Engineering Applications of Artificial Intelligence 26 (2013) 2251–2259

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Adaptive neural sliding mode compensator for a class of nonlinear systems with unmodeled uncertainties

Francisco G. Rossomando [a,*], Carlos Soria [b], Ricardo Carelli [b]

[a] Subsecretaria de Promoción de la Actividad Científica, Secretaria de Estado de Ciencia, Tecnología e Innovación, Gobierno de la Provincia de San Juan, Av. San Martín (oeste), 750 San Juan, Argentina
[b] Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de San Juan, Av. San Martín (oeste), 1109 San Juan, Argentina

ARTICLE INFO

Article history:
Received 8 December 2012
Received in revised form
21 July 2013
Accepted 23 August 2013
Available online 29 September 2013

Keywords:
Nonlinear systems
Neural networks
MIMO systems
sliding mode control
Radial basis functions

ABSTRACT

This paper addresses the problem of adaptive neural sliding mode control for a class of multi-input multi-output nonlinear system. The control strategy is an inverse nonlinear controller combined with an adaptive neural network with sliding mode control using an on-line learning algorithm. The adaptive neural network with sliding mode control acts as a compensator for a conventional inverse controller in order to improve the control performance when the system is affected by variations in its entire structure (kinematics and dynamics). The controllers are obtained by using Lyapunov's stability theory. Experimental results of a case study show that the proposed method is effective in controlling dynamic systems with unexpected large uncertainties.

1. Introduction

In general, different nonlinearities of a system are difficult to treat within a single theoretical framework. Sometimes, owing to limited knowledge about physical phenomena, some explicit phenomena cannot be described accurately by nonlinear functions. These factors make it difficult to design controllers for nonlinear systems. Many methodologies, such as feedback linearisation control (Slotine and Li, 1991; Isidori, 1995) and optimal control (Dierks and Jagannathan, 2010; Imae et al., 2009; Zhang et al., 2009) are based on the assumption that the mathematical model of the nonlinear system is known. However, the explicit formulation of many real complex systems is very difficult and many times the exact mathematical model is not available. In this case, conventional control methods may be insufficient to obtain a controller that provides good performance. Some solutions have been proposed, such as robust control (Chen et al., 2010; Kuperman and Zhong, 2011), neural networks (NNs) (Liu et al., 2004; Das and Kar, 2006; Rossomando et al., 2011) and fuzzy control systems (Sang et al., 2012).

Recently, the design of control systems with neural adaptation has been discussed widely in the literature and the analytical study of adaptive control systems using nonlinear functions with universal approximators has also received much attention. Generally, these methods use fuzzy logic and neural networks (NNs) as approximation models for the unknown nonlinear systems. In Zhang and Quan (2001), a fuzzy system of a hyperbolic model is used to identify and control a nonlinear model and in Li and Tong (2003), a state observer is used to control a multi-input multi-output (MIMO) nonlinear system using an adaptive fuzzy controller. In both cases, the adaptive controller must learn the entire dynamics. In Zhang et al. (2009), an algorithm is developed to determine the approximate optimal control for a nonlinear discrete system with constraints. The effectiveness of the control technique is proven through a simulation study.

Mclain et al. (1999) present an adaptive nonlinear control that does not require a detailed model of the process. Nevertheless, this method is addressed to single-input single-output systems and the controller must identify the entire structure of the system. One common technique by which to obtain the controller of an adaptive nonlinear system is the design based on some stability theory (for example the Lyapunov method), from which the parameter adjustment law is obtained.

The most useful property of NNs in control theory is their learning ability in approximating a nonlinear function. From this, many controllers based on NNs have been developed for

* Corresponding author. Tel.: +54 2644306347.
E-mail addresses: frossomando@sanjuan.gov.ar (F.G. Rossomando), csoria@inaut.unsj.edu.ar (C. Soria), rcarelli@inaut.unsj.edu.ar (R. Carelli).
URLS: http://www.sanjuan.gov.ar (F.G. Rossomando), http://www.inaut.unsj.edu.ar (C. Soria), http://www.inaut.unsj.edu.ar (R. Carelli).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.08.008

compensation of nonlinearities and uncertainties in control systems (Liu et al., 2004; Rossomando et al., 2011).

In this work, the problem of adaptive neural control for a type of nonlinear dynamic system with unmodelled uncertainties is considered. The goal is to develop a feedback linearisation controller and an adaptive NN with sliding surfaces, in order to guarantee asymptotic convergence of the control errors to zero. The adaptive neural sliding mode compensation (ANSMC) will have the effect of reducing the difference in the parameter adjustment between the real model and the nominal model used for the design of the feedback linearisation controller. The ANSMC should be set on-line, reducing the effects of model uncertainties and all possible disturbances that may appear. This control technique does not need to learn the entire model of the system and it allows compensation for all model uncertainties by means of a single neural network.

The paper is organised as follows. Section 2 presents a general view of the nonlinear systems (MIMO) and the description of the uncertainties. In Section 3, the formulation of the analytical model is presented. The study of the feedback linearisation controller and the compensation radial basis function (RBF) network with sliding surface is carried out in Section 4. In Section 5, the stability is analysed to obtain the law for parameter adjustment. In Section 6, an analysis of a mobile robot is performed and in the Section 7, the experimental results are presented. Finally, Section 8 offers conclusions that confirm the effectiveness and applicability of the proposed method.

2. Problem formulation

Considering the parameter uncertainties and unmodelled dynamics, the system model can be expressed in compact form as

$$
\begin{aligned}
\dot{\mathbf{x}} &= f(\mathbf{x})+\tilde{f}(\mathbf{x})+\sum_{i=1}^{m}(g_i(\mathbf{x})+\tilde{g}_i(\mathbf{x}))u_i \\
y_1 &= h_1(\mathbf{x})+\tilde{h}_1(\mathbf{x}) \\
&\vdots \\
y_m &= h_m(\mathbf{x})+\tilde{h}_m(\mathbf{x})
\end{aligned} \tag{1}
$$

where $x=[x_1,x_2,\ldots,x_n]^T\in\Re^n$ is the state vector, $u=[u_1,u_2,\ldots,u_m]^T\in\Re^m$ is the control input vector, $y=[y_1,y_2,\ldots,y_m]^T\in\Re^m$ is the output vector and f, g_i, and h_i, are smooth nonlinear functions.

Assumption 1. Dynamic variations in (1) produce the smooth vector fields represented by $\tilde{f}(\mathbf{x})$, $\tilde{g}(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$, which represent nonlinear unknown functions.

Assumption 2. The desired trajectories y_{jref}; $j=1, 2,\ldots, n$ and their time derivatives up to the nth order are continuous and bounded functions.

Assumption 3. The input signals u_i; $i=1, 2,\ldots, m$ are continuous and bounded functions.

3. Model representation

The MIMO nonlinear system from (1) without uncertainties is given by

$$
\begin{aligned}
\dot{\mathbf{x}} &= f(\mathbf{x})+\sum_{i=1}^{m}g_i(\mathbf{x})u_i \\
y_1 &= h_1(\mathbf{x}) \\
&\vdots \\
y_m &= h_m(\mathbf{x})
\end{aligned} \tag{2}
$$

If the system is feedback linearisable (Slotine and Li, 1991) by static state feedback and it has a well-defined vector relative degree $r=[r_1, r_2,\ldots, r_m]^T$, where the r_is are the smallest integers such that at least one of the inputs appears in $y_i(r_i)$, the input-output (IO) differential equations of the system are given by

$$
y_i^{(ri)} = L_f^{(ri)}h_i(\mathbf{x})+\sum_{j=1}^{m}L_{gj}(L_f^{(ri-1)}h_i(\mathbf{x}))u_j \tag{3}
$$

with at least one of $L_{gj}(L_f^{(ri-1)}h(\boldsymbol{x}))\neq 0$. Note that $L_{gj}h_i(\boldsymbol{x}):R^n\to R$ is the Lie derivative of $h(\boldsymbol{x})$ with respect to f and g, which are given by $L_{fj}h_i(\boldsymbol{x})=(\partial h_i(\boldsymbol{x})/\partial\boldsymbol{x})f(\boldsymbol{x})$ and $L_{gj}h_i(\boldsymbol{x})=(\partial h_i(\boldsymbol{x})/\partial\boldsymbol{x})g(\boldsymbol{x})$.

