

## Control and Cybernetics

vol. 44 (2015) No. 1

### Trajectory tracking of a wheeled mobile robot with uncertainties and disturbances: proposed adaptive neural control\*

by

Nardênio Almeida Martins<sup>1,3</sup>, Maycol de Alencar<sup>1</sup>, Warody Claudinei Lombardi<sup>2</sup>, Douglas Wildgrube Bertol<sup>3</sup>, Edson Roberto De Pieri<sup>3</sup> and Humberto Ferasoli Filho<sup>4</sup>

<sup>1</sup>Universidade Estadual de Maringá, Departamento de Informática, Avenida Colombo, 5790, CEP 87020-900, Maringá, PR, Brasil  
nardenio@din.uem.br, maalencar@msn.com

<sup>2</sup>Lyon Université, INSA - Institut National des Sciences Appliquées, 20, avenue Albert Einstein, 69621 Villeurbanne Cedex, France  
warody.lombardi@lss.supelec.fr

<sup>3</sup>Universidade Federal de Santa Catarina, Departamento de Automação e Sistemas, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Caixa Postal 476, CEP 88040-900, Florianópolis, SC, Brasil  
{nardenio, dwbertol, edson}@das.ufsc.br

<sup>4</sup>Universidade Estadual Paulista Júlio de Mesquita Filho, Faculdade de Ciências, Departamento de Computação, Avenida Luiz Edmundo Carrijo Coube, Caixa Postal 473, CEP 17033-360, Bauru, SP, Brasil  
ferasoli@fc.unesp.br

**Abstract:** This paper analyses a trajectory tracking control problem for a wheeled mobile robot, using integration of a kinematic neural controller (KNC) and a torque neural controller (TNC), in which both the kinematic and dynamic models contain uncertainties and disturbances. The proposed adaptive neural controller (PANC) is composed of the KNC and the TNC and is designed with use of a modeling technique of Gaussian radial basis function neural networks (RBFNNs). The KNC is a variable structure controller, based on the sliding mode theory and is applied to compensate for the disturbances of the wheeled mobile robot kinematics. The TNC is an inertia-based controller composed of a dynamic neural controller (DNC) and a robust neural compensator (RNC) applied to compensate for the wheeled mobile robot dynamics, bounded unknown disturbances, and neural network modeling errors. To minimize the problems found in practical implementations of the classical variable structure controllers (VSC) and sliding mode controllers (SMC), and

---

\*Submitted: October 2012; Accepted: January 2015.

to eliminate the chattering phenomenon, the nonlinear and continuous KNC and RNC of the TNC are applied in lieu of the discontinuous components of the control signals that are present in classical forms. Additionally, the PANC neither requires the knowledge of the wheeled mobile robot kinematics and dynamics nor the time-consuming training process. Stability analysis, convergence of the tracking errors to zero, and the learning algorithms for the weights are guaranteed based on the Lyapunov method. Simulation results are provided to demonstrate the effectiveness of the proposed approach.

**Keywords:** wheeled mobile robot, trajectory tracking, kinematic control, variable structure control, dynamic control, sliding mode theory, neural networks, Lyapunov theory

## 1. Introduction

A wheeled mobile robot of type (2,0) is frequently used in the literature as a test platform due to its mechanical simplicity and adequate representation of the challenges of the control problem, which is treated in this paper (Campion and Chung, 2008). However, considering the practical applications of this nonholonomic system, the difficulty in exact modeling of practical systems, and the unavoidable disturbances in control, effective tracking control design of uncertain nonholonomic systems requires further study.

