\right)\mathbf{u}(t)+\ldots$$

$$+\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta^*\right)-\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)+\boldsymbol{\varepsilon}+\mathbf{u}_S.$$

Considering that

$$\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}_f^*,\mathbf{c}_f^*,\varsigma_f^*\right)-\hat{\mathbf{f}}\left(\mathbf{x},\hat{\boldsymbol{\theta}}_f,\hat{\mathbf{c}}_f,\hat{\varsigma}_f\right)=\ldots$$

$$\boldsymbol{\theta}_f^{*T}\boldsymbol{\xi}_f^*\left(\mathbf{x}\right)-\hat{\boldsymbol{\theta}}_f^T\hat{\boldsymbol{\xi}}\left(\mathbf{x}\right)=\left(\hat{\boldsymbol{\theta}}_f^T+\tilde{\boldsymbol{\theta}}_f^T\right)\left(\hat{\boldsymbol{\xi}}\left(\mathbf{x}\right)+\tilde{\boldsymbol{\xi}}\left(\mathbf{x}\right)\right)-\hat{\boldsymbol{\theta}}_f^T\hat{\boldsymbol{\xi}}\left(\mathbf{x}\right)=\ldots \qquad (35)$$

$$=\tilde{\boldsymbol{\theta}}_f^T\hat{\boldsymbol{\xi}}\left(\mathbf{x}\right)+\hat{\boldsymbol{\theta}}_f^T\tilde{\boldsymbol{\xi}}\left(\mathbf{x}\right)+\tilde{\boldsymbol{\theta}}_f^T\tilde{\boldsymbol{\xi}}\left(\mathbf{x}\right)$$

$$\hat{\mathbf{g}}\left(\boldsymbol{\theta}_g^*\right)-\hat{\mathbf{g}}\left(\hat{\boldsymbol{\theta}}_g\right)=\boldsymbol{\theta}_g^{*T}-\hat{\boldsymbol{\theta}}_g^T=\left(\boldsymbol{\theta}_g^{*T}-\hat{\boldsymbol{\theta}}_g^T\right)$$

$$\hat{\boldsymbol{\delta}}\left(\mathbf{S},\boldsymbol{\theta}_\delta^*\right)-\hat{\boldsymbol{\delta}}\left(\mathbf{S},\hat{\boldsymbol{\theta}}_\delta\right)=\begin{pmatrix}\left(\boldsymbol{\theta}_{\delta1}^{*T}-\hat{\boldsymbol{\theta}}_{\delta1}^T\right)\boldsymbol{\chi}\left(S_1\right)\\ \left(\boldsymbol{\theta}_{\delta2}^{*T}-\hat{\boldsymbol{\theta}}_{\delta2}^T\right)\boldsymbol{\chi}\left(S_2\right)\end{pmatrix}=\begin{pmatrix}\tilde{\boldsymbol{\theta}}_{\delta1}^T\boldsymbol{\chi}\left(S_1\right)\\ \tilde{\boldsymbol{\theta}}_{\delta2}^T\boldsymbol{\chi}\left(S_2\right)\end{pmatrix},$$

where $\tilde{\boldsymbol{\theta}}_f, \tilde{\boldsymbol{\theta}}_g$, $\tilde{\boldsymbol{\theta}}_\delta$ and $\tilde{\boldsymbol{\xi}}(\mathbf{x})$ are defined as:

$$\begin{cases} \tilde{\boldsymbol{\theta}}_f = \hat{\boldsymbol{\theta}}_f - \boldsymbol{\theta}_f^* \\ \tilde{\boldsymbol{\theta}}_g = \hat{\boldsymbol{\theta}}_g - \boldsymbol{\theta}_g^* \\ \tilde{\boldsymbol{\theta}}_{\delta 1} = \hat{\boldsymbol{\theta}}_{\delta 1} - \boldsymbol{\theta}_{\delta 1}^* \\ \tilde{\boldsymbol{\theta}}_{\delta 2} = \hat{\boldsymbol{\theta}}_{\delta 2} - \boldsymbol{\theta}_{\delta 2}^* \\ \tilde{\boldsymbol{\xi}}(\mathbf{x}) = \hat{\boldsymbol{\xi}}(\mathbf{x}) - \boldsymbol{\xi}^*(\mathbf{x}), \end{cases} \tag{36}$$