This way, the plant IO equation can be written as

$$
\begin{aligned}
\begin{pmatrix} y_1^{(r1)} \\ y_2^{(r2)} \\ \vdots \\ y_m^{(rm)} \end{pmatrix} &= \begin{pmatrix} L_f^{(r1)}h_1(\mathbf{x}) \\ L_f^{(r2)}h_2(\mathbf{x}) \\ \vdots \\ L_f^{(rm)}h_m(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \sum_{j=1}^{m}L_{gj}(L_f^{(r1-1)}h_1(\mathbf{x}))u_j \\ \sum_{j=1}^{m}L_{gj}(L_f^{(r2-1)}h_2(\mathbf{x}))u_j \\ \vdots \\ \sum_{j=1}^{m}L_{gj}(L_f^{(rm-1)}h_m(\mathbf{x}))u_j \end{pmatrix} \\
&= \begin{pmatrix} L_f^{(r1)}h_1(\mathbf{x}) \\ L_f^{(r2)}h_2(\mathbf{x}) \\ \vdots \\ L_f^{(rm)}h_m(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} L_{g1}(L_f^{(r1-1)}h_1(\mathbf{x})) & \ldots & L_{gm}(L_f^{(r1-1)}h_1(\mathbf{x})) \\ L_{g1}(L_f^{(r2-1)}h_2(\mathbf{x})) & & L_{gm}(L_f^{(r2-1)}h_2(\mathbf{x})) \\ \vdots & & \vdots \\ L_{g1}(L_f^{(rm-1)}h_m(\mathbf{x})) & \ldots & L_{gm}(L_f^{(rm-1)}h_m(\mathbf{x})) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}
\end{aligned} \tag{4}
$$

or in compact form

$$
\mathbf{y}^{(r)}(t) = \mathbf{A}(\mathbf{x})+\mathbf{B}(\mathbf{x})\mathbf{u}(t) \tag{5}
$$

where

$$
\mathbf{y}^{(r)}(t) = \begin{pmatrix} y_1^{(r1)} \\ y_2^{(r2)} \\ \vdots \\ y_m^{(rm)} \end{pmatrix}; \quad \mathbf{A}(\mathbf{x}) = \begin{pmatrix} L_f^{(r1)}h_1(\mathbf{x}) \\ L_f^{(r2)}h_2(\mathbf{x}) \\ \vdots \\ L_f^{(rm)}h_m(\mathbf{x}) \end{pmatrix};
$$

$$
\mathbf{B}(\mathbf{x})\mathbf{u}(t) = \begin{pmatrix} L_{g1}(L_f^{(r1-1)}h_1(\mathbf{x})) & \ldots & L_{gm}(L_f^{(r1-1)}h_1(\mathbf{x})) \\ L_{g1}(L_f^{(r2-1)}h_2(\mathbf{x})) & & L_{gm}(L_f^{(r2-1)}h_2(\mathbf{x})) \\ \vdots & & \vdots \\ L_{g1}(L_f^{(rm-1)}h_m(\mathbf{x})) & \ldots & L_{gm}(L_f^{(rm-1)}h_m(\mathbf{x})) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}
$$

Assumption 4. The matrix **B**, as defined above, is non-singular, i.e., $\mathbf{B}^{-1}$ exists and has bounded norm for all $\boldsymbol{x}\in S_{\boldsymbol{x}}, t\geq 0$, where $S_{\boldsymbol{x}}\in\Re^n$ is some compact set of allowable state trajectories. This is equivalent to assuming

$$
\begin{aligned}
&\lambda_p(\mathbf{B}) > \lambda_{\min} > 0 \\
&||\mathbf{B}||_2 = \lambda_1(\mathbf{B}) \leq \lambda_{\max} < \infty
\end{aligned} \tag{6}
$$

where $\lambda_p(\boldsymbol{B})$ and $\lambda_1(\boldsymbol{B})$ are the smallest and largest singular values of **B**, respectively. In addition, in order to be able to guarantee state boundedness under state feedback linearisation, the following assumption is required.

Assumption 5. The plant is feedback linearisable by static state feedback; it has a general vector relative degree and its zero dynamics are exponentially attractive (Isidori, 1995) and the state vector is available for measurement. The output tracking error is defined as

$$
\mathbf{e} = \mathbf{y}-\mathbf{y}_r = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} - \begin{pmatrix} y_{ref1} \\ y_{ref2} \\ \vdots \\ y_{refm} \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{pmatrix} \tag{7}
$$

4. Design of adaptive neural sliding mode compensation

The control objective is to design an adaptive neural controller that guarantees boundedness of all closed-loop variables and tracking of a given bounded reference signal vector y_r. The adaptive neural sliding mode method, which is based on the NN-RBF model, can solve this kind of control problem (Tsai et al., 2004).

The state tracking error is defined as $\mathbf{e}(t)=\mathbf{y}(t)-\mathbf{y_r}(t)$ and the control objective is to find a control law such that output $\mathbf{y}$ follows the desired trajectory $\mathbf{y}_r$, in other words, the tracking error should converge to zero.

A sliding surface $\mathbf{S}(t)$ for the MIMO system can be defined in terms of the control error:

$$\mathbf{S}(t)=\begin{pmatrix}\left(\frac{d^{(r1)}}{dt^{(r1)}}+k_1\right) & \cdots & 0\\ \vdots & & \vdots\\ 0 & \cdots & \left(\frac{d^{(rm)}}{dt^{(rm)}}+k_m\right)\end{pmatrix}\int_0^t \mathbf{e}(\tau)\,d\tau=\begin{pmatrix}e_1^{(r1-1)}(t)+k_1\int_0^t e_1(\tau)\,d\tau\\ e_2^{(r2-1)}(t)+k_2\int_0^t e_2(\tau)\,d\tau\\ \vdots\\ e_m^{(rm-1)}(t)+k_m\int_0^t e_m(\tau)\,d\tau\end{pmatrix} \tag{8}$$

The time derivate of the sliding surface $\mathbf{S}(t)$ is

$$\dot{\mathbf{S}}(t)=\frac{d}{dt}\begin{pmatrix}e_1^{(r1-1)}(t)+k_1\int_0^t e_1(\tau)d\tau\\ e_2^{(r2-1)}(t)+k_2\int_0^t e_2(\tau)d\tau\\ \vdots\\ e_m^{(rm-1)}(t)+k_m\int_0^t e_m(\tau)d\tau\end{pmatrix}=\begin{pmatrix}\frac{d^{(r1)}}{dt^{(r1)}}(y_1-y_{ref1})+k_1e_1\\ \frac{d^{(r2)}}{dt^{(r2)}}(y_2-y_{ref2})+k_2e_2\\ \vdots\\ \frac{d^{(rm)}}{dt^{(rm)}}(y_m-y_{refm})+k_me_m\end{pmatrix} \tag{9}$$

In order to make the system state remain on the sliding surface, let $\dot{\mathbf{S}}(t)=0$,

$$\dot{\mathbf{S}}(t)=\begin{pmatrix}\left(L_f^{(r1)}h_1(\mathbf{x})+\sum_{j=1}^m L_{gj}(L_f^{(r1-1)}h_1(\mathbf{x}))u_j-\frac{d^{(r1)}}{dt^{(r1)}}y_{ref1}\right)+k_1e_1\\ \left(L_f^{(r2)}h_2(\mathbf{x})+\sum_{j=1}^m L_{gj}(L_f^{(r2-1)}h_2(\mathbf{x}))u_j--\frac{d^{(r2)}}{dt^{(r2)}}y_{ref2}\right)+k_2e_2\\ \vdots\\ \left(L_f^{(rm)}h_m(\mathbf{x})+\sum_{j=1}^m L_{gj}(L_f^{(rm-1)}h_m(\mathbf{x}))u_j--\frac{d^{(rm)}}{dt^{(rm)}}y_{refm}\right)+k_me_m\end{pmatrix}=\begin{pmatrix}0\\0\\ \vdots\\0\end{pmatrix} \tag{10}$$

Rearranging (10),

$$\begin{pmatrix}L_{g1}(L_f^{(r1-1)}h_1(\mathbf{x})) & L_{g2}(L_f^{(r1-1)}h_1(\mathbf{x})) & \cdots & L_{gm}(L_f^{(r1-1)}h_1(\mathbf{x}))\\ L_{g1}(L_f^{(r2-1)}h_2(\mathbf{x})) & L_{g2}(L_f^{(r2-1)}h_2(\mathbf{x})) & \cdots & L_{gm}(L_f^{(r2-1)}h_2(\mathbf{x}))\\ \vdots & \vdots & & \vdots\\ L_{g1}(L_f^{(rm-1)}h_m(\mathbf{x})) & L_{g2}(L_f^{(rm-1)}h_m(\mathbf{x})) & \cdots & L_{gm}(L_f^{(rm-1)}h_m(\mathbf{x}))\end{pmatrix}\begin{pmatrix}u_1\\u_2\\ \vdots\\u_m\end{pmatrix}$$
$$=\begin{pmatrix}\frac{d^{(r1)}}{dt^{(r1)}}y_{ref1}-L_fh_1(\mathbf{x})-k_1e_1\\ \frac{d^{(r2)}}{dt^{(r2)}}y_{ref2}-L_fh_2(\mathbf{x})-k_2e_2\\ \vdots\\ \frac{d^{(rm)}}{dt^{(rm)}}y_{refm}-L_fh_m(\mathbf{x})-k_me_m\end{pmatrix} \tag{11}$$

the corresponding equivalent control law $\mathbf{u}(t)$ is expressed as follows:

$$\begin{pmatrix}u_1\\u_2\\ \vdots\\u_m\end{pmatrix}=\begin{pmatrix}L_{g1}(L_f^{(r1-1)}h_1(\mathbf{x})) & L_{g2}(L_f^{(r1-1)}h_1(\mathbf{x})) & \cdots & L_{gm}(L_f^{(r1-1)}h_1(\mathbf{x}))\\ L_{g1}(L_f^{(r2-1)}h_2(\mathbf{x})) & L_{g2}(L_f^{(r2-1)}h_2(\mathbf{x})) & & \vdots\\ \vdots & & & \vdots\\ L_{g1}(L_f^{(rm-1)}h_m(\mathbf{x})) & L_{g2}(L_f^{(rm-1)}h_m(\mathbf{x})) & \cdots & L_{gm}(L_f^{(rm-1)}h_m(\mathbf{x}))\end{pmatrix}^{-1}$$
$$\times\begin{pmatrix}-L_fh_1(\mathbf{x})+\frac{d^{(r1)}}{dt^{(r1)}}y_{ref1}-k_1e_1\\ -L_fh_2(\mathbf{x})+\frac{d^{(r2)}}{dt^{(r2)}}y_{ref2}-k_2e_2\\ \vdots\\ -L_fh_m(\mathbf{x})+\frac{d^{(rm)}}{dt^{(rm)}}y_{refm}-k_me_m\end{pmatrix} \tag{12}$$

