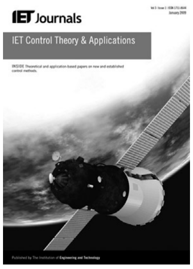
www.ietdl.org

Published in IET Control Theory and Applications
Received on 25th September 2011
Revised on 29th April 2012
doi: 10.1049/iet-cta.2011.0581



ISSN 1751-8644

Neural network-based compensation control of mobile robots with partially known structure

F.G. Rossomando C. Soria R. Carelli

Instituto de Automática (INAUT) – Universidad Nacional de San Juan, Av. San Martín 1109 Oeste,
5400 San Juan, Argentina
E-mail: frosoma@inaut.unsj.edu.ar

Abstract: This study proposes an inverse non-linear controller combined with an adaptive neural network proportional integral (PI) sliding mode using an on-line learning algorithm. The neural network acts as a compensator for a conventional inverse controller in order to improve the control performance when the system is affected by variations on their dynamics and kinematics. Also, the proposed controller can reduce the steady-state error of a non-linear inverse controller using the on-line adaptive technique based on Lyapunov's theory. Experimental results show that the proposed method is effective in controlling dynamic systems with unexpected large uncertainties.

Nomenclature

V	linear velocity of the mobile robot
ω	angular velocity of the mobile robot
r_x, r_y	Cartesian coordinates of the robot (point $\boldsymbol{y}$) in the XY plane
$\mathbf{x}$	velocity vector of the mobile robot
$\mathbf{y}$	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
ϑ	parameters vector of the mobile robot
ϑ_i	elements of the parameters vector, where $i = 1, \ldots, 6$
α	orientation of the mobile robot
δ	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 non-modelled kinematics
$f(\mathbf{x})$	smooth vector field on $\mathbb{R}^{2x1}$ (dynamic model of mobile robot)
$\tilde{f}(\mathbf{x})$	vector of disturbances and non-modelled dynamics
c^*	optimal centres
η^*	optimal widths
$\mathbf{w}$	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
$\boldsymbol{\xi}^{\mathrm{T}}(.)$	vector of RBF functions
$\mathbf{v}_N$	output vector of the RBF networks
$\mathbf{u}$	output vector of the inverse controller $(u_v, u_\omega)^{\mathrm{T}}$
$L_{\mathbf{f}}h_i(\mathbf{x})$ $L_{\mathbf{g}j}h_i(\mathbf{x})$	Lie derivatives of the system without disturbances
$\Delta L_{\mathbf{f}}h_i(\boldsymbol{x})$ $\Delta L_{\mathbf{g}j}h_i(\boldsymbol{x})$	Lie derivatives corresponding to disturbances and non-model structure
$e_{x,y}$	output error, for r_x and r_y, respectively
$\boldsymbol{e}$	vector of position error
$k_{x,y}$	error gain
t	time
t_S	the time required to hit $\boldsymbol{S}$

1 Introduction

Mobile robots are highly non-linear dynamic systems with unmodelled dynamics and uncertainties that are commonly used such as load transportation, multi-robot cooperation and accurate positioning systems. In these applications, they are expected to move quickly from one place to another or follow the desired trajectories while maintaining good dynamic performance. However, the disturbances, uncertainties and non-modelled structure in mobile robots make the design of ideal controllers for such systems a challenge task for control researchers.

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 [1–4]. To perform tasks requiring high speed movements and/or heavy load transportation, it is important to consider the robot dynamics, as well as its kinematics. No matter the uncertainties or changes in its dynamics, the tasks must be performed with precision. As an example, in the case of load transportation, the dynamics characteristics such as mass, centre of mass 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. This 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. Fukao *et al.* [5] 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 Shojaei and Shahri [6], Dong and Guo [7] and Dong and Guo [8], with the performance shown just by means of simulations. Das and Kar [9] show an adaptive fuzzy logic-based controller where the system uncertainty, which includes mobile robot parameters variation and unknown non-linearities, is estimated by a fuzzy logic system and its parameters are tuned on-line. In [10] two controllers are designed. They are based on polar coordinates, which are called position controller and heading controller. The former ensures position tracking and the latter is activated when the tracking error is small enough and the tracking reference does not change its position. This reduces the error over the mobile-robot orientation at the end of the path. Bugeja *et al.* [11] present the use of a radial base function (RBF)–NN for mobile robot dynamics approximation, in which the weights are estimated stochastically in real-time. The authors show simulation results.

A global time-varying universal controller to achieve stabilisation and tracking simultaneously in mobile robots with saturated inputs is proposed by Do and Pan [12]. 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. Oliveira *et al.* [13] present a kinematics control loop based on sliding modes technique and a dynamic control loop based on a neural network technique. Simulation and experiments results are included to demonstrate the effectiveness of the proposed control approach.

Hamerlain *et al.* [14] considers the presence of sliding effects that violate the non-holonomic constraints and the unicycle robot is modelled as a dynamic system which is referenced to the trajectory to be tracked. The proposed dynamic model becomes time varying and nonlinear which impose the use of a robust stabilising control. The efficiency of this control proposed is based on simulations. Based on the differential geometry theory in [15], the exact feedback linearisation on the kinematic error model of a mobile robot is realised. The trajectory-tracking controllers are designed by pole-assignment approach and simulation results show their effectiveness.

The work [16] presents an adaptive neural sliding mode controller for non-holonomic 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 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.

In [17] it is presented a linear parameterisation of a unicycle-like mobile robot and the design of a trajectory tracking controller based on its complete known model. One advantage of their controller is that its parameters are directly related to the robot parameters. However, if the parameters are not correctly identified or change with time due, for example, to load variation, the performance of the controller will be severely affected.

In [18] a robust neural network (NN)-based sliding mode controller (NNSMC) is used. The NNSMC algorithm can effective and coordinately control the omnidirectional mobile platform and the mounted manipulator with different dynamics. The stability of the closed-loop system, the convergence of the NN weight-updating process and the boundedness of the NN weight estimation errors are all strictly guaranteed. In this paper the NNSMC controller must learn the whole dynamic structure and the sliding surface cancels out some system uncertainties, the results are based only on simulation studies.

Kim *et al.* [19] has 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, whereas usual commercial robots accept velocity commands. In similar form, Rossomando *et al.* [20] present an approach to adaptive trajectory tracking of mobile robots which combines a feedback linearisation based on a nominal model and an RBF–NN adaptive dynamic compensation. The design of the controllers is based on discrete time and the stability analysis is based on linear parameterisation of a unicycle-like mobile robot formulated by De La Cruz and Carelli [17]. This work shows experimental results. Similar studies are shown in [21], where a hybrid control algorithm has been proposed based on a NN–RBF dynamic control and an inverse kinematics control. The algorithm was applied on a non-holonomic mobile robot and the kinematics controller designs suppose perfect velocity tracking.

In the works [11, 16, 21], it is assumed that the perturbations occur only in the robot dynamics. Most of the papers present controllers to learn the whole structure of the dynamics system, which usually require a network previously trained with on-line adjustment or an adaptive controller for this purpose. Moreover, in the works that have neural compensation, this is done only on the dynamics of the system.

In this work the main scientific contribution is the development of a neural compensator which acts on the entire system structure (dynamic and kinematics of the robot). This compensator does not need to learn the entire system structure and can be applied to a conventional control. In this case an inverse controller is designed based on a nominal model and a RBF–NN controller with the capacity to compensate the difference between the nominal model and the actual robot kinematics and dynamics. An analysis is done in order to study the effects of the RBF–NN approximation error on the control error when the whole control system. In a tracking control task, the inverse

doi: 10.1049/iet-cta.2011.0581

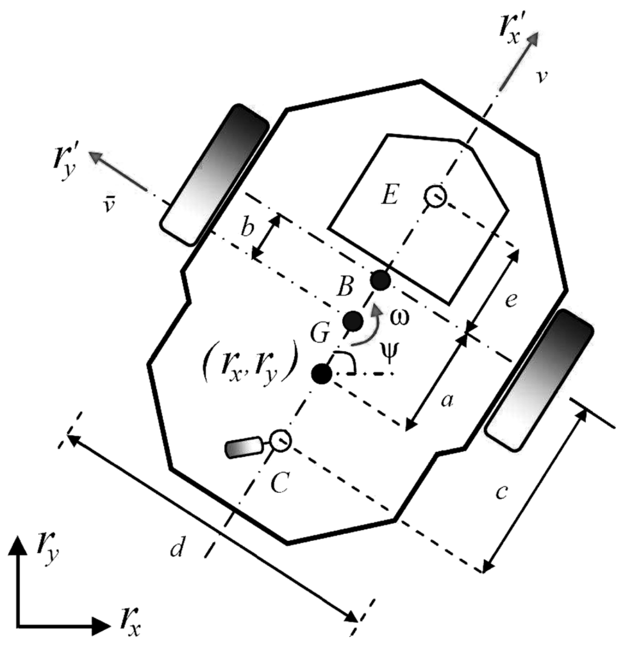


Fig. 1 *Mobile Robot parameters*

controller and the adaptive neural compensator working together, are applied.