$\dot{\mathbf{S}}$ can be approximated by

$$\begin{aligned} \dot{\mathbf{S}} = &\left(\tilde{\boldsymbol{\theta}}_f^T \hat{\boldsymbol{\xi}}(\mathbf{x}) + \tilde{\boldsymbol{\theta}}_f^T \tilde{\boldsymbol{\xi}}(\mathbf{x}) + \hat{\boldsymbol{\theta}}_f^T \tilde{\boldsymbol{\xi}}(\mathbf{x})\right) + \ldots \\ &+ \tilde{\boldsymbol{\theta}}_g^T \mathbf{u} + \begin{pmatrix} \tilde{\boldsymbol{\theta}}_{\delta 1}^T \boldsymbol{\chi}(S_1) \\ \tilde{\boldsymbol{\theta}}_{\delta 2}^T \boldsymbol{\chi}(S_2) \end{pmatrix} + \boldsymbol{\varepsilon} + \mathbf{u}_S \,. \end{aligned} \tag{37}$$

Using an approximation for the function $\tilde{\boldsymbol{\xi}}_f = \boldsymbol{\xi}_f^*\left(\mathbf{x}, \mathbf{c}_f^*, \boldsymbol{\varsigma}_f^*\right) - \hat{\boldsymbol{\xi}}\left(\mathbf{x}, \hat{\mathbf{c}}_f, \hat{\boldsymbol{\varsigma}}_f\right)$. In order to deal with $\tilde{\boldsymbol{\xi}}$, the Taylor's expansion of $\boldsymbol{\xi}^*$ is taken about $\mathbf{c}^* = \hat{\mathbf{c}}$ and $\boldsymbol{\varsigma}^* = \hat{\boldsymbol{\varsigma}}$

$$\begin{aligned} \boldsymbol{\xi}_f^*\left(\mathbf{x}, \mathbf{c}_f^*, \boldsymbol{\varsigma}_f^*\right) = &\hat{\boldsymbol{\xi}}\left(\mathbf{x}, \hat{\mathbf{c}}_f, \hat{\boldsymbol{\varsigma}}_f\right) + \boldsymbol{\Xi}_f^T \tilde{\mathbf{c}}_f + \ldots \\ &+ \boldsymbol{\Phi}_f^T \tilde{\boldsymbol{\varsigma}}_f + \boldsymbol{\rho}_f\left(\mathbf{x}, \tilde{\mathbf{c}}_f, \tilde{\boldsymbol{\varsigma}}_f\right), \end{aligned} \tag{38}$$

Where $\boldsymbol{\rho}$ denotes the high-order arguments in a Taylor's series expansion, and Ξ and $\boldsymbol{\Phi}$ are derivatives of $\boldsymbol{\xi}_f^*\left(\mathbf{x}, \mathbf{c}_f^*, \boldsymbol{\varsigma}_f^*\right)$ with respect to and $\boldsymbol{\varsigma}_f^*$ at $\left(\hat{\mathbf{c}}_f, \hat{\boldsymbol{\varsigma}}_f\right)$. They are expressed as:

$$\begin{cases} \boldsymbol{\Xi}_f^T = \left.\dfrac{\partial \boldsymbol{\xi}_f^*\left(\mathbf{x}, \mathbf{c}_f^*, \boldsymbol{\varsigma}_f^*\right)}{\partial \mathbf{c}_f^*}\right|_{\substack{\mathbf{c}^* = \hat{\mathbf{c}} \\ \boldsymbol{\varsigma}^* = \hat{\boldsymbol{\varsigma}}}} \\ \boldsymbol{\Phi}_f^T = \left.\dfrac{\partial \boldsymbol{\xi}_f^*\left(\mathbf{x}, \mathbf{c}_f^*, \boldsymbol{\varsigma}_f^*\right)}{\partial \boldsymbol{\varsigma}_f^*}\right|_{\substack{\mathbf{c}^* = \hat{\mathbf{c}} \\ \boldsymbol{\varsigma}^* = \hat{\boldsymbol{\varsigma}}}} \end{cases}, \tag{39}$$

$\tilde{\boldsymbol{\xi}}(\mathbf{x})$ can be expressed as

$$\tilde{\boldsymbol{\xi}}_f = \boldsymbol{\Xi}_f^T \tilde{\mathbf{c}}_f + \boldsymbol{\Phi}_f^T \tilde{\boldsymbol{\varsigma}}_f + \boldsymbol{\rho}_f\left(\mathbf{x}, \tilde{\mathbf{c}}_f, \tilde{\boldsymbol{\varsigma}}_f\right). \tag{40}$$

From (39) the high-order term $\boldsymbol{\rho}$ is bounded by

$$\begin{aligned} \left\|\boldsymbol{\rho}_f\left(\mathbf{x}, \tilde{\mathbf{c}}_f, \tilde{\boldsymbol{\varsigma}}_f\right)\right\| &= \left\|\tilde{\boldsymbol{\xi}}_f - \boldsymbol{\Xi}_f^T \tilde{\mathbf{c}}_f - \boldsymbol{\Phi}_f^T \tilde{\boldsymbol{\varsigma}}_f\right\| \\ &\le \left\|\tilde{\boldsymbol{\xi}}\right\| + \left\|\boldsymbol{\Xi}_f^T \tilde{\mathbf{c}}_f\right\| + \left\|\boldsymbol{\Phi}_f^T \tilde{\boldsymbol{\varsigma}}_f\right\| \\ &\le \kappa_1 + \kappa_2 \left\|\tilde{\mathbf{c}}_f\right\| + \kappa_3 \left\|\tilde{\boldsymbol{\varsigma}}_f\right\|, \end{aligned} \tag{41}$$

where κ$_1$, κ$_2$, and κ$_3$ are some bounded constants due to the fact that RBF and its derivative are always bounded by constants.