Now, the problem of controlling the uncertain nonlinear system (4), defining a control law $\mathbf{u}^*$ that guarantees the sliding condition treated in (Tsai et al., 2004), is composed of an equivalent control (12) and a discontinuous term $u_S=-\mathbf{\Gamma}\ \mathrm{sign}(\mathbf{S})$ defined by

$$\begin{pmatrix}u_1\\u_2\\ \vdots\\u_m\end{pmatrix}=\begin{pmatrix}L_{g1}(L_f^{(r1-1)}h_1(\mathbf{x})) & L_{g2}(L_f^{(r1-1)}h_1(\mathbf{x})) & \cdots & L_{gm}(L_f^{(r1-1)}h_1(\mathbf{x}))\\ L_{g1}(L_f^{(r2-1)}h_2(\mathbf{x})) & L_{g2}(L_f^{(r2-1)}h_2(\mathbf{x})) & & \vdots\\ \vdots & & & \vdots\\ L_{g1}(L_f^{(rm-1)}h_m(\mathbf{x})) & L_{g2}(L_f^{(rm-1)}h_m(\mathbf{x})) & \cdots & L_{gm}(L_f^{(rm-1)}h_m(\mathbf{x}))\end{pmatrix}^{-1}\cdots$$
$$\left[\begin{pmatrix}-L_fh_2(\mathbf{x})+\frac{d^{(r1)}}{dt^{(r1)}}y_{ref1}-k_1e_1\\ -L_fh_2(\mathbf{x})+\frac{d^{(r2)}}{dt^{(r2)}}y_{ref2}-k_2e_2\\ \vdots\\ -L_fh_m(\mathbf{x})+\frac{d^{(rm)}}{dt^{(rm)}}y_{refm}-k_me_m\end{pmatrix}-\begin{pmatrix}\Gamma_1 & 0 & \cdots & 0\\ 0 & \Gamma_2 & & 0\\ \vdots & & & \vdots\\ 0 & 0 & \cdots & \Gamma_m\end{pmatrix}\begin{pmatrix}\mathrm{sign}(S_1)\\ \mathrm{sign}(S_2)\\ \vdots\\ \mathrm{sign}(S_m)\end{pmatrix}\right] \tag{13}$$

where Γ_i is a given positive constant and $\mathrm{sign}(S_i)$ is defined by

$$\mathrm{sign}(S_i)=\begin{cases}1 & \text{for}\quad S_i>0\\ 0 & \text{for}\quad S_i=0\\ -1 & \text{for}\quad S_i<0\end{cases} \tag{14}$$

Now, considering the Lyapunov function candidate defined as

$$V=\sum_{i=m}\frac{1}{2}(S_i^2) \tag{15}$$

and differentiating (15) with respect to time along the system trajectory as

$$\dot{V}=\sum_{i=1}^m S_i\dot{S}_i=\sum_{i=1}^m S_i\left(\frac{d^{(ri)}}{dt^{(ri)}}e_i(t)+k_ie_i(t)\right)=\sum_{i=1}^m S_i\left(\frac{d^{(ri)}}{dt^{(ri)}}(y_i-y_{refi})+k_ie_i(t)\right)=\ldots$$
$$=\sum_{i=1}^m S_i\left(L_fh_i(\mathbf{x})+\sum_{j=1}^m L_{gj}(L_f^{(r1-1)}h_i(\mathbf{x}))u_j-\frac{d^{(ri)}}{dt^{(ri)}}y_{refi}+k_ie_i(t)\right) \tag{16}$$

Replacing (13) in (16),

$$\dot{V}=\sum_{i=1}^m S_i\dot{S}_i=\sum_{i=1}^m S_i(-\Gamma_i\ \mathrm{sgn}(S_i))\le-\sum_{i=1}^m\Gamma_i|S_i| \tag{17}$$

Then, dividing every term in (17) by $|S_i|$ and integrating both sides over the interval $0\le t\le t_S$, where t_S is the time required to reach the surface $\mathbf{S}$, the following is obtained:

$$\int_0^{t_S}\left(\frac{S_i}{|S_i|}\dot{S}_i\right)dt\le-\int_0^{t_S}\Gamma_i\,dt$$

which implies that

$$|S_i(t_S)|-|S_i(0)|\le-\Gamma_it_S \tag{18}$$

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

$$t_S\le\frac{|S_i(0)|}{\Gamma_i} \tag{19}$$

and consequently, a finite time convergence to sliding surface $\mathbf{S}$. Now, considering the model indicated in (1) with parameter uncertainties, the unmodelled structure and external disturbances can be expressed as

$$\begin{pmatrix}y_1^{(r1)}\\y_2^{(r2)}\\ \vdots\\y_m^{(rm)}\end{pmatrix}=\begin{pmatrix}L_fh_1(\mathbf{x})+\Delta L_fh_1(\mathbf{x})\\ L_fh_2(\mathbf{x})+\Delta L_fh_2(\mathbf{x})\\ \vdots\\ L_fh_m(\mathbf{x})+\Delta L_fh_m(\mathbf{x})\end{pmatrix}$$
$$+\begin{pmatrix}\sum_{j=1}^m L_{gj}(L_f^{(r1-1)}h_1(\mathbf{x}))u_j+\ldots+\sum_{j=1}^m\Delta L_{gj}(L_f^{(r1-1)}h_1(\mathbf{x}))u_j\\ \sum_{j=1}^m L_{gj}(L_f^{(r2-1)}h_1(\mathbf{x}))u_j+\ldots+\sum_{j=1}^m\Delta L_{gj}(L_f^{(r2-1)}h_1(\mathbf{x}))u_j\\ \vdots \qquad\qquad \vdots\\ \sum_{j=1}^m L_{gj}(L_f^{(rm-1)}h_1(\mathbf{x}))u_j+\ldots+\sum_{j=1}^m\Delta L_{gj}(L_f^{(rm-1)}h_1(\mathbf{x}))u_j\end{pmatrix} \tag{20}$$

The relation between $\tilde{f}(\mathbf{x})$, $\tilde{g}(\mathbf{x})$, $\tilde{h}(\mathbf{x})$, $\Delta L_f h_i(\mathbf{x})$ and $\Delta L_{gj} h_i(\mathbf{x})$ are indicated by

$$L_f h_i(\mathbf{x}) + \Delta L_f h_i(\mathbf{x}) = \frac{\partial h_i(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) + \frac{\partial \tilde{h}_i(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x}) + \frac{\partial h_i(\mathbf{x})}{\partial \mathbf{x}} \tilde{f}(\mathbf{x}) + \frac{\partial \tilde{h}_i(\mathbf{x})}{\partial \mathbf{x}} \tilde{f}(\mathbf{x})$$

$$L_f^2 h_i(\mathbf{x}) + \Delta L_f^2 h_i(\mathbf{x}) = \frac{\partial (L_f h_i(\mathbf{x}) + \Delta L_f h_i(\mathbf{x}))}{\partial \mathbf{x}} [f(\mathbf{x}) + \tilde{f}(\mathbf{x})]$$

$$\vdots$$

$$L_f^{(rm)} h_i(\mathbf{x}) + \Delta L_f^{(rm)} h_i(\mathbf{x}) = \frac{\partial (L_f^{(rm-1)} h_i(\mathbf{x}) + \Delta L_f^{(rm-1)} h_i(\mathbf{x}))}{\partial \mathbf{x}} [f(\mathbf{x}) + \tilde{f}(\mathbf{x})] \tag{21}$$

and in the same way,

$$L_{gj}(L_f^{(ri-1)} h_1(\mathbf{x}) + \Delta L_f^{(ri)} h_i(\mathbf{x})) = L_{gj}(L_f^{(ri-1)} h_1(\mathbf{x})) + \Delta L_{gj}(L_f^{(ri)} h_i(\mathbf{x}))$$
$$= \frac{\partial (L_f^{(ri-1)} h_i(\mathbf{x}) + \Delta L_f^{(ri-1)} h_i(\mathbf{x}))}{\partial \mathbf{x}} [g_j(\mathbf{x}) + \tilde{g}_j(\mathbf{x})] \tag{22}$$

Replacing the proposed control action of (13) in (20) gives

$$\frac{d}{dt}\mathbf{S}(t) = \begin{pmatrix} -k_1 e_1 + \Delta L_f h_1(\mathbf{x}) + \sum_{j=1}^{m} \Delta L_{gj}(L_f^{(r1-1)} h_1(\mathbf{x})) u_j - \Gamma_1 \operatorname{sign}(S_1) \\ -k_2 e_2 + \Delta L_f h_2(\mathbf{x}) + \sum_{j=1}^{m} \Delta L_{gj}(L_f^{(r1-1)} h_2(\mathbf{x})) u_j - \Gamma_2 \operatorname{sign}(S_2) \\ \vdots \\ -k_m e_m + \Delta L_f h_m(\mathbf{x}) + \sum_{j=1}^{m} \Delta L_{gj}(L_f^{(r1-1)} h_m(\mathbf{x})) u_j - \Gamma_m \operatorname{sign}(S_m) \end{pmatrix} \tag{23}$$

Usually, the sliding mode control is considered to improve the robustness of the closed loop system when it has external disturbances and modelling uncertainties. In electromechanical systems, the "chattering" effect in the control action is undesirable, because it could produce a mechanical resonance causing damage to the system. To solve this problem, a term $v_{iN}(t)$ should be added in the control action to compensate the uncertainties and disturbances on the system and to reduce simultaneously the "chattering", which is highly detrimental for closed loop control.