The paper is organised as follows: Sections 2 and 3 presents a system overview and shows the mathematical representation of the complete unicycle-type robot model. The inverse and neural controllers are discussed, respectively, in Sections 4 and 5 and the corresponding error analysis is included in Section 6. Finally, Section 7 presents some experimental results to show the performance of the adaptive controller, and brief conclusions are given in Section 8.

2 Robot model

In this section, the dynamic model of the unicycle-like mobile robot presented in Fig. 1, is reviewed. This figure depicts the mobile robot, with the parameters and variables of interest. There, 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, $\boldsymbol{y}$ 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 [17], is given by the

Kinematic model

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

Dynamic model

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

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

$$\begin{aligned} \vartheta &= \begin{bmatrix} \vartheta_1 & \vartheta_2 & \vartheta_3 & \vartheta_4 & \vartheta_5 & \vartheta_6 \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{\delta} &= \begin{bmatrix} \delta_{rx} & \delta_{ry} & 0 & \delta_v & \delta_\omega \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{3}$$

respectively, where $\boldsymbol{\delta}_{rx}$ y $\boldsymbol{\delta}_{ry}$ are functions of slip velocities and robot orientation, $\boldsymbol{\delta}_v$ y $\boldsymbol{\delta}_\omega$ are functions of physical parameters as mass, inertia, wheel and tires diameters, motor and its servos parameters, forces on the wheels and others (Table 1). 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 linearisation, for both the kinematic and dynamic models of the robot.

3 Problem formulation

Considering the parameters uncertainties and non-modelled structure of systems (1) and (2), these can be expressed in compact form as

$$\begin{aligned} \dot{\mathbf{y}} &= (\mathbf{h}(\mathbf{x}) + \tilde{\mathbf{h}}(\mathbf{x})) \\ \dot{\mathbf{x}} &= (\mathbf{f}(\mathbf{x}) + \tilde{\mathbf{f}}(\mathbf{x})) + (\mathbf{g} + \tilde{\mathbf{g}})\mathbf{u} \end{aligned} \tag{4}$$

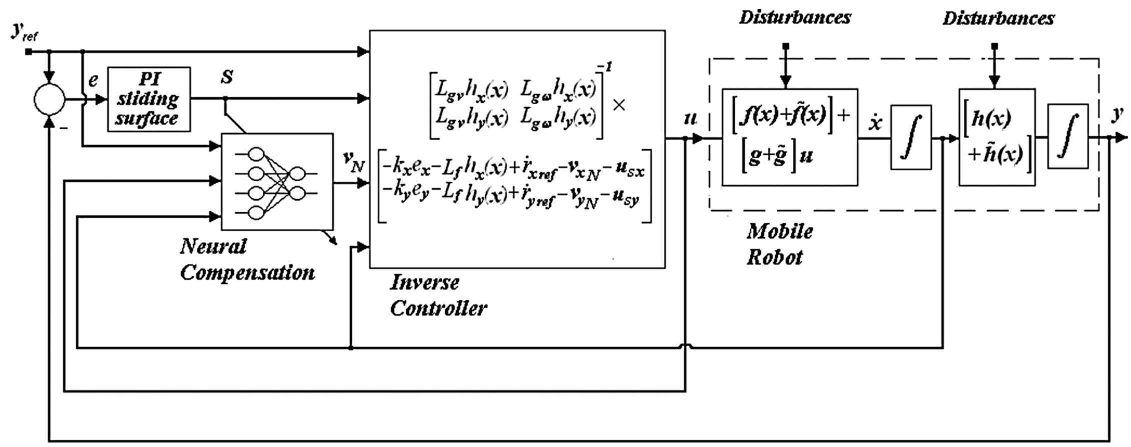
Assumption 1: Dynamic variations on (1) and (2) produce other smooth vector fields represented by $\tilde{\mathbf{f}}(\mathbf{x})$, $\tilde{\mathbf{g}}$ and $\tilde{\mathbf{h}}(\mathbf{x})$. These are composed by non-linear unknown functions. (For purposes of control ψ is not considered in this work.)

$$\dot{\mathbf{y}} = \begin{pmatrix} \dot{r}_x(t) \\ \dot{r}_y(t) \end{pmatrix}; \quad \mathbf{h}(\mathbf{x}) = \begin{pmatrix} v(t)\cos\psi(t) - \omega(t)a\sin\psi(t) \\ v(t)\sin\psi(t) + \omega(t)a\cos\psi(t) \end{pmatrix};$$

$$\tilde{\mathbf{h}}(\mathbf{x}) = \begin{pmatrix} \delta_{rx}(t) \\ \delta_{ry}(t) \end{pmatrix}$$

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	$-3.7256e^{-3}$	$-3.818e^{-3}$	$-1.554e^{-2}$	s/m
ϑ_6	1.0915	1.0725	0.9866	



Disturbances
Disturbances
y_{ref}
e
PI sliding surface
S
$-$
v_N
$\begin{bmatrix} L_{gv}h_x(x) & L_{g\omega}h_x(x) \\ L_{gv}h_y(x) & L_{g\omega}h_y(x) \end{bmatrix}^{-1} \times$
$\begin{bmatrix} -k_xe_x - L_f h_x(x) + \dot{r}_{xref} - v_{xN} - u_{sx} \\ -k_ye_y - L_f h_y(x) + \dot{r}_{yref} - v_{yN} - u_{sy} \end{bmatrix}$
Neural Compensation
Inverse Controller
u
$[f(x)+\tilde{f}(x)]+$
$[g+\tilde{g}]u$
$\dot{x}$
$\int$
$[h(x)$
$+\tilde{h}(x)]$
$\int$
y
Mobile Robot

Fig. 2 *Control structure*

and

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{pmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \dfrac{\vartheta_3}{\vartheta_1}\omega^2(t) - \dfrac{\vartheta_4}{\vartheta_1}v(t) \\ -\dfrac{\vartheta_5}{\vartheta_2}v(t)\omega(t) - \dfrac{\vartheta_6}{\vartheta_2}\omega(t) \end{pmatrix},$$

$$\mathbf{g} = \begin{pmatrix} \dfrac{1}{\vartheta_1} & 0 \\ 0 & \dfrac{1}{\vartheta_2} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u_{v\,\mathrm{ref}}(t) \\ u_{\omega\,\mathrm{ref}}(t) \end{pmatrix}, \quad \tilde{\mathbf{f}}(\mathbf{x}) + \tilde{\mathbf{g}}\mathbf{u} = \begin{pmatrix} \delta_v(t) \\ \delta_\omega(t) \end{pmatrix}$$

where: $\mathbf{x}$ is a $\mathbb{R}^{2x1}$, vector of state variables (output velocities); $\mathbf{u}$ is a $\mathbb{R}^{2x1}$ vector of manipulated input variables (reference velocities); $\mathbf{y}$ is a $\mathbb{R}^{2x1}$ vector of controlled output variables (instantaneous position); $\mathbf{f}(\mathbf{x})$ is a smooth vector field on $\mathbb{R}^{2x1}$; $\mathbf{g}$ is matrix of non-linear functions a $\mathbb{R}^2 \to \mathbb{R}^2$; and $\mathbf{h}(\mathbf{x})$ is a vector of smooth scalar fields on $\mathbb{R}^{2x1}$.

Assumption 2: The desired trajectories $r_{j\,\mathrm{ref}}$; $j = x$; y and their time derivatives up to the nth order, are continuous and bounded.

Assumption 3: The input signals u_i; $i = v$; ω, are continuous and bounded.