Replacing (39) in (36)

$$\begin{aligned} \dot{\mathbf{S}} = &\left(\tilde{\boldsymbol{\theta}}_f^T \hat{\boldsymbol{\xi}}(\mathbf{x}) + \hat{\boldsymbol{\theta}}_f^T \boldsymbol{\Xi}_f^T \tilde{\mathbf{c}}_f + \hat{\boldsymbol{\theta}}_f^T \boldsymbol{\Phi}_f^T \tilde{\boldsymbol{\varsigma}}_f + \boldsymbol{\Delta}\right) + \ldots \\ &+ \tilde{\boldsymbol{\theta}}_g^T \mathbf{u} + \begin{pmatrix} \tilde{\boldsymbol{\theta}}_{\delta 1}^T \boldsymbol{\chi}(S_1) \\ \tilde{\boldsymbol{\theta}}_{\delta 2}^T \boldsymbol{\chi}(S_2) \end{pmatrix} + \boldsymbol{\varepsilon} + \mathbf{u}_S \,. \end{aligned} \tag{42}$$

Remark 1: Where the uncertain $\boldsymbol{\Delta} = \hat{\boldsymbol{\theta}}_f^T \boldsymbol{\rho}_f\left(\mathbf{x}, \tilde{\mathbf{c}}_f, \tilde{\boldsymbol{\varsigma}}_f\right) + \tilde{\boldsymbol{\theta}}_f^T \tilde{\boldsymbol{\xi}}(\mathbf{x})$ is assumed to be bounded by $\|\boldsymbol{\Delta}\| \le \Delta_{Max}$

Theorem Consider the uncertain nonlinear system (9) and assumptions (23)-(25). Then the controller defined by (28) and (40) ensures the finite time convergence of tracking error vector to the boundary layer and its exponential convergence to the closed region Λ$_i$

Proof: Let a positive-definite Lyapunov function candidate V be defined as

$$\begin{aligned} V &= V_1 + V_2 = \frac{1}{2}\sum_{i=1}^{2}\left[\tilde{h}_i^2\right] + \ldots \\ &+ \frac{1}{2}\left(\sum_{i=1}^{2}\left[S_{\Lambda i}^2 + \gamma_1^{-1}\left(\tilde{\boldsymbol{\theta}}_{fi}^T \tilde{\boldsymbol{\theta}}_{fi}\right) + \gamma_2^{-1}\left(\tilde{\boldsymbol{\theta}}_{gi}^T \tilde{\boldsymbol{\theta}}_{gi}\right) + \ldots\right.\right. \\ &\left.\left. + \gamma_3^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta i}^T \tilde{\boldsymbol{\theta}}_{\delta i}\right)\right] + \gamma_4^{-1}\left(\tilde{\mathbf{c}}_f^T \tilde{\mathbf{c}}_f\right) + \gamma_5^{-1}\left(\tilde{\boldsymbol{\varsigma}}_f^T \tilde{\boldsymbol{\varsigma}}_f\right)\right), \end{aligned} \tag{43}$$

where S_Λ is a measure of the distance of the current state to the boundary layer, and can be computed as follows:

$$\begin{aligned} S_{\Lambda i} &= S_i - \Lambda_i \, sat\left(\frac{S_i}{\Lambda_i}\right) \\ sat\left(\frac{S_i}{\Lambda_i}\right) &= \begin{cases} sign\left(S_i\right) & if \quad \left|S_i\right| \ge \Lambda_i \\ \left(S_i / \Lambda_i\right) & if \quad \left|S_i\right| < \Lambda_i . \end{cases} \end{aligned} \tag{44}$$

The presence of a discontinuous term in the control law leads to the well known chattering effect. In order to avoid these undesirable high-frequency oscillations of the controlled variable, the sign function can be replaced by a saturation function, similar as done by Hung and Chung (2007). This substitution smoothes out the control discontinuity and introduces a thin boundary layer S_Λ, in the neighborhood of the sliding surface.

$$S_{\Lambda i} = \left\{e_i \in \mathbb{R} \;\middle/\; S_i\left(e_i\right) \le \left|\Lambda_i\right|\right\}, \tag{45}$$

Where Λ$_i$ is a strictly positive constant that represent the boundary layer thickness.