From (13), the adaptive sliding control compensation is expressed as

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} L_{g1}(L_f^{(r1-1)} h_1(\mathbf{x})) & L_{g2}(L_f^{(r1-1)} h_1(\mathbf{x})) & \cdots & L_{gm}(L_f^{(r1-1)} h_1(\mathbf{x})) \\ L_{g1}(L_f^{(r2-1)} h_2(\mathbf{x})) & L_{g2}(L_f^{(r2-1)} h_2(\mathbf{x})) & & \vdots \\ \vdots & & & \vdots \\ L_{g1}(L_f^{(rm-1)} h_m(\mathbf{x})) & L_{g2}(L_f^{(rm-1)} h_m(\mathbf{x})) & \cdots & L_{gm}(L_f^{(rm-1)} h_m(\mathbf{x})) \end{pmatrix}^{-1}$$
$$\times \left[\begin{pmatrix} -L_f h_2(\mathbf{x}) + \frac{d^{(r1)}}{dt^{(r1)}} y_{ref1} - k_1 e_1 \\ -L_f h_2(\mathbf{x}) + \frac{d^{(r2)}}{dt^{(r2)}} y_{ref2} - k_2 e_2 \\ \vdots \\ -L_f h_m(\mathbf{x}) + \frac{d^{(rm)}}{dt^{(rm)}} y_{refm} - k_m e_m \end{pmatrix} - \begin{pmatrix} v_{1N} \\ v_{2N} \\ \vdots \\ v_{mN} \end{pmatrix} - \begin{pmatrix} \Gamma_1 \operatorname{sign}(S_1) \\ \Gamma_2 \operatorname{sign}(S_2) \\ \vdots \\ \Gamma_m \operatorname{sign}(S_m) \end{pmatrix} \right] \tag{24}$$

where v_{iN} is a compensation variable that can be approximated by RBF-NN; the parameters being tuned on-line.

$$v_{iN}^* = \mathbf{w}_{ij}^{*T} \xi_j^*(\mathbf{x}, \mathbf{c}^*, \boldsymbol{\eta}^*) + \sum_{j=1}^{m} \boldsymbol{\varphi}_{ij}^* \xi_j^*(\mathbf{x}, \mathbf{c}_j^*, \boldsymbol{\eta}_j^*) u_j + \varepsilon_{in} \quad i = 1, 2, 3..m \tag{25}$$

$$\frac{d}{dt}\begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_m \end{pmatrix} = \begin{pmatrix} -k_1 e_1 - \tilde{\mathbf{w}}_1^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_1^T \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) - \sum_{j=1}^{m} [\tilde{\boldsymbol{\varphi}}_{1j} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{1j} \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j) u_j] - \varepsilon_{1n} \\ -k_2 e_2 - \tilde{\mathbf{w}}_2^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_2^T \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) - \sum_{j=1}^{m} [\tilde{\boldsymbol{\varphi}}_{2j} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{2j} \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j) u_j] - \varepsilon_{2n} \\ \vdots \\ -k_m e_m - \tilde{\mathbf{w}}_m^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_m^T \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) - \sum_{j=1}^{m} [\tilde{\boldsymbol{\varphi}}_{mj} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{mj} \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j) u_j] - \varepsilon_{mn} \end{pmatrix} - \begin{pmatrix} \Gamma_1 \operatorname{sign}(S_1) \\ \Gamma_2 \operatorname{sign}(S_2) \\ \vdots \\ \Gamma_m \operatorname{sign}(S_m) \end{pmatrix} \tag{31}$$

where $\mathbf{w}^*(\sigma \times m)$, $\boldsymbol{\varphi}^*(\sigma \times m)$ and $\xi^*(\sigma \times 1)$ are optimal parameter vectors of weight $\mathbf{w}$, the input weights vector $\boldsymbol{\varphi}$ and the radial basis functions ξ, respectively; $\mathbf{c}^*$ and $\boldsymbol{\eta}^*$ are optimal parameter vectors of centres $\mathbf{c}$ and widths $\boldsymbol{\eta}$, respectively; σ is the neurons number and $\boldsymbol{\varepsilon_n}$ is the approximation error.

In practice, the functions (unmodelled uncertainties in our case) (21) can be approximated correctly in a compact set by using a sufficient number of RBF neurons (Park and Sandberg, 1991).

Assumption 6. $\Delta L_f h_i(\mathbf{x})$ and $\Delta L_{gj}(L_f^{(ri-1)} h_i(\mathbf{x}))$ functions can be approximated by the output of an RBF-NN (Cybenko, 1989) with the approximation error bounded by

$$|\Delta L_f h_i(\mathbf{x}) - \mathbf{w}_i^{*T} \xi^*(\mathbf{x}, \mathbf{c}^*, \boldsymbol{\eta}^*)| + \left| \sum_{j=1}^{m} \Delta L_{gj}(L_f^{(ri-1)} h_i(\mathbf{x})) u_j - \sum_{j=1}^{m} \boldsymbol{\varphi}_{ij}^* \xi_j^*(\mathbf{x}, \mathbf{c}_j^*, \boldsymbol{\eta}_j^*) u_j \right|$$
$$\le \varepsilon_{in} \quad \forall \mathbf{x} \in \mathbb{R}^m \tag{26}$$

where $\mathbf{x}$ is the input vector to the RBF-NN, $\varepsilon_{Max} \ge \varepsilon_{in} > 0$ is the bound of the approximation error, $\mathbf{w}^*$ and $\boldsymbol{\varphi}^*$, are the output optimal weight vector, $l > 1$ is the number of the NN nodes and $\xi(\mathbf{x}) = [\xi_1(\mathbf{x}), \xi_2(\mathbf{x}) \ldots \xi_l(\mathbf{x})]^T$ is defined by

$$\xi_i^*(\mathbf{x}, \mathbf{c}^*, \eta^*) = \exp[-\eta_i^{*2} (\mathbf{x} - \mathbf{c}_i^*)^T (\mathbf{x} - \mathbf{c}_i^*)^T] \tag{27}$$

with $c^* = [c_1^* \quad c_2^* \quad \ldots \quad c_n^*]^T$ in the centre of the receptive field and η^* the width of the Gaussian function.

Assumption 7. The function approximation weights $\mathbf{w}$ and $\boldsymbol{\varphi}$ are bounded:

$$w_{Max} = \sup_{t \in \mathfrak{R}^+} \|\mathbf{w}(t)\| \quad \varphi_{Max} = \sup_{t \in \mathfrak{R}^+} \|\boldsymbol{\varphi}(t)\| \tag{28}$$

The optimal parameters of (28) are unknown and therefore, it is necessary to estimate their values. Defining an estimation function:

$$\hat{v}_{iN} = \hat{\mathbf{w}}_i^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \sum_{j=v}^{\omega} \hat{\boldsymbol{\varphi}}_{ij} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\boldsymbol{\eta}}_j) u_j + \varepsilon_{in} \quad i = 1, 2, .., m \tag{29}$$

where $\hat{\mathbf{w}}$, $\hat{\xi}$ and $\hat{\boldsymbol{\varphi}}$ are estimated parameter vectors of $\mathbf{w}$, ξ and $\boldsymbol{\varphi}$, respectively and $\hat{\mathbf{c}}$ and $\hat{\boldsymbol{\eta}}$ are estimated parameter vectors of c and η, respectively.

Defining $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$, $\tilde{\xi} = \xi^* - \hat{\xi}$ and $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}^* - \hat{\boldsymbol{\varphi}}$, the neural compensation $\hat{\mathbf{v}}_N$ can be written as

$$\hat{\mathbf{v}}_{iN} = \hat{\mathbf{w}}_i^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \tilde{\mathbf{w}}_i^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \hat{\mathbf{w}}_i^T \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) + \tilde{\mathbf{w}}_i^T \tilde{\xi}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) + \ldots$$
$$+ \sum_{j=1}^{m} [\hat{\boldsymbol{\varphi}}_{ij} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \tilde{\boldsymbol{\varphi}}_{ij} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{ij} \tilde{\xi}_j(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j) u_j$$
$$+ \tilde{\boldsymbol{\varphi}}_{ij} \tilde{\xi}_j(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j) u_j] + \varepsilon_{in} \tag{30}$$

where $\tilde{\mathbf{w}}^T \tilde{\xi} + \hat{\mathbf{w}}^T \tilde{\xi}$ represents the learning error and it is considered $\tilde{\mathbf{w}}^T \tilde{\xi}$ and $\sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \tilde{\xi}_j u_j$ into ε_{in}.