4 Inverse control design

For this analysis, a multiple input and multiple output (MIMO) non-linear system from (4), without uncertainties, is considered

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{h}(\mathbf{x}) \\ \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}\mathbf{u} \end{aligned} \tag{5}$$

For ease of manipulation, the case of exact input–output linearisation, in which the number of inputs is equal to the number of outputs, is considered in [22]. Then, each individual output equation, r_i, is differentiated with respect to time

$$\frac{\mathrm{d}}{\mathrm{d}t}r_i = L_{\mathrm{f}}h_i(\mathbf{x}) + \sum_{j=v}^{\omega} L_{\mathrm{g}j}h_i(\mathbf{x})u_j, \quad i = x, y \tag{6}$$

or

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{y} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} r_x \\ r_y \end{pmatrix} &= \begin{pmatrix} L_{\mathrm{f}}h_x(\mathbf{x}) \\ L_{\mathrm{f}}h_y(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \sum_{j=v}^{\omega} L_{\mathrm{g}j}h_x(\mathbf{x})u_j \\ \sum_{j=v}^{\omega} L_{\mathrm{g}j}h_y(\mathbf{x})u_j \end{pmatrix} \\ &= \begin{pmatrix} L_{\mathrm{f}}h_x(\mathbf{x}) \\ L_{\mathrm{f}}h_y(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} L_{\mathrm{g}v}h_x(\mathbf{x}) & L_{\mathrm{g}\omega}h_x(\mathbf{x}) \\ L_{\mathrm{g}v}h_y(\mathbf{x}) & L_{\mathrm{g}\omega}h_y(\mathbf{x}) \end{pmatrix}\begin{pmatrix} u_v \\ u_\omega \end{pmatrix} \end{aligned} \tag{7}$$

where $L_f h_i(\mathbf{x})$ is the ith Lie derivative of the function $\mathbf{h}(\mathbf{x})$ with respect to the vectorial field $\mathbf{f}(\mathbf{x})$. Owing to relative degree of the all outputs is one, then for each output r_i the terms $L_{\mathrm{g},j}h_i(\mathbf{x})$ will be non-zero and the input u_j appears in the derivative of the output $\mathbf{y}$. Defining the output tracking error as

$$\mathbf{e} = \mathbf{y} - \mathbf{y}_{\mathrm{r}} = \begin{pmatrix} r_x \\ r_y \end{pmatrix} - \begin{pmatrix} r_{x\,\mathrm{ref}} \\ r_{y\,\mathrm{ref}} \end{pmatrix} = \begin{pmatrix} e_x \\ e_y \end{pmatrix} \tag{8}$$

5 Design of adaptive neural sliding mode compensation

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 vector $\boldsymbol{y}_{\mathrm{r}}$.

The neural feedback linearisation method which is based on the NN-RBF model can solve this kind of control problem [22]. The state tracking error is defined as $\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{y}_{\mathrm{r}}(t)$ and the control objective is to find a control law such that the state $\boldsymbol{x}$ of the closed-loop system will follow the desired state $\mathbf{y}_{\mathrm{r}}$, in other words, the tracking error should converge to zero.

A sliding surface for MIMO system can be defined in the error state $\boldsymbol{S}(t)$, from (8).

$$\begin{aligned} \mathbf{S}(t) &= \begin{pmatrix} \left(\dfrac{\mathrm{d}}{\mathrm{d}t} + k_x\right) & 0 \\ 0 & \left(\dfrac{\mathrm{d}}{\mathrm{d}t} + k_y\right) \end{pmatrix} \int_0^t \mathbf{e}(\tau)\mathrm{d}\tau \\ &= \begin{pmatrix} e_x(t) + k_x \displaystyle\int_0^t e_x(\tau)\,\mathrm{d}\tau \\ e_y(t) + k_y \displaystyle\int_0^t e_y(\tau)\,\mathrm{d}\tau \end{pmatrix} \end{aligned} \tag{9}$$

doi: 10.1049/iet-cta.2011.0581

The derivate of sliding surface $\boldsymbol{S}(\boldsymbol{t})$ is

$$\dot{\mathbf{S}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} e_x(t) + k_x\displaystyle\int_0^t e_x(\tau)\,\mathrm{d}\tau \\ e_y(t) + k_y\displaystyle\int_0^t e_y(\tau)\,\mathrm{d}\tau \end{pmatrix} = \begin{pmatrix} \dot{e}_x + k_x e_x \\ \dot{e}_y + k_y e_y \end{pmatrix}$$
$$= \begin{pmatrix} \dfrac{\mathrm{d}}{\mathrm{d}t}(r_x - r_{x\,\mathrm{ref}}) + k_x e_x \\ \dfrac{\mathrm{d}}{\mathrm{d}t}(r_y - r_{y\,\mathrm{ref}}) + k_y e_y \end{pmatrix} \tag{10}$$

In order to make the system state tracks remain on the sliding surface, let $\dot{\boldsymbol{S}}$, (see (11))

Rearranging (16)

$$\begin{pmatrix} L_{\mathrm{g}v}h_x(\mathbf{x}) & L_{\mathrm{g}\omega}h_x(\mathbf{x}) \\ L_{\mathrm{g}v}h_y(\mathbf{x}) & L_{\mathrm{g}\omega}h_y(\mathbf{x}) \end{pmatrix}\begin{pmatrix} u_v \\ u_\omega \end{pmatrix}$$
$$= \begin{pmatrix} -L_{\mathrm{f}}h_x(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{x\,\mathrm{ref}} - k_x e_x \\ -L_{\mathrm{f}}h_y(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{y\,\mathrm{ref}} - k_y e_y \end{pmatrix} \tag{12}$$

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

$$\begin{pmatrix} u_v \\ u_\omega \end{pmatrix} = \begin{pmatrix} L_{\mathrm{g}v}h_x(\mathbf{x}) & L_{\mathrm{g}\omega}h_x(\mathbf{x}) \\ L_{\mathrm{g}v}h_y(\mathbf{x}) & L_{\mathrm{g}\omega}h_y(\mathbf{x}) \end{pmatrix}^{-1}$$
$$\times \begin{pmatrix} -L_{\mathrm{f}}h_x(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{x\,\mathrm{ref}} - k_x e_x \\ -L_{\mathrm{f}}h_y(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{y\,\mathrm{ref}} - k_y e_y \end{pmatrix} \tag{13}$$

Now, the problem of controlling the uncertain non-linear system (7), defining a control law $\mathbf{u}^*$ that guarantees the sliding condition treated in [23], is composed by an equivalent control (13) and a discontinuous term $\mathbf{u}_{\mathrm{S}} = -\Gamma\ \mathrm{sign}(\mathbf{S})$ defined by (see (14))

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

$$\mathrm{sign}(S_{\mathrm{i}}) = \begin{cases} 1, & \text{for } S_{\mathrm{i}} > 0 \\ 0, & \text{for } S_{\mathrm{i}} = 0 \\ -1, & \text{for } S_{\mathrm{i}} < 0 \end{cases} \tag{15}$$

Considering the Lyapunov function candidate defined as

$$V = \sum_{i=x,y}\frac{1}{2}(S_i^2) \tag{16}$$

Differentiating (16) with respect to time $\dot{V}$ along the system trajectory as

$$\dot{V} = \sum_{i=x,y} S_{\mathrm{i}}\dot{S}_{\mathrm{i}} = \sum_{i=x,y} S_{\mathrm{i}}\left(\frac{\mathrm{d}}{\mathrm{d}t}e_{\mathrm{i}}(t) + k_{\mathrm{i}}e_{\mathrm{i}}(t)\right)$$
$$= \sum_{i=x,y} S_{\mathrm{i}}\left(\frac{\mathrm{d}}{\mathrm{d}t}(r_{\mathrm{i}y} - r_{\mathrm{i\,ref}}) + k_{\mathrm{i}}e_{\mathrm{i}}(t)\right)$$
$$= \sum_{i=x,y} S_{\mathrm{i}}\left(L_{\mathrm{f}}h_{\mathrm{i}}(\mathbf{x}) + \sum_{j=v}^{\omega} L_{\mathrm{g}j}h_{\mathrm{i}}(\mathbf{x})u_j - \frac{\mathrm{d}}{\mathrm{d}t}r_{\mathrm{i\,ref}} + k_{\mathrm{i}}e_{\mathrm{i}}(t)\right) \tag{17}$$

replacing (14) in (17).

$$\dot{V} = \sum_{i=x,y} S_{\mathrm{i}}\dot{S}_{\mathrm{i}} = \sum_{i=x,y} S_{\mathrm{i}}(-\Gamma_{\mathrm{i}}\ \mathrm{sgn}(S_{\mathrm{i}})) \le -\sum_{i=x,y}\Gamma_{\mathrm{i}}|S_{\mathrm{i}}| \tag{18}$$

Then, dividing every term in (18) by $|S_{\mathrm{i}}|$ and integrating both sides over the interval $0 \le t \le t_S$, where t_S is the time required to hit $\boldsymbol{S}$, we obtain

$$\int_0^{tS}\left(\frac{S_{\mathrm{i}}}{|S_{\mathrm{i}}|}\dot{S}_{\mathrm{i}}\right)\mathrm{d}t \le -\int_0^{tS}\Gamma_{\mathrm{i}}\,\mathrm{d}t \Rightarrow |S_{\mathrm{i}}(t_S)| - |S_{\mathrm{i}}(0)| \le -\Gamma_{\mathrm{i}}t_S \tag{19}$$

In this way, noting that $S_{\mathrm{i}}\ (t_S) = 0$, one has

$$t_S \le \frac{|S_{\mathrm{i}}(0)|}{\Gamma_{\mathrm{i}}} \tag{20}$$

and consequently, the finite time convergence to sliding surface $\boldsymbol{S}$.