Now, differentiate (43) with respect to time and noting that $S_{\Lambda i}$=0 inside the boundary layer and $\dot{S}_{\Lambda i} = \dot{S}_i$. Getting $\dot{V}_2 = 0$, inside S$_{\Lambda i}$ and outside is

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 = \sum_{i=1}^{2}\left[\tilde{h}_i \dot{\tilde{h}}_i\right] + \sum_{i=1}^{2}\left[S_{\Lambda i}\dot{S}_i + \ldots\right. \\ &\left. + \gamma_1^{-1}\left(\tilde{\boldsymbol{\theta}}_{fi}^T \dot{\tilde{\boldsymbol{\theta}}}_{fi}\right) + \gamma_2^{-1}\left(\tilde{\boldsymbol{\theta}}_{gi}^T \dot{\tilde{\boldsymbol{\theta}}}_{gi}\right) + \gamma_3^{-1}\left(\tilde{\boldsymbol{\theta}}_{\delta i}^T \dot{\tilde{\boldsymbol{\theta}}}_{\delta i}\right)\right] + \ldots \\ &+ \gamma_4^{-1}\left(\tilde{\mathbf{c}}_f^T \dot{\tilde{\mathbf{c}}}_f\right) + \gamma_5^{-1}\left(\tilde{\boldsymbol{\varsigma}}_f^T \dot{\tilde{\boldsymbol{\varsigma}}}_f\right), \end{aligned} \tag{46}$$

where $\dot{V}_1$ can be expressed by:

$$\begin{aligned} \dot{V}_1 &= \sum_{i=1}^{2}\left[\tilde{h}_i \dot{\tilde{h}}_i\right] = \tilde{\mathbf{h}}^T \dot{\tilde{\mathbf{h}}} = -\tilde{r}_x(t) l_x \tanh\left(\frac{k_x}{l_x}\tilde{r}_x(t)\right) - \ldots \\ &- \tilde{r}_y(t) l_y \tanh\left(\frac{k_y}{l_y}\tilde{r}_y(t)\right) \le 0, \end{aligned} \tag{47}$$

for small values of control error $\tilde{\mathbf{h}}(t)$

$$\begin{aligned} \dot{V}_1 &= -\tilde{r}_x(t) l_x \tanh\left(\frac{k_x}{l_x}\tilde{r}_x(t)\right) - \dots \\ &\quad -\tilde{r}_y(t) l_y \tanh\left(\frac{k_y}{l_y}\tilde{r}_y(t)\right) \approx \dots \\ &\quad \approx -k_x\tilde{r}_x^2(t) - k_y\tilde{r}_y^2(t) \le 0. \end{aligned} \tag{48}$$

Which implies that $\tilde{\mathbf{h}}(t) \to 0$ when $t \to \infty$.

Substituting (42) in $\dot{V}_2$ and considering u_{Si}=-(ε_{Max}+$\varDelta_{Max}$)$sign(S_{\Lambda i})$=-$\eta sign(S_{\Lambda i})$

$$\begin{aligned} \dot{V}_2 = \sum_{\mathrm{i}=1}^{2}\Big[& S_{\Lambda\mathrm{i}}\Big(\Big(\tilde{\boldsymbol{\theta}}_{f\mathrm{i}}^T\hat{\boldsymbol{\xi}}(\mathbf{x}) + \hat{\boldsymbol{\theta}}_{f\mathrm{i}}^T\boldsymbol{\Xi}_f^T\tilde{\mathbf{c}}_f + \hat{\boldsymbol{\theta}}_{f\mathrm{i}}^T\boldsymbol{\Phi}_f^T\tilde{\boldsymbol{\varsigma}}_f + \Delta_\mathrm{i}\Big) + \dots \\ & + \tilde{\boldsymbol{\theta}}_{g\mathrm{i}}^T\mathbf{u} + \tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\boldsymbol{\chi}(S_\mathrm{i}) + \varepsilon_\mathrm{i} - \eta sign(S_{\Lambda\mathrm{i}})\Big) + \dots \\ & + \gamma_1^{-1}\Big(\tilde{\boldsymbol{\theta}}_{f\mathrm{i}}^T\dot{\tilde{\boldsymbol{\theta}}}_{f\mathrm{i}}\Big) + \gamma_2^{-1}\Big(\tilde{\boldsymbol{\theta}}_{g\mathrm{i}}^T\dot{\tilde{\boldsymbol{\theta}}}_{g\mathrm{i}}\Big) + \gamma_3^{-1}\Big(\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\dot{\tilde{\boldsymbol{\theta}}}_{\delta\mathrm{i}}\Big)\Big] + \dots \\ & + \gamma_4^{-1}\Big(\dot{\tilde{\mathbf{c}}}_f^T\tilde{\mathbf{c}}_f\Big) + \gamma_5^{-1}\Big(\dot{\tilde{\boldsymbol{\varsigma}}}_f^T\tilde{\boldsymbol{\varsigma}}_f\Big), \end{aligned} \tag{49}$$