Now, it is expressed $\hat{\mathbf{w}}_i \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) = \Delta L_f h_i(\mathbf{x})$ and $\sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) u_j = \sum_{j=1}^{m} \Delta L_{gj}(L_f^{(r_i-1)} h_i(\mathbf{x})) u_j$

Combining control law (24) and neural compensation (30) into the robotic model (20), the closed loop error equation becomes

Using an approximation for the function $\tilde{\xi} = \xi^*(\mathbf{x}, \mathbf{c}^*, \boldsymbol{\eta}^*) - \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}})$, in order to deal with ξ^*, Taylor's expansion of $\tilde{\xi}$ is taken from $\mathbf{c}^* = \hat{\mathbf{c}}$ and $\boldsymbol{\eta}^* = \hat{\boldsymbol{\eta}}$.

$$\xi^*(\mathbf{x}, \mathbf{c}^*, \boldsymbol{\eta}^*) = \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \boldsymbol{\Xi}^T \tilde{\mathbf{c}} + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) \tag{32}$$

where $\mathbf{O}$ denotes the high-order arguments in a Taylor's series expansion and Ξ and Φ are derivatives of $\xi^*(\mathbf{x}, \mathbf{c}^*, \boldsymbol{\eta}^*)$ with respect to $\mathbf{c}^*$ and $\boldsymbol{\eta}^*$ at $(\hat{\mathbf{c}}, \hat{\boldsymbol{\eta}})$. They are expressed as

$$\begin{cases} \Xi^T = \left.\dfrac{\partial \xi(\mathbf{x}, \mathbf{c}^*, \eta^*)}{\partial \mathbf{c}^*}\right|_{\substack{\mathbf{c}^* = \hat{\mathbf{c}} \\ \boldsymbol{\eta}^* = \hat{\boldsymbol{\eta}}}} \\ \\ \Phi^T = \left.\dfrac{\partial \xi(\mathbf{x}, \mathbf{c}^*, \eta^*)}{\partial \boldsymbol{\eta}^*}\right|_{\substack{\mathbf{c}^* = \hat{\mathbf{c}} \\ \boldsymbol{\eta}^* = \hat{\boldsymbol{\eta}}}} \end{cases} \tag{33}$$

Eq. (32) can be expressed as

$$\tilde{\xi} = \boldsymbol{\Xi}^T \tilde{\mathbf{c}} + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) \tag{34}$$

From (34), the high-order term $\mathbf{O}$ is bounded by

$$\begin{aligned} \|\mathbf{O}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\eta})\| &= \|\tilde{\xi} - \boldsymbol{\Xi}^T \tilde{\mathbf{c}} - \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}\| \le \|\tilde{\xi}\| + \|\boldsymbol{\Xi}^T \tilde{\mathbf{c}}\| + \|\boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}\| \\ &\le \kappa_1 + \kappa_2 \|\tilde{\mathbf{c}}\| + \kappa_3 \|\tilde{\boldsymbol{\eta}}\| \le \mathbf{O}_{Max} \end{aligned} \tag{35}$$

where κ_1, κ_2 and κ_3 are some constants owing to the fact that RBF and its derivative are always bounded by constants. Substituting (34) into (31) gives

$$\frac{d}{dt}\begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_m \end{pmatrix} = \begin{pmatrix} -k_1 e_1 - \tilde{\mathbf{w}}_1^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_1^T (\boldsymbol{\Xi}^T \tilde{\mathbf{c}} + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}) - \sum_{j=1}^{m} \left[\tilde{\boldsymbol{\varphi}}_{1j} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{1j} (\boldsymbol{\Xi}^T \tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j) u_j\right] - \varepsilon_{1n} \\ -k_2 e_2 - \tilde{\mathbf{w}}_2^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_2^T (\boldsymbol{\Xi}^T \tilde{\mathbf{c}} + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}) - \sum_{j=1}^{m} \left[\tilde{\boldsymbol{\varphi}}_{2j} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{2j} (\boldsymbol{\Xi}^T \tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j) u_j\right] - \varepsilon_{2n} \\ \vdots \\ -k_m e_m - \tilde{\mathbf{w}}_m^T \hat{\xi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_m^T (\boldsymbol{\Xi}^T \tilde{\mathbf{c}} + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}) - \sum_{j=1}^{m} [\tilde{\boldsymbol{\varphi}}_{mj} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j + \hat{\boldsymbol{\varphi}}_{mj} (\boldsymbol{\Xi}^T \tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j) u_j] - \varepsilon_{mn} \end{pmatrix} - \begin{pmatrix} \Gamma_1 sign(S_1) \\ \Gamma_2 sign(S_2) \\ \vdots \\ \Gamma_m sign(S_m) \end{pmatrix} \tag{36}$$

where the uncertain $\mathbf{w}_i^T \mathbf{O} + \sum_{j=1}^{m} (\hat{\boldsymbol{\varphi}}_{ij}^T \mathbf{O}_j u_j) + \varepsilon_{in}$ is assumed to be bounded by

$$|\varepsilon_{Max}| = \left|\hat{\mathbf{w}}_i^T \mathbf{O} + \sum_{j = v,\omega} (\hat{\boldsymbol{\varphi}}_{ij}^T \mathbf{O}_j u_j) + \varepsilon_{in}\right| \le \Gamma_I \quad i = 1, 2, .., m \tag{37}$$

5. Stability analysis and neural parameters adjustment

To derive a stable tuning law, the following Lyapunov function is chosen taking into account the error, neural weights, spreads and centres.

$$V = \frac{1}{2} \sum_{i=x}^{y} \left[p_I S_i^2 + \tilde{\mathbf{w}}_i^T \theta_i \tilde{\mathbf{w}} + \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij}^T \rho_i \tilde{\boldsymbol{\varphi}}_j\right] + \frac{1}{2}(\tilde{\mathbf{c}}^T \Lambda_1 \tilde{\mathbf{c}} + \tilde{\boldsymbol{\eta}}^T \Lambda_2 \tilde{\boldsymbol{\eta}}) \tag{38}$$

where $\mathbf{P}$ is an $m \times m$ diagonal positive definite matrix and θ_i and $\Lambda_{1,2}$ are $dim(\mathbf{c}) \times dim(\mathbf{c})$ and $dim(\boldsymbol{\eta}) \times dim(\boldsymbol{\eta})$ non-negative definite matrices, respectively. The derivative of the Lyapunov function is given by

$$\frac{dV}{dt} = \sum_i \left[p_i S_i \frac{dS_i}{dt} + \tilde{\mathbf{w}}_i^T \theta_i \frac{d\tilde{\mathbf{w}}_i}{dt} + \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij}^T \rho_{ij} \frac{d\tilde{\boldsymbol{\varphi}}_{ij}}{dt}\right] + \left(\frac{d\tilde{\mathbf{c}}^T}{dt} \Lambda_1 \tilde{\mathbf{c}} + \frac{d\tilde{\boldsymbol{\eta}}^T}{dt} \Lambda_2 \tilde{\boldsymbol{\eta}}\right) \tag{39}$$

Substituting (36) into (39) and considering that $\boldsymbol{K^T P} = (\boldsymbol{K^T P})^{\boldsymbol{T}}$, $\boldsymbol{P}$ and $\boldsymbol{K}$ are diagonal matrices and defining $\boldsymbol{Q} = \boldsymbol{K^T P}$, (39) can be written as

$$\begin{aligned} \frac{dV}{dt} = \sum_{i=1}^{m} \Big[& -q_i S_i^2 - p_i S_i \tilde{\mathbf{w}}_i^T \hat{\xi} - p_i S_i \hat{\mathbf{w}}_i^T \boldsymbol{\Xi}^T \tilde{\mathbf{c}} - p_i S_i \hat{\mathbf{w}}_i^T \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} - p_i S_i \hat{\mathbf{w}}_i^T \mathbf{O} \\ & - p_i S_i \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij} \hat{\xi}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j) u_j - \ldots - p_i S_i \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} (\boldsymbol{\Xi}^T \tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j) u_j \\ & - p_i S_i \varepsilon_{in} - p_i S_i \Gamma_i \, \text{sign}(S_i) + \tilde{\mathbf{w}}_i^T \theta_i \frac{d\tilde{\mathbf{w}}_i}{dt} + \sum_{j=v}^{\omega} \tilde{\boldsymbol{\varphi}}_{ij}^T \rho_{ij} \frac{d\tilde{\boldsymbol{\varphi}}_{ij}}{dt} \Big] + \ldots \\ & + \left(\frac{d\tilde{\mathbf{c}}^T}{dt} \Lambda_1 \tilde{\mathbf{c}} + \frac{d\tilde{\boldsymbol{\eta}}^T}{dt} \Lambda_2 \tilde{\boldsymbol{\eta}}\right) \end{aligned} \tag{40}$$