Now, considering the model indicated in (4) with parameter uncertainties, the non-modelled structure and external disturbances can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} r_x \\ r_y \end{pmatrix} = \begin{pmatrix} L_{\mathrm{f}}h_x(\mathbf{x}) + \Delta L_{\mathrm{f}}h_x(\mathbf{x}) \\ L_{\mathrm{f}}h_y(\mathbf{x}) + \Delta L_{\mathrm{f}}h_y(\mathbf{x}) \end{pmatrix}$$
$$+ \begin{pmatrix} \displaystyle\sum_{j=v}^{\omega} L_{\mathrm{g}j}h_x(\mathbf{x})u_j + \sum_{j=v}^{\omega}\Delta L_{\mathrm{g}j}h_x(\mathbf{x})u_j \\ \displaystyle\sum_{j=v}^{\omega} L_{\mathrm{g}j}h_y(\mathbf{x})u_j + \sum_{j=v}^{\omega}\Delta L_{\mathrm{g}j}h_y(\mathbf{x})u_j \end{pmatrix} \tag{21}$$

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

$$\Delta L_{\mathrm{f}}h_i(\mathbf{x}) = \frac{\partial\tilde{h}_i(\mathbf{x})}{\partial\mathbf{x}}f(\mathbf{x}) + \frac{\partial h_i(\mathbf{x})}{\partial\mathbf{x}}\tilde{f}(\mathbf{x}) + \frac{\partial\tilde{h}_i(\mathbf{x})}{\partial\mathbf{x}}\tilde{f}(\mathbf{x})$$
$$\Delta L_{\mathrm{g}j}h_i(\mathbf{x}) = \frac{\partial\tilde{h}_i(\mathbf{x})}{\partial\mathbf{x}}g_j + \frac{\partial h_i(\mathbf{x})}{\partial\mathbf{x}}\tilde{g}_j + \frac{\partial\tilde{h}_i(\mathbf{x})}{\partial\mathbf{x}}\tilde{g}_j \tag{22}$$

$$\dot{\mathbf{S}}(t) = \begin{pmatrix} \left(L_{\mathrm{f}}h_x(\mathbf{x}) + L_{\mathrm{g}v}h_x(\mathbf{x})u_v + L_{\mathrm{g}\omega}h_x(\mathbf{x})u_\omega - \dfrac{\mathrm{d}}{\mathrm{d}t}r_{x\,\mathrm{ref}}\right) + k_x e_x \\ \left(L_{\mathrm{f}}h_y(\mathbf{x}) + L_{\mathrm{g}v}h_y(\mathbf{x})u_v + L_{\mathrm{g}\omega}h_y(\mathbf{x})u_\omega - \dfrac{\mathrm{d}}{\mathrm{d}t}r_{y\,\mathrm{ref}}\right) + k_y e_y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{11}$$

$$\begin{pmatrix} u_v \\ u_\omega \end{pmatrix} = \begin{pmatrix} L_{\mathrm{g}v}h_x(\mathbf{x}) & L_{\mathrm{g}\omega}h_x(\mathbf{x}) \\ L_{\mathrm{g}v}h_y(\mathbf{x}) & L_{\mathrm{g}\omega}h_y(\mathbf{x}) \end{pmatrix}^{-1} \times \left[\begin{pmatrix} -L_{\mathrm{f}}h_x(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{x\,\mathrm{ref}} - k_x e_x \\ -L_{\mathrm{f}}h_y(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{y\,\mathrm{ref}} - k_y e_y \end{pmatrix} - \begin{pmatrix} \Gamma_x & 0 \\ 0 & \Gamma_y \end{pmatrix}\begin{pmatrix} \mathrm{sign}\,(S_x) \\ \mathrm{sign}\,(S_y) \end{pmatrix}\right] \tag{14}$$

doi: 10.1049/iet-cta.2011.0581

Fig. 3 *Mobile robot Pioneer 2DX*

Replacing the control action proposed in (14) in (21). (see (23))

The sliding mode control is generally used to improve the closed-loop robustness when the system is subjected to bounded external disturbance and residual modelling uncertainties. Since mobile robots are electromechanical devices, chattering at the control action is undesiderable because it could excite mechanical resonance Fig. 3. To overcome this problem, it must be introduced an adaptive term $\mathbf{v}_{\mathbf{iN}}(t)$ in the control law in order to compensate the uncertainties and disturbances of the system, reducing the chattering effect. From (22), the adaptive sliding control compensation is indicated as (see (24))

where v_{iN} is a compensation variable and can be approximated by radial basis function–neural network (RBF–NN), the parameters are tuned on-line

$$v_{iN}^{*} = \mathbf{w}_{ij}^{*\mathrm{T}}\boldsymbol{\xi}_j^{*}(\mathbf{x}, \mathbf{c}^{*}, \boldsymbol{\eta}^{*}) + \sum_{j=v}^{\omega} \boldsymbol{\varphi}_{ij}^{*}\boldsymbol{\xi}_j^{*}(\mathbf{x}, \mathbf{c}_j^{*}, \boldsymbol{\eta}_j^{*})u_j + \varepsilon_{\mathrm{in}} \quad i = x, y \tag{25}$$

where $\mathbf{w}^{*}(5 \times 2)$, $\boldsymbol{\varphi}^{*}(5 \times 2)$ and $\boldsymbol{\xi}^{*}(5 \times 1)$ are optimal parameter vectors of weights $\mathbf{w}$, input weights vector $\boldsymbol{\varphi}$ and radial basis functions $\boldsymbol{\xi}$, respectively; $\mathbf{c}^{*}$ and η^{*} are optimal parameter vectors of centres $\mathbf{c}$ and widths η, respectively; and $\boldsymbol{\varepsilon}_n$ is the approximation error.

Assumption 4: , $\Delta L_{\mathbf{f}}h_i(\mathbf{x})$ and $\Delta L_{\mathrm{g}_j}h_i(\mathbf{x})$ functions can be approximated by the output of a RBF-NN [24] with the approximation error bounded by

$$\left| \Delta L_{\mathrm{f}}h_i(\mathbf{x}) + \sum_{j=v}^{\omega} \Delta L_{\mathrm{g}j}h_i(\mathbf{x})u_j - \mathbf{w}_i^{*\mathrm{T}}\boldsymbol{\xi}^{*}(\mathbf{x}, \mathbf{c}^{*}, \boldsymbol{\eta}^{*}) + \sum_{j=v}^{\omega} \boldsymbol{\varphi}_{ij}^{*}\boldsymbol{\xi}_j^{*}(\mathbf{x}, \mathbf{c}_j^{*}, \boldsymbol{\eta}_j^{*})u_j \right| \leq \varepsilon_{\mathrm{in}} \quad \forall \mathbf{x} \in \mathbb{R}^2 \tag{26}$$

where $\mathbf{x}$ is the input vector to the RBF-NN, $\varepsilon_{\mathrm{Max}} \geq \varepsilon_{\mathrm{in}} > 0$ 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 $\xi(\mathbf{x}) = [\xi_1(\mathbf{x}), \xi_2(\mathbf{x}), \ldots, \xi_l(\mathbf{x})]^{\mathrm{T}}$ is defined by

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

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

Assumption 5: Function approximation weights $\mathbf{w}$ and φ are bounded.