In addition to the stabilization problem in nonholonomic systems, the tracking control problem appears to be at least as interesting in practice. Based on whether the system is described by a kinematic model or a dynamic model, the tracking control problem is classified as either a kinematic or a dynamic tracking control problem. The kinematic tracking control problem has been studied over the last two decades (Morin and Samson, 2008). The dynamic tracking control problem of the nonholonomic system has received attention because most practical nonholonomic mechanical systems are dynamic systems and contain uncertainties and disturbances (Coelho and Nunes, 2005; Dong and Kuhnert, 2005). The motion of a wheeled mobile robot is eventually driven by force or torque; thus, it is more suitable to design a controller that integrates the nonholonomic kinematics controller with the dynamic model of the wheeled mobile robot (Oh et al., 2004). Moreover, it is impossible to obtain the exact kinematics and the dynamics of a wheeled mobile robot in reality. The design of a nonholonomic kinematic controller and dynamic controller under these conditions is still an open question. In the literature, most of the results on the dynamic tracking control problem for nonholonomic systems have been proposed based on the assumption that the kinematics of the system are exactly known and uncertainties and disturbances exist only in the dynamics of the system. However, in practice, uncertainties and disturbances exist in both the kinematics and dynamics.

This paper describes the adaptive neural controller, PANC, which addresses the problem of integration of the KNC (based on the VSC) and the TNC (based on the inertia-related control and SMC) and considers the presence of distur-

bances in the kinematic and dynamic models, as well as the unknown dynamic parameters.

The VSC and SMC designs use a high-speed switching control law to drive the nonlinear predefined state trajectories onto a specified surface (known as the sliding or switching surface) to attain the conventional goals of control, i.e., stabilization and tracking.

Due to robustness properties against uncertainties, modeling imprecision, and disturbances, the VSC and SMC have become quite popular and are used in many application areas (Utkin et al., 2009). However, this control scheme contains important drawbacks that limit its practical applicability, such as high frequency switching (chattering) and large authority control, which deteriorate the system performance (Shuwen et al., 2000). The first drawback mentioned is due to control actions that are discontinuous on the sliding surfaces and cause high frequency switching in a boundary of the sliding surfaces. This high frequency switching might excite unmodeled dynamics and impose undue wear on the actuators, such that the control law will not be deemed acceptable. The second drawback mentioned is based on the requirement of a priori knowledge of the boundary of uncertainty in the compensators. If the boundary is unknown, a large value must be applied to the gain of the discontinuous component of the control signal, and this large control gain may intensify the high frequency switching on the sliding surfaces.

There are studies, which have introduced the application of soft computing methodologies, such as artificial neural networks, to improve performance and reduce problems found in practical implementations of VSC and SMC, as mentioned in Yu and Kaynak (2009). In this paper, radial basis function neural networks (RBFNNs) are applied to avoid chattering and compensate for uncertainties and disturbances, because the structure of an RBFNN is simpler than that of a multi-layer perceptron (MLP), the learning rate of an RBFNN is generally faster than that of an MLP, and an RBFNN is easily mathematically tractable (Haykin, 2008). Moreover, it should be emphasized at this point that RBFNNs and fuzzy rule systems are functionally equivalent under certain mild conditions (Jin and Sendhoff, 2003; Ciftcioglu, 2003).

In contrast with other works that consider the kinematics of wheeled mobile robots without disturbances and applying the VSC and SMC theories to the wheeled mobile robots (Liu et al., 2011; Elyoussef et al., 2010; Solea et al., 2009; Kungpeng et al., 2009; Li et al., 2009; Lee et al., 2009; Park et al., 2009; Defoort et al., 2007; Chwa, 2004; Chwa et al., 2002; Yang and Kim, 1999a, b; Shim et al., 1995; Guldner and Utkin, 1994), the contributions of this paper are as follows:

- The PANC (KNC plus TNC) is implemented based on partitioning of the RBFNNs into several smaller subnets to obtain more efficient computation;
- The KNC is used to replace the discontinuous portion of the classical VSC, thus avoiding chattering as well as suppressing the kinematic disturbances without the need for prior knowledge of their boundaries;

- The DNC of the TNC is used to compensate for the wheeled mobile robot dynamics or unknown dynamic parameters (parametric uncertainties);
- The RNC of the TNC is used in the replacement of the discontinuous components of the classical SMC to avoid chattering and to suppress the bounded unknown disturbances without the need for any prior knowledge of their boundaries;
- The stability analysis, convergence of the wheeled mobile robot control system, and the learning algorithms for the weights are proven using Lyapunov theory;
- The PANC neither requires the knowledge of the wheeled mobile robot kinematics and dynamics nor the time-consuming training process.