Rearranging (40),

$$\begin{aligned} \frac{dV}{dt} = \sum_i \Big[& -q_i S_i^2 - p_i S_i \tilde{\mathbf{w}}_i^T \hat{\xi} - p_i S_i \hat{\mathbf{w}}_i^T \boldsymbol{\Xi}^T \tilde{\mathbf{c}} - p_i S_i \hat{\mathbf{w}}_i^T \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}} - p_i S_i \hat{\mathbf{w}}_i^T \mathbf{O} \\ & + \tilde{\mathbf{w}}_i^T \theta_i \frac{d\tilde{\mathbf{w}}_i}{dt} + \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij}^T \rho_{ij} \frac{d\tilde{\boldsymbol{\varphi}}_{ij}}{dt} + \ldots - p_i S_i \varepsilon_{in} - p_i S_i \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \boldsymbol{\Xi}^T \tilde{\mathbf{c}}_j u_j \\ & - p_i S_i \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \boldsymbol{\Phi}^T \tilde{\boldsymbol{\eta}}_j u_j - p_i S_i \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \mathbf{O}_j u_j - p_i S_i \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij}^T \hat{\xi}_j u_j - p_i \Gamma_i |S_i| \Big] \\ & + \left(\frac{d\tilde{\mathbf{c}}^T}{dt} \Lambda_1 \tilde{\mathbf{c}} + \frac{d\tilde{\boldsymbol{\eta}}^T}{dt} \Lambda_2 \tilde{\boldsymbol{\eta}}\right) \end{aligned} \tag{41}$$

Rearranging and grouping the terms,

$$\begin{aligned} \frac{dV}{dt} = \sum_i \Big[& -q_i S_i^2 + \tilde{\mathbf{w}}_i^T \left(-p_i S_i \hat{\xi} + \theta_i \frac{d\tilde{\mathbf{w}}_i}{dt}\right) + \sum_{j=1}^{m} \tilde{\boldsymbol{\varphi}}_{ij}^T \left(-p_i S_i \hat{\xi}_j u_j + \rho_{ij} \frac{d\tilde{\boldsymbol{\varphi}}_{ij}}{dt}\right) - \ldots \\ & - p_i S_i \left(\hat{\mathbf{w}}_i^T \mathbf{O} + \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij} \mathbf{O}_j u_j + \varepsilon_{in}\right) - p_i \Gamma_i |S_i| \Big] \\ & + \left(-\sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Phi}^T + \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Phi}^T u_j\right] + \frac{d\tilde{\boldsymbol{\eta}}^T}{dt} \Lambda_2\right) \tilde{\boldsymbol{\eta}} + \ldots \\ & + \left(-\sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Xi}^T + \sum_{j=1}^{m} \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Xi}^T u_j\right] + \frac{d\tilde{\mathbf{c}}^T}{dt} \Lambda_1\right) \tilde{\mathbf{c}} \end{aligned} \tag{42}$$

Now $\dot{\tilde{\mathbf{c}}}$, $\dot{\tilde{\boldsymbol{\eta}}}$ and $\dot{\tilde{\varphi}}$ are selected as

$$\frac{d\tilde{\mathbf{w}}_i}{dt} = \theta_i^{-1} p_i S_i \hat{\xi} \tag{43}$$

$$\frac{d\tilde{\boldsymbol{\varphi}}_{ij}}{dt} = \rho_{ij}^{-1} p_i S_i \hat{\xi}_j u_j \tag{44}$$

$$\frac{d\tilde{\boldsymbol{\eta}}^T}{dt} = \Lambda_2^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Phi}^T + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Phi}^T u_j\right] \tag{45}$$

$$\frac{d\tilde{\mathbf{c}}^T}{dt} = \Lambda_1^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Xi}^T + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Xi}^T u_j\right] \tag{46}$$

Considering (43–46) into (41), then (41) can be rewritten as

$$\begin{aligned} \frac{dV}{dt} &= \sum_i -q_i S_i^2 - p_i \Gamma_i |S_i| - p_i S_i \left(\hat{\mathbf{w}}_i^T \mathbf{O} + \sum_j \hat{\boldsymbol{\varphi}}_{ij} \mathbf{O}_j u_j + \varepsilon_{in}\right) \\ \frac{dV}{dt} &\le \sum_i (-q_i |S_i|^2 + p_i |\varepsilon_{Max}| |S_i| - p_i \Gamma_i |S_i|) < 0 \end{aligned} \tag{47}$$

From (47), it follows that

$$\frac{dV}{dt} \le -\sum_i q_i |S_i|^2 < 0 \tag{48}$$

Integrating both sides of (48), it can be expressed as

$$\int_0^{t_L} \sum_i q_i |S_i|^2 \, dt \le -\int_0^{t_L} \left(\frac{dV}{dt}\right) dt = V(0) - V(t_L) \tag{49}$$

Because $V(0)$ is bounded and $V(t_L)$ is non-increasing and bounded, it can be obtained that

$$\lim_{t_L \to \infty} \left[-\int_0^{t_L} \left(\frac{dV}{dt}\right) dt\right] = V(0) - V(t_L) < \infty \tag{50}$$

Therefore, by Barbalat's lemma (Slotine and Li, 1991), it can be shown $\lim_{t\to\infty} \; [-dV/dt] = 0$. That is, $S(t) \to 0$ as $t \to \infty$.

As a result, the proposed control system is stable. Moreover, the tracking error of the control system will converge to zero according to $S(t) \to 0$.

From (43) to (46) considering $\dot{\mathbf{w}}_i^* = 0$, $\dot{\mathbf{c}}^* = 0$, $\dot{\boldsymbol{\eta}}^* = 0$ and $\dot{\boldsymbol{\varphi}}^* = 0$, the tuning rules are

$$\frac{d\mathbf{w}_i}{dt} = \theta_i^{-1} p_i S_i \hat{\xi} \tag{51}$$

$$\frac{d\boldsymbol{\varphi}_{ij}}{dt} = \rho_{ij}^{-1} p_i S_i \hat{\xi}_j u_j \tag{52}$$

$$\frac{d\boldsymbol{\eta}^T}{dt} = \Lambda_2^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Phi}^T + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Phi}^T u_j\right] \tag{53}$$

$$\frac{d\mathbf{c}^T}{dt} = \Lambda_1^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^T \boldsymbol{\Xi}^T + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^T \boldsymbol{\Xi}^T u_j\right] \tag{54}$$

6. Application to a robot model

In this section, the dynamic model of the unicycle-like mobile robot presented in Figs. 1 and 2, is reviewed. This figure depicts the mobile robot, which has the following parameters and variables of interest: v and ω are the linear and angular velocities developed by the robot, respectively, G is the centre of mass of the robot, c is the position of the castor wheel, E is the tool location, y is the point of interest with coordinate r_x, r_y in the XY plane, ψ is the robot orientation and 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 as follows.

Kinematic model:

$$\begin{pmatrix} \dot{r}_x(t) \\ \dot{r}_y(t) \\ \dot{\psi}(t) \end{pmatrix} = \begin{pmatrix} \cos\,\psi(t) & -a\,\sin\,\psi(t) \\ \sin\,\psi(t) & a\,\cos\,\psi(t) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v(t) \\ \omega(t) \end{pmatrix} + \begin{pmatrix} \delta_{rx}(t) \\ \delta_{ry}(t) \\ 0 \end{pmatrix} \tag{55}$$

Dynamic model:

$$\begin{pmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{pmatrix} = \begin{pmatrix} \frac{\vartheta_3}{\vartheta_1}\omega^2(t) - \frac{\vartheta_4}{\vartheta_1}v(t) \\ -\frac{\vartheta_5}{\vartheta_2}v(t)\omega(t) - \frac{\vartheta_6}{\vartheta_2}\omega(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{\vartheta_1} & 0 \\ 0 & \frac{1}{\vartheta_2} \end{pmatrix} \begin{pmatrix} u_{vref}(t) \\ u_{\omega ref}(t) \end{pmatrix} + \begin{pmatrix} \delta_v(t) \\ \delta_\omega(t) \end{pmatrix} \tag{56}$$

The vector of the identifying parameters (Table 1) and the vector of the uncertain parameters associated with the mobile robot are

$$\boldsymbol{\vartheta} = [\,\vartheta_1 \quad \vartheta_2 \quad \vartheta_3 \quad \vartheta_4 \quad \vartheta_5 \quad \vartheta_6\,]^T$$
$$\boldsymbol{\delta} = [\,\delta_{rx} \quad \delta_{ry} \quad 0 \quad \delta_v \quad \delta_\omega\,]^T \tag{57}$$

respectively, where δ_{rx} and δ_{ry} are functions of slip velocities and robot orientation, δ_v and δ_ω are functions of physical parameters, such as mass, inertia, wheel and tire diameters, motor and its servo parameters, forces on the wheels and others. These are considered as disturbances.

The robot model presented in (55) and (56) is split into a kinematics and a dynamic part, respectively, as shown in Fig. 3.

Now, from (55) and (56) and by taking into account the complete control law (24), the Lie derivatives are

$$L_{gv}h_x(\mathbf{x}) = \frac{1}{\vartheta_1}\cos\,\psi$$

$$L_{gv}h_y(\mathbf{x}) = -\frac{1}{\vartheta_2}a\,\sin\,\psi$$

$$L_{g\omega}h_x(\mathbf{x}) = \frac{1}{\vartheta_1}\sin\,\psi$$

Fig. 2. Mobile robot Pioneer 2DX.