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

The optimal parameters of (28) are unknown, so it is necessary to estimate the values. Defining an estimated function

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

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

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

$$\begin{aligned} \hat{v}_{iN} = {} & \hat{\mathbf{w}}_i^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \tilde{\mathbf{w}}_i^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\eta}}) + \hat{\mathbf{w}}_i^{\mathrm{T}}\tilde{\boldsymbol{\xi}}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) \\ & + \tilde{\mathbf{w}}_i^{\mathrm{T}}\tilde{\boldsymbol{\xi}}(\mathbf{x}, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\eta}}) + \cdots + \sum_{j=v,\omega} [\hat{\boldsymbol{\varphi}}_{ij}\hat{\boldsymbol{\xi}}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j)u_j \\ & + \tilde{\boldsymbol{\varphi}}_{ij}\hat{\boldsymbol{\xi}}_j(\mathbf{x}, \hat{\mathbf{c}}_j, \hat{\eta}_j)u_j + \hat{\boldsymbol{\varphi}}_{ij}\tilde{\boldsymbol{\xi}}_j(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j)u_j \\ & + \tilde{\boldsymbol{\varphi}}_{ij}\tilde{\boldsymbol{\xi}}_j(\mathbf{x}, \tilde{\mathbf{c}}_j, \tilde{\boldsymbol{\eta}}_j)u_j] + \varepsilon_{\mathrm{in}} \end{aligned} \tag{30}$$

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{S}(t) = \begin{pmatrix} -k_x e_x + \Delta L_{\mathrm{f}}h_x(\mathbf{x}) + \sum\limits_{j=v}^{\omega} \Delta L_{\mathrm{g}j}h_x(\mathbf{x})u_j - \Gamma_x \ \mathrm{sign}(S_x) \\ -k_y e_y + \Delta L_{\mathrm{f}}h_y(\mathbf{x}) + \sum\limits_{j=v}^{\omega} \Delta L_{\mathrm{g}j}h_y\mathbf{x}u_j - \Gamma_y \ \mathrm{sign}(S_y) \end{pmatrix} \tag{23}$$

$$\begin{pmatrix} u_{vN} \\ u_{\omega N} \end{pmatrix} = \begin{pmatrix} L_{\mathrm{g}v}h_x(\mathbf{x}) & L_{\mathrm{g}\omega}h_x(\mathbf{x}) \\ L_{\mathrm{g}v}h_y(\mathbf{x}) & L_{\mathrm{g}\omega}h_y(\mathbf{x}) \end{pmatrix}^{-1} \times \begin{pmatrix} -L_{\mathrm{f}}h_x(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{dt}}r_{x\,\mathrm{ref}} - k_x e_x - (v_{xN} + \Gamma_x\mathrm{sign}(S_x)) \\ -L_{\mathrm{f}}h_y(\mathbf{x}) + \dfrac{\mathrm{d}}{\mathrm{d}t}r_{y\,\mathrm{ref}} - k_y e_y - (v_{yN} + \Gamma_y\mathrm{sign}(S_y)) \end{pmatrix} \tag{24}$$

doi: 10.1049/iet-cta.2011.0581

Doing $\hat{\mathbf{w}}_i\hat{\boldsymbol{\xi}}(\mathbf{x},\hat{\mathbf{c}},\hat{\eta}) = \Delta L_{\mathrm{f}}h_i(\mathbf{x})$ and $\sum_{j=v,\omega}\hat{\varphi}_{ij}\hat{\boldsymbol{\xi}}_j(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\boldsymbol{\eta}}_j)u_j = \sum_{j=v,\omega}\Delta L_{\mathrm{g}j}h_i(\mathbf{x})u_j$.

Combining control law (24) and neural compensation (30) into the robotic model (18) the close loop error equation yields (see (31))

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

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

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

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

Equation (32) can be expressed as

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

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

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

where κ_1, κ_2 and κ_3 are some constants because of the fact that RBF and its derivative are always bounded by constants (the proof is omitted here to save space). Substituting (34) into (31), it can be obtained (see (36))

where the uncertain $\hat{\mathbf{w}}_i^T\mathbf{O} + \sum_{j=v}^{\omega}(\hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}}\mathbf{O}_ju_j) + \varepsilon_{\mathrm{in}}$, is assumed to be bounded by

$$|\varepsilon_{\mathrm{Max}}| = \left|\hat{\mathbf{w}}_i^{\mathrm{T}}\mathbf{O} + \sum_{j=v,\omega}(\hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}}\mathbf{O}_ju_j) + \varepsilon_{\mathrm{in}}\right| \le \Gamma_i, \quad i = x,y \quad (37)$$

6 Stability analysis and neural parameters adjustment

To derive the stable tuning law, the following Lyapunov function is chosen

$$\begin{aligned} V = \frac{1}{2}\sum_{i=x}^{y}\left[p_iS_i^2 + \tilde{\mathbf{w}}_i^{\mathrm{T}}\theta_i\tilde{\mathbf{w}}_i + \sum_{j=v}^{\omega}\tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}}\rho_{ij}\tilde{\varphi}_{ij}\right] \\ + \frac{1}{2}(\tilde{\mathbf{c}}^{\mathrm{T}}\Lambda_1\tilde{\mathbf{c}} + \tilde{\boldsymbol{\eta}}^{\mathrm{T}}\Lambda_2\tilde{\boldsymbol{\eta}}) \end{aligned} \quad (38)$$

where $\mathbf{P}$ is a 2×2 diagonal positive definite matrix, and θ_i and $\Lambda_{1,2}$ are 5×5 and 2×2 non-negative definite matrices, respectively. The derivative of the Lyapunov function is given by

$$\begin{aligned} \frac{\mathrm{d}V}{\mathrm{d}t} = \sum_i\left[p_iS_i\frac{\mathrm{d}S_i}{\mathrm{d}t} + \tilde{\mathbf{w}}_i^{\mathrm{T}}\theta_i\frac{\mathrm{d}\tilde{\mathbf{w}}_i}{\mathrm{d}t} + \sum_{j=v}^{\omega}\tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}}\rho_{ij}\frac{\mathrm{d}\tilde{\boldsymbol{\varphi}}_{ij}}{\mathrm{d}t}\right] \\ + \left(\frac{\mathrm{d}\tilde{\mathbf{c}}^{\mathrm{T}}}{\mathrm{d}t}\Lambda_1\tilde{\mathbf{c}} + \frac{\mathrm{d}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}}{\mathrm{d}t}\Lambda_2\tilde{\boldsymbol{\eta}}\right) \end{aligned} \quad (39)$$

Substituting (36) in (39), and considering that $\mathbf{K}^{\mathrm{T}}\mathbf{P} = (\mathbf{K}^{\mathrm{T}}\mathbf{P})^{\mathrm{T}}$, $\mathbf{P}$ and $\mathbf{K}$ are a diagonal matrix. And doing $\mathbf{Q} = \mathbf{K}^{\mathrm{T}}\mathbf{P}$, (39) can be written as (see (40))

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} S_x \\ S_y \end{pmatrix} = \begin{pmatrix} -k_xe_x - \tilde{\mathbf{w}}_x^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x},\hat{\mathbf{c}},\hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_x^{\mathrm{T}}\tilde{\boldsymbol{\xi}}(\mathbf{x},\tilde{\mathbf{c}},\tilde{\boldsymbol{\eta}}) - \sum_{j=v}^{\omega}[\tilde{\boldsymbol{\varphi}}_{xj}\hat{\boldsymbol{\xi}}_j(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\eta}_j)u_j + \hat{\boldsymbol{\varphi}}_{xj}\tilde{\boldsymbol{\xi}}(\mathbf{x},\tilde{\mathbf{c}}_j,\tilde{\boldsymbol{\eta}}_j)u_j] - \varepsilon_{xn} \\ -k_ye_y - \tilde{\mathbf{w}}_y^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x},\hat{\mathbf{c}},\hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_y^{\mathrm{T}}\tilde{\boldsymbol{\xi}}(\mathbf{x},\tilde{\mathbf{c}},\tilde{\boldsymbol{\eta}}) - \sum_{j=v}^{\omega}[\tilde{\boldsymbol{\varphi}}_{yj}\hat{\boldsymbol{\xi}}_j(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\eta}_j)u_j + \hat{\boldsymbol{\varphi}}_{yj}\tilde{\boldsymbol{\xi}}(\mathbf{x},\tilde{\mathbf{c}}_j,\tilde{\boldsymbol{\eta}}_j)u_j] - \varepsilon_{yn} \end{pmatrix} - \begin{pmatrix} \Gamma_x\mathrm{sign}(S_x) \\ \Gamma_y\mathrm{sign}(S_y) \end{pmatrix} \quad (31)$$