rearranging (49),

$$\begin{aligned} \dot{V}_2 = \sum_{\mathrm{i}=1}^{2}\Big[& S_{\Lambda\mathrm{i}}\tilde{\boldsymbol{\theta}}_{f\mathrm{i}}^T\hat{\boldsymbol{\xi}}(\mathbf{x}) + S_{\Lambda\mathrm{i}}\hat{\boldsymbol{\theta}}_{f\mathrm{i}}^T\boldsymbol{\Xi}_f^T\tilde{\mathbf{c}}_f + S_{\Lambda\mathrm{i}}\hat{\boldsymbol{\theta}}_{f\mathrm{i}}^T\boldsymbol{\Phi}_f^T\tilde{\boldsymbol{\varsigma}}_f + \dots \\ & + S_\mathrm{i}\Delta_\mathrm{i} + S_{\Lambda\mathrm{i}}\tilde{\boldsymbol{\theta}}_{g\mathrm{i}}^T\mathbf{u} + S_{\Lambda\mathrm{i}}\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\boldsymbol{\chi}(S_\mathrm{i}) + S_{\Lambda\mathrm{i}}\varepsilon_\mathrm{i} - \eta\left|S_{\Lambda\mathrm{i}}\right| + \dots \\ & + \gamma_1^{-1}\Big(\tilde{\boldsymbol{\theta}}_{f\mathrm{i}}^T\dot{\tilde{\boldsymbol{\theta}}}_{f\mathrm{i}}\Big) + \gamma_2^{-1}\Big(\tilde{\boldsymbol{\theta}}_{g\mathrm{ii}}^T\dot{\tilde{\boldsymbol{\theta}}}_{g\mathrm{ii}}\Big) + \gamma_3^{-1}\Big(\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\dot{\tilde{\boldsymbol{\theta}}}_{\delta\mathrm{i}}\Big)\Big] + \dots \\ & + \gamma_4^{-1}\Big(\dot{\tilde{\mathbf{c}}}_f^T\tilde{\mathbf{c}}_f\Big) + \gamma_5^{-1}\Big(\dot{\tilde{\boldsymbol{\varsigma}}}_f^T\tilde{\boldsymbol{\varsigma}}_f\Big), \end{aligned} \tag{50}$$

and.

$$\begin{aligned} \dot{V}_2 = \sum_{\mathrm{i}=1}^{2}\Big[& \gamma_1^{-1}\tilde{\boldsymbol{\theta}}_f^T\Big(\gamma_1 S_{\Lambda\mathrm{i}}\hat{\boldsymbol{\xi}}(\mathbf{x}) + \dot{\tilde{\boldsymbol{\theta}}}_{f\mathrm{i}}\Big) + \dots \\ & + \gamma_2^{-1}\tilde{\boldsymbol{\theta}}_{g\mathrm{i}}^T\Big(\gamma_2 S_{\Lambda\mathrm{i}}\mathbf{u} + \dot{\tilde{\boldsymbol{\theta}}}_{g\mathrm{i}}\Big) + \gamma_3^{-1}\tilde{\boldsymbol{\theta}}_{\delta\mathrm{i}}^T\Big(S_{\Lambda\mathrm{i}}\boldsymbol{\chi}(S_\mathrm{i}) + \dot{\tilde{\boldsymbol{\theta}}}_{\delta\mathrm{i}}\Big) + \dots \\ & + \Big(S_{\Lambda\mathrm{i}}\big(\varepsilon_\mathrm{i} + \Delta_\mathrm{i}\big) - \eta\left|S_{\Lambda\mathrm{i}}\right|\Big) + \dots \\ & + \gamma_3^{-1}\tilde{\mathbf{c}}_f^T\Big(\gamma_3\boldsymbol{\Xi}_f\hat{\boldsymbol{\theta}}_{f\mathrm{i}}S_{\Lambda\mathrm{i}} + \dot{\tilde{\mathbf{c}}}_f\Big) + \dots \\ & + \gamma_4^{-1}\tilde{\boldsymbol{\varsigma}}_f^T\Big(\gamma_4\boldsymbol{\Phi}_f\hat{\boldsymbol{\theta}}_{f\mathrm{i}}S_{\Lambda\mathrm{i}} + \dot{\tilde{\boldsymbol{\varsigma}}}_f\Big)\Big], \end{aligned} \tag{51}$$