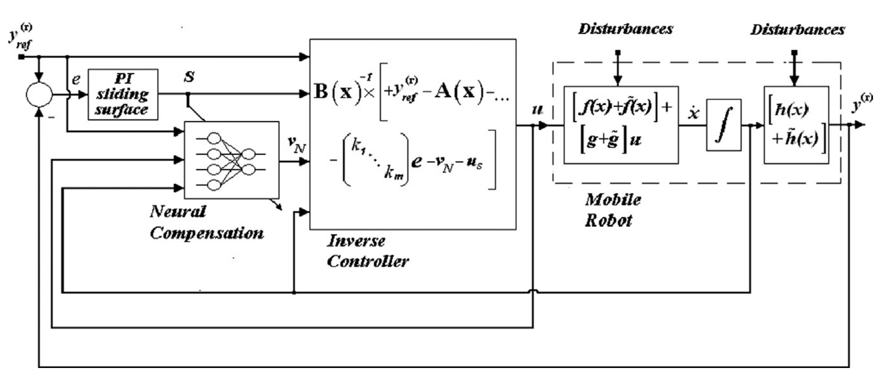

Fig. 1. Control structure.

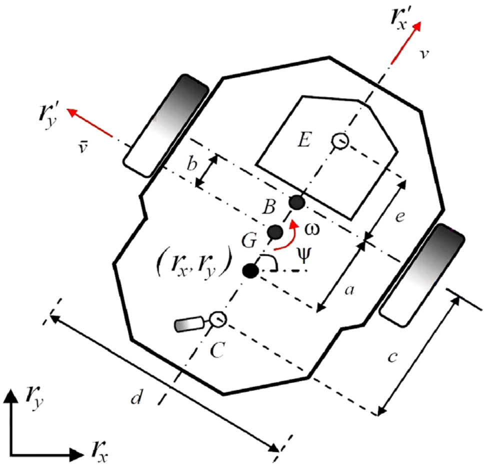

Fig. 3. Mobile robot parameters.

$$L_{g\omega}h_y(\mathbf{x}) = \frac{1}{\vartheta_2} a \cos\psi \tag{58}$$

and

$$\begin{aligned} L_f h_x(\mathbf{x}) &= \left(\tfrac{\vartheta_3}{\vartheta_1}\omega^2 - \tfrac{\vartheta_4}{\vartheta_1}v\right)\cos\psi - \left(-\tfrac{\vartheta_5}{\vartheta_2}v\omega - \tfrac{\vartheta_6}{\vartheta_2}\omega\right) a \sin\psi \\ L_f h_y(\mathbf{x}) &= \left(\tfrac{\vartheta_3}{\vartheta_1}\omega^2 - \tfrac{\vartheta_4}{\vartheta_1}v\right)\sin\psi + \left(-\tfrac{\vartheta_5}{\vartheta_2}v\omega - \tfrac{\vartheta_6}{\vartheta_2}\omega\right) a \cos\psi \end{aligned} \tag{59}$$

and considering that

$$\begin{pmatrix} \frac{1}{\vartheta_1}\cos\psi & -\frac{1}{\vartheta_2} a \sin\psi \\ \frac{1}{\vartheta_1}\sin\psi & \frac{1}{\vartheta_2} a \cos\psi \end{pmatrix}^{-1} = \begin{pmatrix} \vartheta_1 \cos\psi & \vartheta_1 \sin\psi \\ -\frac{\vartheta_2}{a}\sin\psi & \frac{\vartheta_2}{a}\cos\psi \end{pmatrix} \tag{60}$$

The complete control law can be expressed as

$$\begin{aligned} \begin{pmatrix} u_{vN} \\ u_{\omega N} \end{pmatrix} &= \begin{pmatrix} \vartheta_1 \cos\psi & \vartheta_1 \sin\psi \\ -\frac{\vartheta_2}{a}\sin\psi & \frac{\vartheta_2}{a}\cos\psi \end{pmatrix} \\ &\times \begin{pmatrix} -\left[\left(\frac{\vartheta_3}{\vartheta_1}\omega^2 - \frac{\vartheta_4}{\vartheta_1}v\right)\cos\psi - \left(-\frac{\vartheta_5}{\vartheta_2}v\omega - \frac{\vartheta_6}{\vartheta_2}\omega\right) a \sin\psi\right] - k_x e_x - (v_{xN} + \Gamma_x\,\mathrm{sign}(S_x)) \\ -\left[\left(\frac{\vartheta_3}{\vartheta_1}\omega^2 - \frac{\vartheta_4}{\vartheta_1}v\right)\sin\psi + \left(-\frac{\vartheta_5}{\vartheta_2}v\omega - \frac{\vartheta_6}{\vartheta_2}\omega\right) a \cos\psi\right] - k_y e_y - (v_{yN} + \Gamma_y\,\mathrm{sign}(S_y)) \end{pmatrix} \end{aligned} \tag{61}$$

7. Experimental results

To show the performance of the proposed controller, several experiments and simulations were executed and some of the results are presented in this section. The proposed controller was implemented on a Pioneer 2DX mobile robot with 4 kg of load (the parameters are indicated in Table 1 of Appendix A), which admits linear and angular velocities as input reference signals. The hardware on Pioneer2DX includes an 800 MHz Pentium III with 512 Mb RAM on-board computer in which the controller was programmed. In order to sense the robot position, odometric sensors were used.

In the experiment, the advantage of using the proposed control technique with respect to other methods in the literature will be proven.

The experiment was carried out using three different controllers on the mobile robot Pioneer 2DX with 4 kg of load. The first one uses the feedback linearisation controller (FLC), which does not use any on-line calibration, as shown in (12). The second experiment used an adaptive dynamic controller, similar to Martins et al. (2008). In this case, only the uncertainties in the robot dynamics are compensated. Finally, in the third experiment, the method proposed in this paper is applied, which consists of using a feedback controller and an adaptable compensation network (the ANSMC). In this last case, the uncertainties of the entire structure are compensated (the dynamics as well as the kinematics of the robot). For this controller, the setup parameters are: $k_x = k_y = 4$, $\Gamma_x = \Gamma_y = 0.001$ and the NN has five RBFs.

The reference trajectory to test the three controllers is in the shape of a figure eight. Fig. 4 shows the trajectories followed by the robot using each of the controllers.

In Fig. 5, the square norm of the control errors (the error norm is defined by $||\boldsymbol{e}|| = \sqrt{e_x^2 + e_y^2}$) of the three controllers is shown.

The highest error was obtained by the feedback linearisation controller, which has no on-line adaptation. In this case, the effect of the uncertainties on the error can be observed clearly. The method (Martins et al., 2008) that compensates the dynamics of the structure has a lower error than the previous case. Finally, the lowest error was obtained by the compensation method proposed in this work, which decreases the error caused by the unmodelled structure (dynamics as well as kinematics). In Fig. 6, the control actions of the ANSMC are shown.

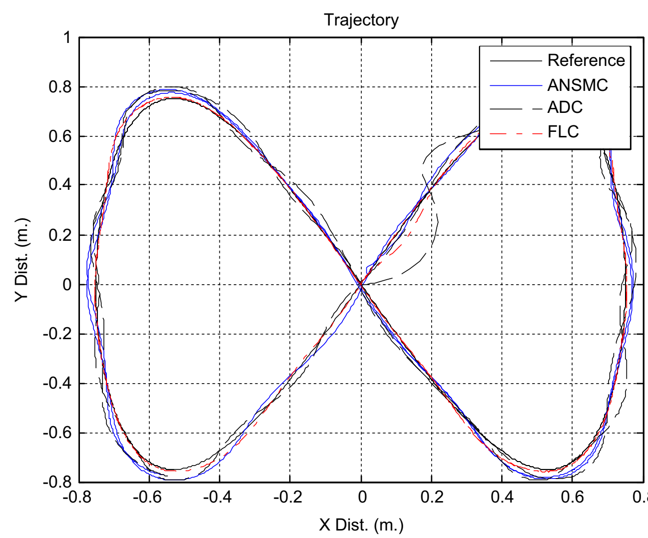

Fig. 4. Reference (dotted line) and actual trajectory of ANSMC (solid line) with RAC (dash dot line) and FLC (dashed line).

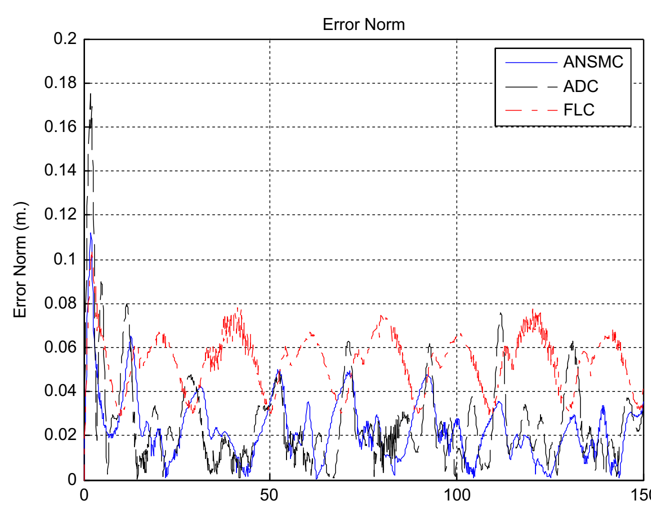

Fig. 5. Trajectory tracking error norm with ADSMC (solid line) with RAC (dash dot line) and FLC (dashed line).