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} S_x \\ S_y \end{pmatrix} = &\begin{pmatrix} -k_xe_x - \tilde{\mathbf{w}}_x^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x},\hat{\mathbf{c}},\hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_x^{\mathrm{T}}(\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}} + \boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}} + \mathbf{O}) - \cdots - \sum_{j=v,\omega}[\tilde{\boldsymbol{\varphi}}_{xj}\hat{\boldsymbol{\xi}}_j(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\eta}_j)u_j + \hat{\boldsymbol{\varphi}}_{xj}(\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j)u_j] - \varepsilon_{xn} \\ -k_ye_y - \tilde{\mathbf{w}}_y^{\mathrm{T}}\hat{\boldsymbol{\xi}}(\mathbf{x},\hat{\mathbf{c}},\hat{\boldsymbol{\eta}}) - \hat{\mathbf{w}}_y^{\mathrm{T}}(\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}} + \boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}} + \mathbf{O}) - \cdots - \sum_{j=v,\omega}[\tilde{\boldsymbol{\varphi}}_{yj}\hat{\boldsymbol{\xi}}_j(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\eta}_j)u_j + \hat{\boldsymbol{\varphi}}_{yj}(\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j)u_j] - \varepsilon_{yn} \end{pmatrix} \\ &- \cdots - \begin{pmatrix} \Gamma_x & 0 \\ 0 & \Gamma_y \end{pmatrix}\begin{pmatrix} \mathrm{sign}(S_x) \\ \mathrm{sign}(S_y) \end{pmatrix} \end{aligned} \quad (36)$$

$$\begin{aligned} \frac{\mathrm{d}V}{\mathrm{d}t} = \sum_i\Bigg[&-q_iS_i^2 - p_iS_i\tilde{\mathbf{w}}_i^{\mathrm{T}}\hat{\boldsymbol{\xi}} - p_iS_i\hat{\mathbf{w}}_i^{\mathrm{T}}\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}} - p_iS_i\hat{\mathbf{w}}_i^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}} - p_iS_i\hat{\mathbf{w}}_i^{\mathrm{T}}\mathbf{O} - p_iS_i\sum_{j=v,\omega}\tilde{\boldsymbol{\varphi}}_{ij}\hat{\boldsymbol{\xi}}_i(\mathbf{x},\hat{\mathbf{c}}_j,\hat{\eta}_j)u_j - \cdots \\ &- p_iS_i\sum_{j=v,\omega}\hat{\boldsymbol{\varphi}}_{ij}(\boldsymbol{\Xi}^{\mathrm{T}}\tilde{\mathbf{c}}_j + \boldsymbol{\Phi}^{\mathrm{T}}\tilde{\boldsymbol{\eta}}_j + \mathbf{O}_j)u_j - p_iS_i\varepsilon_{\mathrm{in}} - p_iS_i\Gamma_i\mathrm{sign}(S_i) + \tilde{\mathbf{w}}_i^{\mathrm{T}}\theta_i\frac{\mathrm{d}\tilde{\mathbf{w}}_i}{\mathrm{d}t} + \sum_{j=v}^{\omega}\tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}}\rho_{ij}\frac{\mathrm{d}\tilde{\boldsymbol{\varphi}}_{ij}}{\mathrm{d}t}\Bigg] \\ &+ \left(\frac{\mathrm{d}\tilde{\mathbf{c}}^{\mathrm{T}}}{\mathrm{d}t}\Lambda_1\tilde{\mathbf{c}} + \frac{\mathrm{d}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}}{\mathrm{d}t}\Lambda_2\tilde{\boldsymbol{\eta}}\right) \end{aligned} \quad (40)$$

Rearranging (40)

$$\begin{aligned}\frac{\mathrm{d}V}{\mathrm{d}t} = \sum_i \Bigg[& -q_i S_i^2 - p_i S_i \tilde{\mathbf{w}}_i^{\mathrm{T}} \hat{\boldsymbol{\xi}} - p_i S_i \hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Xi}^{\mathrm{T}} \tilde{\mathbf{c}} - p_i S_i \hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{\eta}} \\ & - p_i S_i \hat{\mathbf{w}}_i^{\mathrm{T}} \mathbf{O} + \tilde{\mathbf{w}}_i^{\mathrm{T}} \theta_i \frac{\mathrm{d}\tilde{\mathbf{w}}_i}{\mathrm{d}t} + \sum_{j=v,\omega} \tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \rho_{ij} \frac{\mathrm{d}\tilde{\boldsymbol{\varphi}}_{ij}}{\mathrm{d}t} \\ & - p_i S_i \sum_{j=v,\omega} \tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \hat{\boldsymbol{\xi}}_j u_j - p_i S_i \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{ij} \boldsymbol{\Xi}^{\mathrm{T}} \tilde{\mathbf{c}}_j u_j \\ & - p_i S_i \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{ij} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{\eta}}_j u_j - p_i S_i \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{jj} \mathbf{O}_j u_j \\ & - p_i S_i \varepsilon_{\mathrm{in}} - p_i \Gamma_i |S_i| \Bigg] + \left(\frac{\mathrm{d}\tilde{\mathbf{c}}^{\mathrm{T}}}{\mathrm{d}t} \Lambda_1 \tilde{\mathbf{c}} + \frac{\mathrm{d}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}}{\mathrm{d}t} \Lambda_2 \tilde{\boldsymbol{\eta}} \right) \end{aligned} \tag{41}$$

Rearranging and grouping terms (see (42))

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

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

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

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

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

Considering (43), (44), (45) and (46) into (41), then (41) can be rewritten as

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

from (47), it follows that

$$\frac{\mathrm{d}V}{\mathrm{d}t} \le \sum_i -q_i |S_i|^2 < 0 \tag{48}$$

From (43), (44), (45) and (46) considering $\dot{\mathbf{w}}_i^* = 0, \dot{\mathbf{c}}^* = 0, \dot{\boldsymbol{\eta}}^* = 0$, and $\dot{\boldsymbol{\varphi}}^* = 0$ the tuning rules are

$$\frac{\mathrm{d}\mathbf{w}_i}{\mathrm{d}t} = \theta_i^{-1} p_i S_i \hat{\boldsymbol{\xi}} \tag{49}$$

$$\frac{\mathrm{d}\boldsymbol{\varphi}_{ij}}{\mathrm{d}t} = \rho_{ij}^{-1} p_i S_i \hat{\boldsymbol{\xi}}_j u_j \tag{50}$$

$$\frac{\mathrm{d}\boldsymbol{\eta}^{\mathrm{T}}}{\mathrm{d}t} = \Lambda_2^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} u_j \right] \tag{51}$$

$$\frac{\mathrm{d}\mathbf{c}^{\mathrm{T}}}{\mathrm{d}t} = \Lambda_1^{-1} \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Xi}^{\mathrm{T}} + \sum_j \hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \boldsymbol{\Xi}^{\mathrm{T}} u_j \right] \tag{52}$$

From (48) the negativeness of the time derivative of the Lyapunov candidate function is guaranteed, resulting in the stability of the overall system.