doing $\dot{\tilde{\boldsymbol{\theta}}}_{f\mathrm{i}} = \dot{\boldsymbol{\theta}}_{f\mathrm{i}}, \dot{\tilde{\boldsymbol{\theta}}}_{g\mathrm{i}} = \dot{\boldsymbol{\theta}}_{g\mathrm{i}}$, $\dot{\tilde{\boldsymbol{\theta}}}_{\delta\mathrm{i}} = \dot{\boldsymbol{\theta}}_{\delta\mathrm{i}}$ and $\dot{\tilde{\mathbf{c}}}_{f\mathrm{i}} = \dot{\mathbf{c}}_{f\mathrm{i}}, \dot{\tilde{\boldsymbol{\varsigma}}}_{f\mathrm{i}} = \dot{\boldsymbol{\varsigma}}_{f\mathrm{i}}$ the parameter adaptation law should be chosen as

$$\dot{\boldsymbol{\theta}}_{f\mathrm{i}} = -\gamma_1 S_{\Lambda\mathrm{i}}\boldsymbol{\xi}(\mathbf{x}), \tag{52}$$

$$\dot{\boldsymbol{\theta}}_{g\mathrm{i}} = -\gamma_2 S_{\Lambda\mathrm{i}}\mathbf{u}, \tag{53}$$

$$\dot{\boldsymbol{\theta}}_{\delta\mathrm{i}} = -\gamma_3 S_{\Lambda\mathrm{i}}\boldsymbol{\chi}(S_\mathrm{i}), \tag{54}$$

$$\dot{\mathbf{c}}_f = -\gamma_3\boldsymbol{\Xi}_f\sum_{i=1}^{2}\Big(\hat{\boldsymbol{\theta}}_{f\mathrm{i}}S_{\Lambda\mathrm{i}}\Big), \tag{55}$$

$$\dot{\boldsymbol{\varsigma}}_f = -\gamma_4\boldsymbol{\Phi}_f\sum_{i=1}^{2}\Big(\hat{\boldsymbol{\theta}}_{f\mathrm{i}}S_{\Lambda\mathrm{i}}\Big), \tag{56}$$

in order to make $\dot{V}_2 < 0$ substituting (52),(53),(54), (55) and (56) in (51)

$$\dot{V}_2 = \sum_{\mathrm{i}=1}^{2}\Big[-\eta\left|S_{\Lambda\mathrm{i}}\right| + \Big(\varepsilon_\mathrm{i} + \Delta_\mathrm{i}\Big)S_{\Lambda\mathrm{i}}\Big] \le 0, \tag{57}$$

wich implies $V_2(t) \le V_2(0)$ and that $\boldsymbol{S}_{\Lambda\mathrm{i}}$ is bounded. From the definition of $\boldsymbol{u}_\mathrm{s}$, it can be easily verified that $\boldsymbol{S}$ is bounded. Considering (13), it can be verified that $\boldsymbol{e}$ is also bounded. Hence (14) implies that $\boldsymbol{S}$ is also bounded.

This result produce a better control of the mobile robot since the error velocities of the dynamic controller tends to zero and the error positions of the kinematics controller tends to zero too.

5. EXPERIMENTAL RESULTS

The effectiveness of the proposed sliding mode neuro-adaptive control approach has been experimentally tested by implementing the control scheme in real-time simultaneously for trajectory tracking control of Pioneer 2DX mobile robot (produced by Active Media Inc.-Fig.4), which admits linear and angular velocities as input reference signals. The Pioneer2DX has an 800MHz Pentium III with 512Mb ram onboard computer in which was programmed the controller. To sensing the robot position, it is used the odometric sensors.

Fig. 4. Pioneer 2DX mobile robot.

For the adjustment of the neuronal net parameters (weights, centers, spread), the so-called parameters with random values are initialized. At the same time, the robot must follow a pre-determined trajectory (in this case a tray. Elliptical) where the proposed control system adjusts its parameters according to the adjustment laws established by the Lyapunov criterion (eq. 52, 53 54, 55 y 56), that after a certain moment, they stabilize and remain constant. The experiment will be conducted with the values obtained during the training. In this experiment, the robot starts at (xd, yd) = (0,0) m. The robot must follow an eight form trajectory reference like this:

$$\begin{cases} x_d = 1.5\sin(0.025\pi t) \\ y_d = 0.4\sin(0.05\pi t) \end{cases}. \tag{58}$$

During the tracking process of the reference trajectories, the control signals of mobile robot has been simultaneously switched from using only conventional kinematic controller to the proposed sliding mode feedback-error-learning neurocontrol schemes (Sw1 in Fig. 2). The gains of the implemented kinematic controller are k$_x$=0.7 and k$_y$=0.7 (Carelli and De la Cruz, 2006). The recorded experimental results are presented in Figures 5–7. The response driven by the kinematic controller only is characterized by a larger value of the tracking error because of the existing time-varying inertia and loads, and model uncertainties. Since the *SMNAC* can compensate for these phenomena through learning, the actual response approached better the desired response. It can be seen that after the performed switching to the proposed neurocontrol scheme, the kinematic velocity signals are changed by the *SMNAC*, the tracking errors are significantly decreased and the error position closely follow the required trajectories demonstrating a very good tracking performance of the investigated control scheme.