This paper is organized as follows. Section 2 presents the kinematic and dynamic models for wheeled mobile robots with disturbances, the GL (Ge-Lee) matrix and operator, RBFNNs, neural networks modeling via RBFNNs, the trajectory tracking control problem, and the corresponding error dynamics. The PANC (KNC plus TNC) for reference trajectory tracking and stability analysis are described in Section 3. Section 4 shows the simulation results, and Section 5 presents the conclusions.

## 2. Problem formulation

This section describes the kinematic and dynamic models, GL (Ge-Lee) matrix and operator, RBFNNs, neural networks modeling by RBFNNs, trajectory tracking control problem, and error dynamics for a wheeled mobile robot.

### 2.1. Kinematics and dynamics of a wheeled mobile robot

A typical example of a wheeled mobile robot is shown in Fig. 1. The wheeled mobile robot contains two driving wheels, mounted on the same axis, and a free front wheel. The two driving wheels are independently driven by two actuators to achieve motion and orientation.

In addition, the position of the wheeled mobile robot in the Cartesian inertial frame  $\{X_o, O, Y_o\}$  can be described by a vector  $\overrightarrow{OC_M}$ , the orientation  $\theta$  between the wheeled mobile robot base frame  $\{X_c, C_M, Y_c\}$ , and the Cartesian inertial frame, where  $C_M$  identifies the center of mass coordinates (guidance point), with  $P_M$ ,  $d$ ,  $r$ , and  $2R$  representing the intersection of the axis of symmetry with the driven wheel axis, the distance from the point  $C_M$  to the point  $P_M$ , the radius of the wheels, and the distance between the driven wheels, respectively.

The posture vector  $q \in \mathbb{R}^3$  of the wheeled mobile robot is described by three generalized coordinates in the form of  $q = [x_c \ y_c \ \theta]^T$ , where  $x_c$  and  $y_c$  are the coordinates of  $C_M$ .

Under the conditions of pure rolling and non-slipping, considering  $d = 0$ , and taking into account the measurement noise, modeling uncertainties, and disturbances, a kinematic model of the wheeled mobile robot can be described by

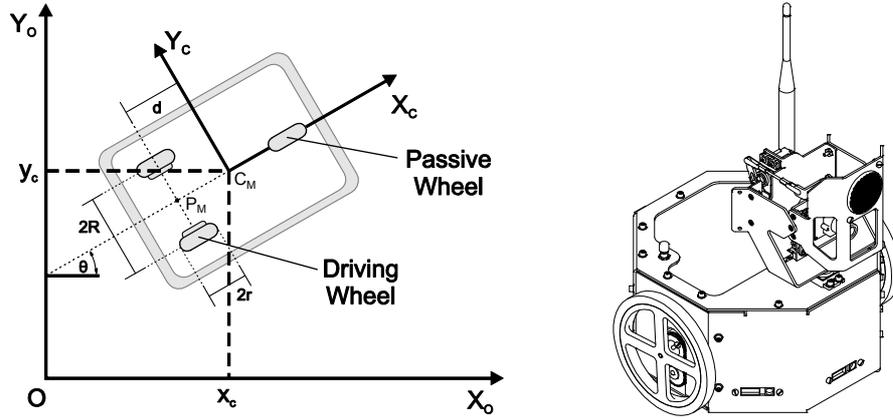


Figure 1. Wheeled mobile robot and coordinate systems

the following:

$$\dot{q} = S(q) (v(t) + d_v(t)), \quad S(q) = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix}, \quad (1)$$