Now from (1) and (2) and taking into account the complete control law (24), the Lie derivatives are

$$\begin{aligned} L_{\mathrm{g}v} h_x(\mathbf{x}) &= \frac{1}{\vartheta_1} \cos \psi \\ L_{\mathrm{g}v} h_y(\mathbf{x}) &= -\frac{1}{\vartheta_2} a \sin \psi \\ L_{\mathrm{g}\omega} h_x(\mathbf{x}) &= \frac{1}{\vartheta_1} \sin \psi \\ L_{\mathrm{g}\omega} h_y(\mathbf{x}) &= \frac{1}{\vartheta_2} a \cos \psi \\ L_{\mathrm{f}} h_x(\mathbf{x}) &= \left(\frac{\vartheta_3}{\vartheta_1} \omega^2 - \frac{\vartheta_4}{\vartheta_1} v \right) \cos \psi \\ &\quad - \left(-\frac{\vartheta_5}{\vartheta_2} v\omega - \frac{\vartheta_6}{\vartheta_2} \omega \right) a \sin \psi \\ L_{\mathrm{f}} h_y(\mathbf{x}) &= \left(\frac{\vartheta_3}{\vartheta_1} \omega^2 - \frac{\vartheta_4}{\vartheta_1} v \right) \sin \psi \\ &\quad + \left(-\frac{\vartheta_5}{\vartheta_2} v\omega - \frac{\vartheta_6}{\vartheta_2} \omega \right) a \cos \psi \end{aligned} \tag{53}$$

And considering that

$$\begin{pmatrix} \frac{1}{\vartheta_1} \cos \psi & -\frac{1}{\vartheta_2} a \sin \psi \\ \frac{1}{\vartheta_1} \sin \psi & \frac{1}{\vartheta_2} a \cos \psi \end{pmatrix}^{-1} = \begin{pmatrix} \vartheta_1 \cos \psi & \vartheta_1 \sin \psi \\ -\frac{\vartheta_2}{a} \sin \psi & \frac{\vartheta_2}{a} \cos \psi \end{pmatrix} \tag{54}$$

$$\begin{aligned}\frac{\mathrm{d}V}{\mathrm{d}t} = \sum_i \Bigg[& -q_i S_i^2 + \tilde{\mathbf{w}}_i^{\mathrm{T}} \left(-p_i S_i \hat{\boldsymbol{\xi}} + \theta_i \frac{\mathrm{d}\tilde{\mathbf{w}}_i}{\mathrm{d}t} \right) + \cdots + \sum_{j=v,\omega} \tilde{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \left(-p_i S_i \hat{\boldsymbol{\xi}}_j u_j + \rho_{ij} \frac{\mathrm{d}\tilde{\boldsymbol{\varphi}}_{ij}}{\mathrm{d}t} \right) \\ & - p_i S_i \left(\hat{\mathbf{w}}_i^{\mathrm{T}} \mathbf{O} + \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{ij} \mathbf{O}_j u_j + \varepsilon_{\mathrm{in}} \right) - p_i \Gamma_i |S_i| \Bigg] + \cdots + \left(- \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} + \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} u_j \right] + \frac{\mathrm{d}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}}{\mathrm{d}t} \Lambda_2 \right) \tilde{\boldsymbol{\eta}} + \cdots \\ & + \left(- \sum_i p_i S_i \left[\hat{\mathbf{w}}_i^{\mathrm{T}} \boldsymbol{\Xi}^{\mathrm{T}} + \sum_{j=v,\omega} \hat{\boldsymbol{\varphi}}_{ij}^{\mathrm{T}} \boldsymbol{\Xi}^{\mathrm{T}} u_j \right] + \frac{\mathrm{d}\tilde{\mathbf{c}}^{\mathrm{T}}}{\mathrm{d}t} \Lambda_1 \right) \tilde{\mathbf{c}} \end{aligned} \tag{42}$$

doi: 10.1049/iet-cta.2011.0581

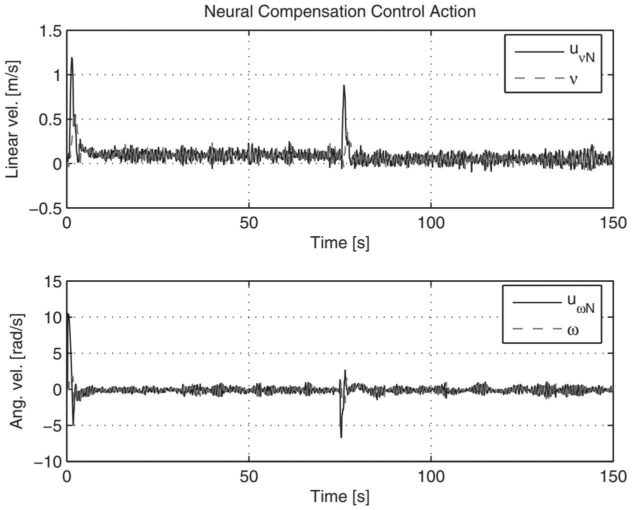


Fig. 4 *Angular and linear velocities output and control actions of sliding neural compensator control technique*

The complete control law can be expressed as (see (55))

7 Experimental results

To show the performance of the proposed controller, several experiments were executed and some of the results are presented in this section. The proposed controller was implemented on a Pioneer 2DX mobile robot, which admits linear and angular velocities as input reference signals. The Pioneer2DX has an 800 MHz Pentium III with 512 Mb RAM onboard computer in which the controller was programmed. The controller setup parameters are: $k_x = k_y = 4$, $\Gamma_x = 0.001$ and $\Gamma_y = 0.001$, the NN has 5 RBFs. In order to sense the robot position and velocities, odometric sensors were used, the sensitivity of the odometres is 1 mm for position, $1\,\mathrm{mms}^{-1}$ for linear velocity and 1°/s for angular velocity. It was verified that the wheels of the mobile robot roll without slipping.

$$\begin{cases} r_x = 0.75\sin(0.03\pi t) \\ r_y = 0.75\cos(0.03\pi t) \end{cases} \tag{56}$$

The analysis of the controller performance was accomplished in two stages Fig. 4. In the first one, the control system was analysed in an inverse dynamic scheme in the absence of sliding neural compensation. For this situation, the controller without compensation performs the trajectory tracking as shown in Fig. 5 and the tracking error in Fig. 6 (both cases are indicated by dashed line). This figure shows a behaviour with greater error of the control system in tracking the desired robot trajectory.

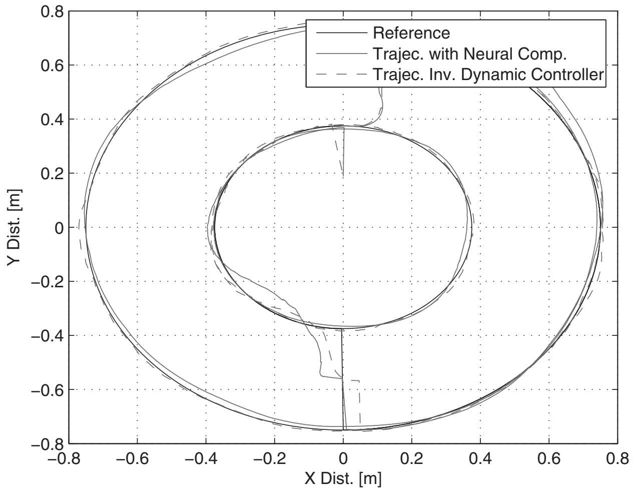


Fig. 5 *Reference and actual trajectory neural compensator (solid line) and inverse controller (dashed line)*

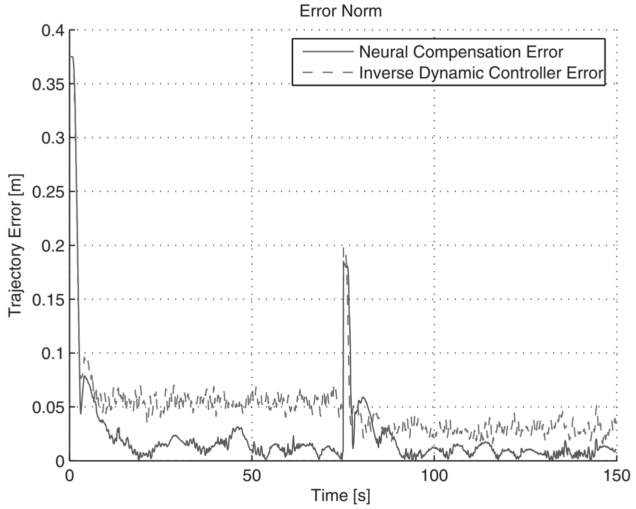


Fig. 6 *Trajectory Error with sliding neural compensator (solid line) and inverse controller (dashed line)*

In the second stage, a neural compensation was included. The control scheme with compensation has a very good performance. On the other hand, adding the neural structure allows a significant improvement and an effective control of robot trajectory, which can be verified in Figs. 5 and 6 (solid line). Fig. 6 shows the distance errors for experiments using the proposed RBF compensator to follow the desired reference trajectory. The distance error is defined as the instantaneous distance between the reference and the robot position. Note the high initial error, which is because of the

$$\begin{pmatrix} u_{vN} \\ u_{\omega N} \end{pmatrix} = \begin{pmatrix} \vartheta_1 \cos\psi & \vartheta_1 \sin\psi \\ -\dfrac{\vartheta_2}{a}\sin\psi & \dfrac{\vartheta_2}{a}\cos\psi \end{pmatrix} \times \begin{pmatrix} -\left[\left(\dfrac{\vartheta_3}{\vartheta_1}\omega^2 - \dfrac{\vartheta_4}{\vartheta_1}v\right)\cos\psi - \left(-\dfrac{\vartheta_5}{\vartheta_2}v\omega - \dfrac{\vartheta_6}{\vartheta_2}\omega\right)a\sin\psi\right] - \lambda_x e_x - (v_{xN} + \Gamma_x \mathrm{sign}(S_x)) \\ -\left[\left(\dfrac{\vartheta_3}{\vartheta_1}\omega^2 - \dfrac{\vartheta_4}{\vartheta_1}v\right)\sin\psi + \left(-\dfrac{\vartheta_5}{\vartheta_2}v\omega - \dfrac{\vartheta_6}{\vartheta_2}\omega\right)a\cos\psi\right] - \lambda_y e_y - (v_{yN} + \Gamma_y \mathrm{sign}(S_y)) \end{pmatrix} \tag{55}$$

fact that the reference trajectory starts on a point that is far from the initial robot position.