During the experiment the *SMNAC* controller was initialized with trained parameters of previous experiments. Figure 5 depicts the speeds and control actions of *SMNAC* controller and outputs velocities, and Fig. 6 shows the trajectory followed by the mobile robot with neural mode sliding controller. Figure 7 shows the distance errors for experiments using the proposed *SMNAC* controller to follow the desired reference trajectory. The distance error is defined as the instantaneous distance between the reference and the robot position. It does easily verify that the error after the switching is less than the position error without *SMNAC* control.

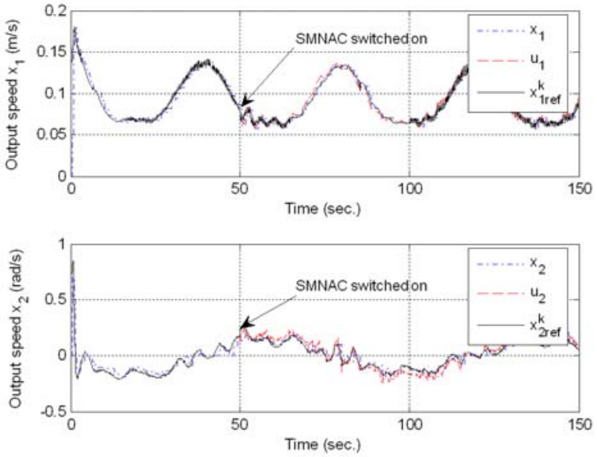

Fig. 5. Mobile robot velocities and control actions when switching to sliding mode neuro-adaptive control (SMNAC).

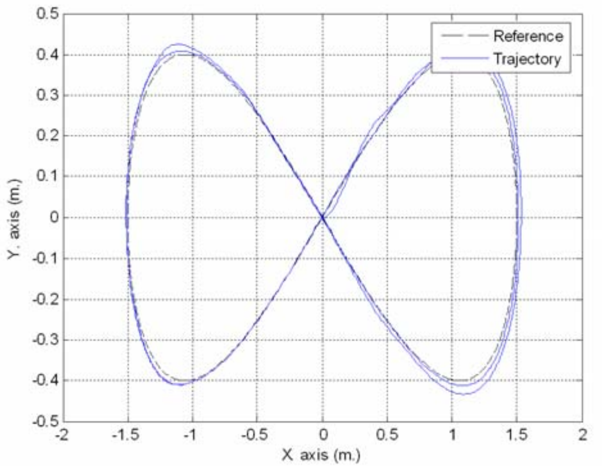

Fig. 6. Reference trajectory and trajectory followed by the mobile robot using SMNAC.

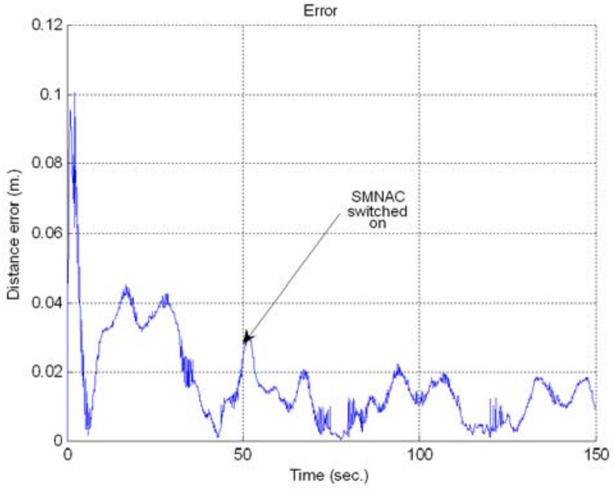

Fig. 7. Norm of the trajectory error using an SMNAC controller and an inverse control for kinematics.

6. CONCLUSIONS

An adaptive neural controller based sliding mode control has been proposed for the robust trajectory tracking of MIMO control systems with unknown nonlinear dynamics. The core of this structure does not require knowledge of the system dynamics and parameters to compute the equivalent control, and an adaptive neural system is developed to further compensate the system uncertainty and knowledge incompleteness. This design obtains robustness in the sense that the self-tuning mechanism can automatically adapt the neural controller by using a learning algorithm and the global asymptotic stability of the algorithm is established via the Lyapunov stability criterion. When matching with the model occurs, the overall control system becomes equivalent to a stable dynamic system.