where  $\dot{q}$  are the constrained velocities in Cartesian coordinates,  $S(q)$  is a Jacobian matrix,  $v(t) = [v_l \ \omega_a]^T$  represents the linear and angular velocities of the wheeled mobile robot at point  $C_M$ , and  $d_v(t) = [\delta_{v_l} \ \delta_{\omega_a}]^T$  represents the disturbances in  $v(t)$  only, which are assumed to have the following upper bounds:

$$|\delta_{v_l}| < \varepsilon_{v_l}, \quad |\delta_{\omega_a}| < \varepsilon_{\omega_a}, \quad (2)$$

with  $\varepsilon_{v_l}$  and  $\varepsilon_{\omega_a}$  being positive bounded constants. Thus, the kinematic model of a wheeled mobile robot (1) may be subject to the so-called matched disturbance (Martins et al., 2012).

In Fierro and Lewis (1998), if the surface friction  $F(\dot{q})$  and gravitational torques  $G(q)$  are disregarded, the wheeled mobile robot dynamics for control purposes are described as follows:

$$\bar{H}(q)\dot{v} + \bar{C}(q, \dot{q})v + \bar{\tau}_{cp} = \bar{D}(q)\tau = \bar{\tau}, \quad (3)$$

where:

$\bar{H}(q) = S^T(q)H(q)S(q)$  is the symmetric positive definite inertia matrix,

$\bar{C}(q, \dot{q}) = S^T(q) \left( H(q)\dot{S}(q, \dot{q}) + C(q, \dot{q})S(q) \right)$  is the centripetal and Coriolis matrix,

$\bar{\tau} = \bar{D}(q)\tau = S^T(q)D(q)\tau$  is the input vector,

and

$\bar{\tau}_{cp} = S^T(q) \left( H(q)S(q)\dot{d}_v + H(q)\dot{S}(q, \dot{q})d_v + C(q, \dot{q})S(q)d_v + \tau_p \right)$  denotes the bounded unknown disturbances, including the unstructured and unmodeled dynamics.

It is necessary to emphasize that the pattern properties of the boundedness of  $\bar{H}(q)$ , the bounded norm of the  $\bar{C}(q, \dot{q})$  and  $\bar{\tau}_{cp}$ , and the skew-symmetry  $\dot{\bar{H}}(q, \dot{q}) - 2\bar{C}(q, \dot{q})$  must be considered in the stability analysis of the control system.

## 2.2. GL matrix and operator

This section briefly discusses the definition of GL matrix, which is denoted by  $\{\cdot\}$ , and its product operator by “ $\bullet$ ”. Readers are referred to Ge (1996) for a detailed discussion on the motivation for using the GL matrix. To avoid any possible confusion,  $[\cdot]$  is used to denote the ordinary vector and matrix.

Assume that  $I_0$  is the set of integers and  $\partial_{kj}, \zeta_{kj} \in R^{n_{kj}}$ , where  $n_{kj} \in I_0$ ,  $k = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, n$ . The GL row vector  $\{\partial_k\}$  and its transpose  $\{\partial_k\}^T$  are defined in the following manner:

$$\{\partial_k\} = \{\partial_{k1} \ \partial_{k2} \ \dots \ \partial_{kn}\}, \quad \{\partial_k\}^T = \{\partial_{k1}^T \ \partial_{k2}^T \ \dots \ \partial_{kn}^T\}. \quad (4)$$