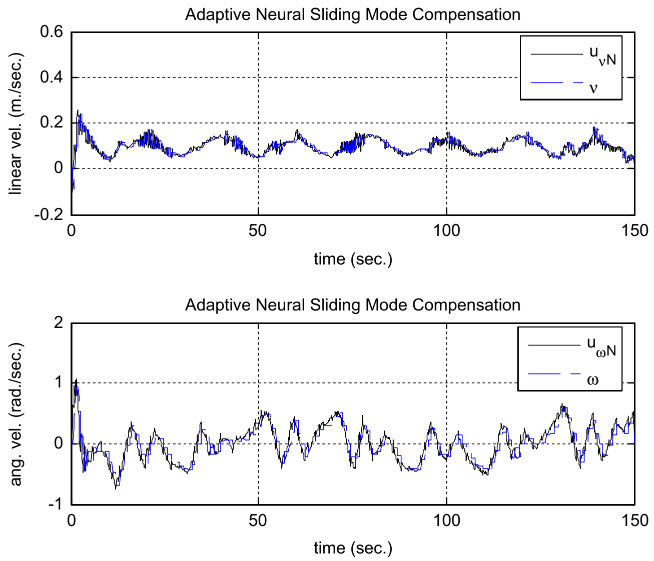

Fig. 6. Angular and linear output velocities and control actions for ANSMC technique.

Table 1
Mobile robot parameters.

Parameters	Pioneer 3DX	Pioneer 2DX	Pioneer 2DX with load (4 Kg)	Units
ϑ_1	0.24089	0.3037	0.1992	s
ϑ_2	0.2424	0.2768	0.13736	s
ϑ_3	$-9.3603e^{-4}$	$-4.018e^{-4}$	$-1.954e^{-3}$	s m/rad^2
ϑ_4	0.99629	0.9835	0.9907	
ϑ_5	$-3.725e^{-3}$	$-3.818e^{-3}$	$-1.554e^{-2}$	s/m
ϑ_6	1.0915	1.0725	0.9866	

8. Conclusions

In this paper, the problem of ANSMC for nonlinear systems with unmodelled uncertainties was considered. A feedback linearisation controller together with an adaptive NN with sliding surface was proposed. This control technique ensures an asymptotic convergence of the errors to zero.

The ANSMC reduces the control error caused by the uncertainty in the model that affects the feedback linearisation based on the nominal model. The ANSMC modifies the control action to decrease the effects of the model uncertainties and all possible disturbances that may arise. The NN in this control technique does not need to learn the entire dynamics of the system structure and it is designed to compensate for uncertainties in the model. All remaining errors are driven to zero by means of the sliding compensator. Experiments on a mobile robot have been developed to show the performance of the proposed technique, including a comparison with other controllers.

Appendix A

Variable	Description
v	Linear velocity of the mobile robot
ω	Angular velocity of the mobile robot
r_x, r_y	Cartesian coordinates of the robot (point **y**) in the XY plane
x	Velocity vector of the mobile robot
y	Point of interest with coordinate r_x, r_y in the XY plane
G	Centre of mass of the mobile robot
C	Position of the castor wheel
$\boldsymbol{\vartheta}$	Parameters vector of the mobile robot
$\boldsymbol{\vartheta}_i$	Elements of the parameters vector, where $i=1,\ldots,6$
α	Orientation of the mobile robot
$\boldsymbol{\delta}$	Uncertainties vector of the robot model.
a	Distance between the point of interest and the central point of the virtual axis of the traction wheels.
$h(\mathbf{x})$	Vector of smooth scalar fields on $\mathbb{R}^{2x1}$(kinematic model of the mobile robot)
$\tilde{h}(\mathbf{x})$	Vector of disturbances and unmodelled kinematics
$f(\mathrm{x})$	Smooth vector field on $\mathbb{R}^{2x1}$(dynamic model of mobile robot)
$\tilde{f}(\mathbf{x})$	Vector of disturbances and unmodelled dynamics
c^*	Optimal centres
η^*	Optimal widths
w	Output weights vector of the RBF neural network
$\tilde{\mathbf{w}}$	Error of weights of the output layer
$\mathbf{w}^*$	Optimal weights vector of the output layer
$\xi_i(.)$	RBF functions
$\boldsymbol{\xi}^T(.)$	Vector of RBF functions
$\mathbf{v}_N$	Output vector of the RBF networks
u	Output vector of the inverse controller $(u_v,u_\omega)^T$
$L_f h_i(\mathbf{x}) L_{gj} h_i(\mathbf{x})$	Lie derivatives of the system without disturbances.
$\Delta L_{\mathbf{f}} h_i(\mathbf{x})$ $\Delta L_{\mathbf{gj}} h_i(\mathbf{x})$	Lie derivatives corresponding to disturbances and unmodelled structure.
$e_{x,y}$	Output error for r_x and r_y, respectively
e	Vector of position error
t	Time
t_S	Time required to hit **S**

Parameters description

The identified parameters can be described by

$$\vartheta_1 = \left(\frac{((R_a/k_a)(MR_t r + 2I_e) + 2rk_{DT})}{2rk_{PT}}\right)$$

$$pt\vartheta_2 = \left(\frac{((R_a/k_a)(I_e d^2 + 2R_t r(I_z + Mb^2)) + 2rdk_{DR})}{2rdk_{PR}}\right)$$

$$\vartheta_3 = \left(\frac{(R_a/k_a)MbR_t}{2k_{PT}}\right); \quad \vartheta_4 = \left(\frac{(R_a/k_a)((k_a k_b/R_a) + B_e)}{rk_{PT}} + 1\right)$$

$$\vartheta_5 = \left(\frac{(R_a/k_a)MbR_t}{dk_{PR}}\right); \quad \vartheta_6 = \left(\frac{(R_a/k_a)((k_a k_b/R_a) + B_e)d}{2rk_{PR}} + 1\right) \tag{62}$$

In these relations, M is the robot mass, r is the radius of the left and right wheels, k_b is equal to the electromotor force constant multiplied by the reduction constant, R_a is the electric resistance, k_a is the constant of torque multiplied by the reduction constant, k_{PR}, k_{PT} and k_{DT} are positive constants, I_e and B_e are the moment of inertia and the viscous friction coefficient belonging to the combination of motor, gearbox and wheel and R_t is the nominal radius of the wheel.

References

Chen, M., Ge, S.S., How, B., 2010. Robust adaptive neural network control for a class of uncertain MIMO nonlinear systems with input nonlinearities. IEEE Transactions on Neural Networks 21 (5), 796–812.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2 (4), 303–314.

Das, T., Kar, I.N., 2006. Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. IEEE Transactions on Control Systems Technology 14 (3), 501–510.

De La Cruz, C., Carelli, R., 2006. Dynamic modeling and centralized formation control of mobile robots. In: Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics, pp. 3880–3885.

Dierks, T., Jagannathan, S., 2010. Optimal control of affine nonlinear continuous-time systems. In: Proceedings of the American Control Conference, ACC, pp. 1568–1573.

Imae, J., Ando, T., Kobayashi, T., Zhai, G., 2009. A constrained LQ approach to numerical solutions for constrained nonlinear optimal control problems. In: Proceedings of the 2009 IEEE International Conference on Networking Sensing and Control, pp. 176–180.

Isidori, A., 1995. Nonlinear Control Systems, 3rd edition Springer-Verlag, Berlin Heidelberg, Germany.

Kuperman, A., Zhong, Q.C., 2011. Robust control of uncertain nonlinear systems with state delays based on an uncertainty and disturbance estimator. International Journal of Robust and Nonlinear Control 21 (1), 79–92.

Li, H.X., Tong, S., 2003. A hybrid adaptive fuzzy control for a class of nonlinear MIMO systems. IEEE Transactions on Fuzzy Systems 11 (1), 24–34.

Liu, S., Zhang, H., Yang, S.X., Yu, J., 2004. Dynamic control of a mobile robot using an adaptive neurodynamics and sliding mode strategy. In: Proceedings of the Fifth World Congress on Intelligent Control and Automation, WCICA, vol. 6, pp. 5007–5011.

Martins, F.N., Celeste, W.C., Carelli, R., Sarcinelli-Filho, M., Bastos-Filho, T.F., 2008. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. Control Engineering Practice 16 (11), 1354–1363.

Mclain, R.B., Henson, M.A., Pottmann, M., 1999. Direct adaptive control of partially known nonlinear systems. IEEE Transactions on Neural Networks. 10 (3), 714–721.

Park, J., Sandberg, I.W., 1991. Universal approximation using radial-basis-function networks. Neural Computation 3 (2), 246–257.

Rossomando, F.G., Soria, C., Carelli, R., 2011. Autonomous mobile robots navigation using RBF neural compensator. Control Engineering Practice 19 (3), 215–222.

Sang, Y.J., Xu, C.Q., Liu, B., Kong, Q.X., Huang, F., Mao, G.Y., 2012. TS fuzzy neural network algorithm application in nonlinear control. In: Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering, ICCE, pp. 165–172.

Slotine, J.J.E., Li, W., 1991. Applied Nonlinear Control. Prentice-Hall.

Tsai, C.H., Chung, H.Y., Yu, F.M., 2004. Neuro-sliding mode control with its applications to seesaw systems. IEEE Transactions on Neural Networks 15 (1), 124–134.

Zhang, H., Luo, Y., Liu, D., 2009. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. IEEE Transactions on Neural Networks 20 (9), 1490–1503.

Zhang, H., Quan, Y., 2001. Modeling, identification, and control of a class of nonlinear systems. IEEE Transactions on Fuzzy Systems 9 (2), 349–354.