The design can achieve the goal of composite nonlinear multivariable control and also guarantee that the output tracking error can ultimately converge to zero.

REFERENCES

Bugeja M.K.,. Fabri S.G, and Camilleri L., (2009), 'Dual adaptive dynamic control of mobile robots using neural networks', *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 39 , Issue 1, February, Pages: 129-141.

Carelli R. and De La Cruz C. (2006) Dynamic Modeling and Centralized Formation Control of Mobile Robots", *32nd Annual Conference of the IEEE Industrial Electronics Society IECON*, Paris.

Chwa D. (2004). Sliding-Mode Tracking Control of Nonholonomic Wheeled Mobile Robots in Polar Coordinates. *IEEE Trans. on Control Systems Technology*, Vol. 12, no. 4, pp. 637-644.

Edwards and. Spurgeon S. K. (1998), *Sliding Mode Control: Theory and Applications*, Taylor & Francis, U.K.

Das T. and Kar I. N.,(2006) Design and Implementation of an Adaptive Fuzzy Logic-Based Controller for Wheeled

Mobile Robots, *IEEE Transactions on Control Systems Technology*, Vol. 14, no. 3, May.

De Oliveira V. M., De Pieri E. R., and Lages W. F. (2003), Mobile robot control using sliding mode and neural network. *In Proceedings of the 7th IFAC Symposium on Robot Control*, Vol. 2, pp. 581-586, Wroclaw, Elsevier.

Do, K. D., Jiang, Z. P. e Pan, J. (2002). A universal saturation controller design for mobile robots, Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV,USA, pp. 2044–2049.

Dong W. and Guo Y., (2005) Dynamic tracking control of uncertain mobile robots, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2774-2779.

Dong W. and Huo W., (1999) Tracking Control of Wheeled Mobile Robots with Unknown Dynamics, Proceedings of the IEEE International Conference on Robotics & Automation, Detroit, Michigan, pp. 2645-2650.

Hamerlain, F.; Achour, K.; Floquet, T.; Perruquetti, W. (2005); Higher Order Sliding Mode Control of wheeled mobile robots in the presence of sliding effects, *44th IEEE Conference on Decision and Control*, and the European Control Conference 2005 Seville, Spain, December 12-15.

Hung L. C., Chung H. Y., (2007), Decoupled control using neural network-based sliding-mode controller for nonlinear systems, *Expert Systems with Applications,* Elsevier, 32, 1168–1182

Kim M. S., Shin J. H. and Lee J. J.,(2000) Design of a Robust Adaptive Controller for a Mobile Robot, *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1816-1821.

Künhe F., Gomes J. and Fetter W., (2005) Mobile Robot Trajectory Tracking Using Model Predictive Control, *II IEEE Latin-American Robotics Symposium*, São Luis, Brazil.

Li, Y., Wang, Z., & Zhu, L; (2010), Adaptive neural network PID sliding mode dynamic control of nonholonomic mobile robot, Conference on Information and Automation (ICIA), IEEE International, pp. 753 - 757

Liu S., Zhang H., Yang S. X. and Yu J. (2004), Dynamic Control of a Mobile Robot Using an Adaptive Neurodynamics and Sliding Mode Strategy, *Proceedings of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, China, pp. 5007-5011.

Mehrjerdi, H., Zhang, Y., & Saad, M. (2012). Adaptive exponential sliding mode control for dynamic tracking of a nonholonomic mobile robot. *Intelligent Robotics and Applications*, 643-652.

Park B. S., Yoo S. J., Park J. B., and Choi Y. H. (2009) Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *IEEE Trans. Control Syst. Technol.*, Vol. 17, no. 1,pp. 207–214, Jan.

Petre E., An Adaptive Tracking Controller For A Mobile Robot Using Neural Networks, *Journal of Control Engineering and Applied Informatics*, Vol.4 No. 3, pp. 25-32, 2002

Rossomando F.G., Soria C., and Carelli R. 'Autonomous Mobile Robot Navigation using RBF Neural Compensator', *Control Engineering Practice*, Elsevier, 2010, Vol.19, No. 3, pp. 215–222.

Scaglia G., Montoya L.Q., Mut V.A., and Sciascio F.D., (2009) Numerical methods based controller design for mobile robots", *Robotica*, Vol.7, Nro 2, pp.269-279.

Slotine J.J.E. and Li W., (1991), *Applied Nonlinear Control*, Prentice Hall, Upper Saddle River, NJ.

Utkin, V.I. (1992). Sliding Modes in Control and Optimization. Springer-Verlag. ISBN 978-0387535166.

Wu W., Chen H., Wang Y.and Woo P., (1999)Adaptive Exponential Stabilization of Mobile Robots with Uncertainties, Proceedings of the IEEE 38th Conference on Decision and Control, Phoenix, Arizona, USA, pp.3484-3489.