The GL matrix  $\{\Theta\}$  and its transpose  $\{\Theta\}^T$  are defined, accordingly, as follows:

$$\{\Theta\} = \begin{Bmatrix} \partial_{11} & \partial_{12} & \dots & \partial_{1n} \\ \partial_{21} & \partial_{22} & \dots & \partial_{2n} \\ \dots & \dots & \dots & \dots \\ \partial_{n1} & \partial_{n2} & \dots & \partial_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\partial_1\} \\ \{\partial_2\} \\ \{\cdot\} \\ \{\partial_n\} \end{Bmatrix}, \quad (5)$$

$$\{\Theta\}^T = \begin{Bmatrix} \partial_{11}^T & \partial_{12}^T & \dots & \partial_{1n}^T \\ \partial_{21}^T & \partial_{22}^T & \dots & \partial_{2n}^T \\ \dots & \dots & \dots & \dots \\ \partial_{n1}^T & \partial_{n2}^T & \dots & \partial_{nn}^T \end{Bmatrix}. \quad (6)$$

For a given GL matrix  $\{\Xi\}$

$$\{\Xi\} = \begin{Bmatrix} \zeta_{11} & \zeta_{12} & \dots & \zeta_{1n} \\ \zeta_{21} & \zeta_{22} & \dots & \zeta_{2n} \\ \dots & \dots & \dots & \dots \\ \zeta_{n1} & \zeta_{n2} & \dots & \zeta_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\zeta_1\} \\ \{\zeta_2\} \\ \{\cdot\} \\ \{\zeta_n\} \end{Bmatrix}, \quad (7)$$

and the GL product of  $\{\Theta\}^T$  and  $\{\Xi\}$  is an  $n \times n$  matrix defined as follows:

$$\left[ \{\Theta\}^T \bullet \{\Xi\} \right] = \begin{Bmatrix} \partial_{11}^T \zeta_{11} & \partial_{12}^T \zeta_{12} & \dots & \partial_{1n}^T \zeta_{1n} \\ \partial_{21}^T \zeta_{21} & \partial_{22}^T \zeta_{22} & \dots & \partial_{2n}^T \zeta_{2n} \\ \dots & \dots & \dots & \dots \\ \partial_{n1}^T \zeta_{n1} & \partial_{n2}^T \zeta_{n2} & \dots & \partial_{nn}^T \zeta_{nn} \end{Bmatrix}. \quad (8)$$

The GL product of a square matrix and a GL row vector is defined as follows. Let  $\Lambda_k = \Lambda_k^T = [\pi_{k1} \ \pi_{k2} \ \dots \ \pi_{kn}]$ ,  $\pi_{kj} \in \mathbb{R}^{m \times n_{kj}}$ ,  $m = \sum_{j=1}^n n_{kj}$ , then one obtains the following:

$$\Lambda_k \bullet \{\zeta_k\} = \{\Lambda_k\} \bullet \{\zeta_k\} = [\pi_{k1}\zeta_{k1} \ \pi_{k2}\zeta_{k2} \ \dots \ \pi_{kn}\zeta_{kn}] \in \mathbb{R}^{m \times n}. \quad (9)$$

Note that the GL product should be computed first in a mixed matrix product. For instance, in  $\{A\} \bullet \{B\}C$ , the matrix  $[\{A\} \bullet \{B\}]$  should be computed first, followed by the multiplication of  $[\{A\} \bullet \{B\}]$  with matrix  $C$ .

### 2.3. Radial basis function neural networks (RBFNNs)

In the field of control engineering, neural networks are often used to approximate a given nonlinear function  $f(y_{ent})$  up to a small error tolerance. The function approximation problem can be stated formally as follows.

*Definition 1:* Given that  $f(y_{ent}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a continuous function defined on the set  $y_{ent} \in \mathbb{R}^n$ , and  $\hat{f}(W, y_{ent}) : \mathbb{R}^{l \times m} \times \mathbb{R}^n \rightarrow \mathbb{R}^m$  is an approximating function that depends continuously on the parameter matrix  $W$  and  $y_{ent}$ , the approximation problem is designed to determine the optimal parameter  $W^*$  such that, for a certain metric (or distance function)  $d_f$ ,

$$d_f(\hat{f}(W^*, y_{ent}), f(y_{ent})) \leq \varepsilon_{NN}, \quad (10)$$

for an acceptably small  $\varepsilon_{NN}$  (Haykin, 2008).