The Fig. 4 depicts the speeds and control actions of the RBF adaptive controller and outputs velocities.

8 Conclusions

In this paper, a trajectory tracking controller for a unicycle-like mobile robot, including a neural adaptive compensator, is proposed. The controller is capable of generating smooth and continuous velocity commands to the robot. The tracking control errors can asymptotically converge to zero. The RBF–NN controller compensates the difference between a known nominal dynamics structure and the actual dynamics structure of the robot. Therefore the computational effort is significantly smaller than a NN learning the complete inverse model of the robot. The proposed neural compensation scheme behaves with strong robustness with respect to unknown dynamics and non-linearities and the stability of the closed loop system and convergence towards zero of the tracking error is guaranteed. Experimental results show the good performance of the proposed tracking controller and its capacity to adapt to the actual robot dynamics.

9 References

1 Scaglia, G., Rosales, A., Quintero, L., Mut, V., Agarwal, R.: 'A linear-interpolation-based controller design for trajectory tracking of mobile robots', *Control Eng. Pract.*, 2010, **18**, (3), pp. 318–329

2 Wu, W., Chen, H., Wang, Y., Woo, P.: 'Adaptive exponential stabilization of mobile robots with uncertainties'. Proc. IEEE 38th Conf. on Decision and Control, Phoenix, Arizona, USA, 1999, pp. 3484–3489

3 Gracia, L., Tornero, J.: 'Kinematic control of wheeled mobile robots', *Latin Am. Appl. Res.*, 2008, **38**, pp. 7–16

4 Künhe, F., Gomes, J., Fetter, W.: 'Mobile robot trajectory tracking using model predictive control'. Second IEEE Latin-American Robotics Symp., São Luis, Brazil, 2005

5 Fukao, T., Nakagawa, H., Adachi, N.: 'Adaptive tracking control of a nonholonomic mobile robot', *IEEE Trans. Robot. Autom.*, 2000, **16**, (5), pp. 609–615

6 Shojaei, K., Shahri, A.M.: 'Adaptive robust time-varying control of uncertain non-holonomic robotic systems', *IET Control Theory Appl.*, 2012, **6**, (1), pp. 90–102

7 Dong, W., Guo, Y.: 'Dynamic tracking control of uncertain mobile robots'. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2005, pp. 2774–2779

8 Dong, W., Huo, W.: 'Tracking control of wheeled mobile robots with unknown dynamics'. Proc. IEEE Int. Conf. on Robotics & Automation, Detroit, Michigan, 1999, pp. 2645–2650

9 Das, T., Kar, I.N.: 'Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots', *IEEE Trans. Control Syst. Technol.*, 2006, **14**, (3), pp. 501–510

10 Chwa, D.: 'Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates', *IEEE Trans. Control Syst. Technol.*, 2004, **12**, (4), pp. 637–644

11 Bugeja, M.K., Fabri, S.G., Camilleri, L.: 'Dual adaptive dynamic control of mobile robots using neural networks', *IEEE Trans. Syst. Man, Cybern.*, 2009, **39**, (1), pp. 129–141

12 Do, K.D., Pan, J.: 'Global output-feedback path tracking of unicycle-type mobile robots', *Robot. Comput. – Integr. Manuf.*, 2006, **22**, pp. 166–179

13 Oliveira de, V.M., De Pieri, E.R., Lages, W.F.: 'Mobile robot control using sliding mode and neural network'. Proc. Seventh IFAC Symp. on Robot Control, Wroclaw, Elsevier, 2003, vol. 2, pp. 581–586

14 Hamerlain, F., Achour, K., Floquet, T., Perruquetti, W.: 'Higher Order Sliding Mode Control of wheeled mobile robots in the presence of sliding effects'. 44th IEEE Conf. on Decision and Control, and the European Control Conf., Seville, Spain, December, 2004, pp. 12–15

15 Sun, S.: 'Designing approach on trajectory-tracking control of mobile robot', *Rob. Comput. – Integr. Manuf.*, 2005, **21**, (1), pp. 81–85

16 Park, B.S., Yoo, S.J., Park, J.B., Choi, Y.H.: 'Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty', *IEEE Trans. Control Syst. Technol.*, 2009, **17**, (1), pp. 207–214

17 De La Cruz, C., Carelli, R.: 'Dynamic modeling and centralized formation control of mobile robots'. 32nd Annual Conf. on IEEE Industrial Electronics Society IECON, Paris, 2006

18 Xu, D., Zhao, D., Yi, J., Tan, X.: 'Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network based sliding mode approach', *IEEE Trans. Syst. Man Cybern. B*, 2009, **39**, (3), pp. 788–799

19 Kim, M.S., Shin, J.H., Lee, J.J.: 'Design of a robust adaptive controller for a mobile robot'. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2000, pp. 1816–1821

20 Rossomando, F.G., Soria, C., Carelli, R.: 'Autonomous mobile robot navigation using RBF neural compensator', *Control Eng. Pract., Elsevier*, 2010, **19**, (3), pp. 215–222

21 Rossomando, F.G., Soria, C., Patiño, D., Carelli, R.: 'Model reference adaptive control for mobile robots in trajectory tracking using radial basis function neural networks', *Latin Am. Applicated Res. (LAAR)*, 2011, **41**, (2), pp. 177–182

22 Slotine, J.J.E., Li, W.: 'Applied nonlinear control' (Prentice-Hall, 1991)

23 Lin, W.-S., Chen, C.-S.: 'Robust adaptive sliding mode control using fuzzy modelling for a class of uncertain MIMO nonlinear systems', *IEE Proc. – Control Theory Appl.*, 2002, **149**, (3), pp. 193–201

24 Cyhenko, G.: 'Approximation by superpositions of a sigmoidal function', *Math. Control Signal, Syst.*, 1989, **2**, pp. 303–314

10 Appendix: parameters description

The identified parameters can be described by

$$\begin{aligned}
\vartheta_1 &= \left(\frac{((R_a/k_a)(MR_t r + 2I_e) + 2rk_{\mathrm{DT}})}{2rk_{\mathrm{PT}}}\right)\\
\vartheta_2 &= \left(\frac{((R_a/k_a)(I_e d^2 + 2R_t r(I_z + Mb^2)) + 2rdk_{\mathrm{DR}})}{2rdk_{\mathrm{PR}}}\right)\\
\vartheta_3 &= \left(\frac{(R_a/k_a)MbR_t}{2k_{\mathrm{PT}}}\right)\\
\vartheta_4 &= \left(\frac{(R_a/k_a)((k_a k_b/R_a) + B_e)}{rk_{\mathrm{PT}}} + 1\right)\\
\vartheta_5 &= \left(\frac{(R_a/k_a)MbR_t}{\mathrm{d}k_{\mathrm{PR}}}\right)\\
\vartheta_6 &= \left(\frac{(R_a/k_a)((k_a k_b/R_a) + B_e)d}{2rk_{\mathrm{PR}}} + 1\right)
\end{aligned} \tag{57}$$

In these relations, M is the robot mass; r is the radius of the left and right wheels; k_b is equal to the electro motoric 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}, y k_{DT} are positive constants; I_e and B_e are the moment of inertia and the viscous friction coefficient both belonging to the combination of motor, gear box and wheel; and R_t is the nominal radius of the wheel.