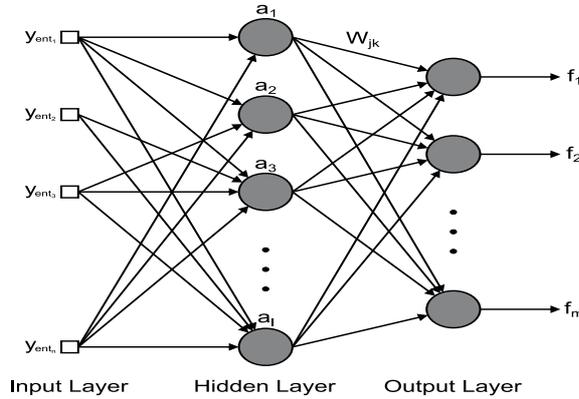


Figure 2. Schematic diagram of a RBFNN

In this paper, the Gaussian RBFNN is considered, which is a particular network architecture that uses  $l$  Gaussian functions of the following form:

$$a_i(y_{ent}) = \exp\left(-\frac{(y_{ent} - m_i)^T (y_{ent} - m_i)}{2\sigma_v^2}\right), \quad i = 1, 2, \dots, l, \quad (11)$$

where  $m_i \in \mathfrak{R}^n$  is the vector of centers, and  $\sigma_v^2 \in \mathfrak{R}$  is the variance. As shown in Fig. 2, each Gaussian RBFNN consists of three layers: the input layer, the hidden layer that contains the Gaussian function, and the output layer. At the input layer, the input space is divided into grids with a basis function at each node defining a receptive field in  $\mathfrak{R}^n$ . The output of the network,  $\hat{f}(W, y_{ent})$ , is given by the following:

$$\hat{f}(W, y_{ent}) = W^T a(y_{ent}), \quad (12)$$

where  $a(y_{ent}) = [a_1(y_{ent}) \ a_2(y_{ent}) \ \dots \ a_l(y_{ent})]^T$  is the vector of the basis function. Note that only the connections from the hidden layer to the output are weighted.

The Gaussian RBFNN has been quite successful in representing the complex nonlinear function. It has been shown that a linear superposition of Gaussian radial basis function (GRBF) gives an optimal mean square approximation to an unknown function that is infinitely differentiable, the values of which are specified by a finite set of points in  $\mathfrak{R}^n$ . Furthermore, it has been proven that any continuous functions (not necessary infinitely smooth) can be uniformly approximated by a linear combination of Gaussians (Haykin, 2008).

#### 2.4. Neural networks modeling by RBFNNs

Based on (3), it can be verified that  $\bar{H}(q)$ , and  $P_k(\cdot)$  are functions of  $q$  and any variable expressed by “.” only; thus, static neural networks are sufficient to model them. Assuming that  $\bar{h}_{kj}(q)$  (see Fig. 3),  $p_k(\cdot)$  (see Fig. 4) can be modeled as follows:

$$\bar{h}_{kj}(q) = \sum_l W_{\bar{h}_{kjl}} \xi_{\bar{h}_{kjl}}(q) + \varepsilon_{\bar{h}_{kj}}(q) = W_{\bar{h}_{kj}}^T \xi_{\bar{h}_{kj}}(q) + \varepsilon_{\bar{h}_{kj}}(q), \quad (13)$$

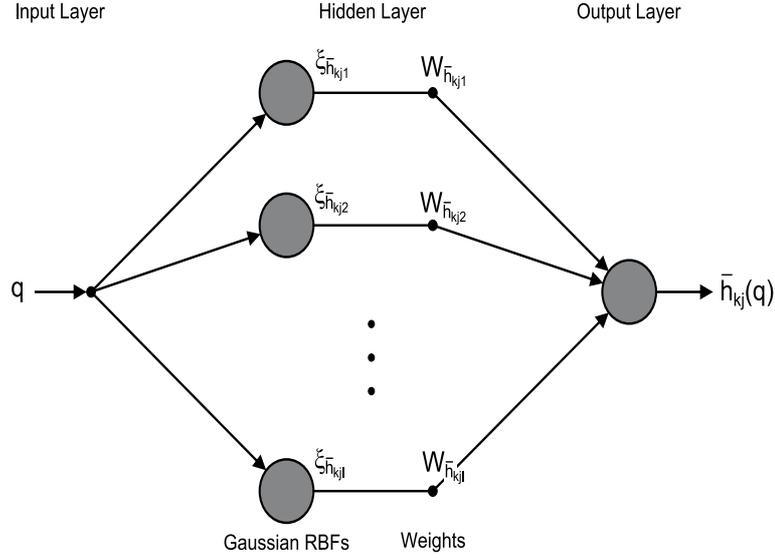
$$p_k(\cdot) = \sum_l W_{p_{kl}} \xi_{p_{kl}}(\cdot) + \varepsilon_{p_k}(\cdot) = W_{p_k}^T \xi_{p_k}(\cdot) + \varepsilon_{p_k}(\cdot), \quad (14)$$

where  $W_{\bar{h}_{kjl}}, W_{p_{kl}} \in \mathfrak{R}$  are the weights of the neural networks;  $\xi_{\bar{h}_{kjl}}(q), \xi_{p_{kl}}(\cdot) \in \mathfrak{R}$  are the GRBFs with their respective input vectors  $q$  and “.” only; and  $\varepsilon_{\bar{h}_{kj}}(q), \varepsilon_{p_k}(\cdot) \in \mathfrak{R}$  are the modeling errors of  $\bar{h}_{kj}(q)$  and  $p_k(\cdot)$ , respectively, and are assumed to be bounded.

In Figs. 3 and 4,  $W_{\bar{h}_{kj}}$ , and  $W_{p_k}$  are the vectors of the weights of RBFNNs;  $\xi_{\bar{h}_{kj}}(q)$ , and  $\xi_{p_k}(\cdot)$  are the vectors of GRBFs, expressed similarly as in (11), and the  $l$ -th element is defined as follows:

$$\xi_{\bar{h}_{kjl}}(q) = \exp\left(\frac{-\|q - m_{\bar{h}_{kjl}}\|^2}{2\sigma_{\bar{h}_{kjl}}^2}\right) = \exp\left(\frac{-(q - m_{\bar{h}_{kjl}})^T(q - m_{\bar{h}_{kjl}})}{2\sigma_{\bar{h}_{kjl}}^2}\right), \quad (15)$$

$$\xi_{p_{kl}}(\cdot) = \exp\left(\frac{-(\cdot - m_{p_{kl}})^T(\cdot - m_{p_{kl}})}{2\sigma_{p_{kl}}^2}\right), \quad (16)$$

Figure 3. Implementation of  $\bar{h}_{kj}(q)$ 

where  $m_{\bar{h}_{kjl}}, m_{p_{kl}} \in \mathfrak{R}^n$  are the vectors of centers, and  $\sigma_{\bar{h}_{kjl}}^2, \sigma_{p_{kl}}^2 \in \mathfrak{R}$  are the variances, respectively.

Bearing in mind that  $\bar{C}(q, \dot{q})$ , as used in (3), is a dynamic neural network, because it is a function of  $q$  and  $\dot{q}$ , its modeling is required. Assume that  $\bar{c}_{kj}(q, \dot{q})$  (see Fig. 5) can be modeled as follows:

$$\bar{c}_{kj}(q, \dot{q}) = \sum_l W_{\bar{c}_{kjl}} \xi_{\bar{c}_{kjl}}(z) + \varepsilon_{\bar{c}_{kj}}(z) = W_{\bar{c}_{kj}}^T \xi_{\bar{c}_{kj}}(z) + \varepsilon_{\bar{c}_{kj}}(z), \quad (17)$$

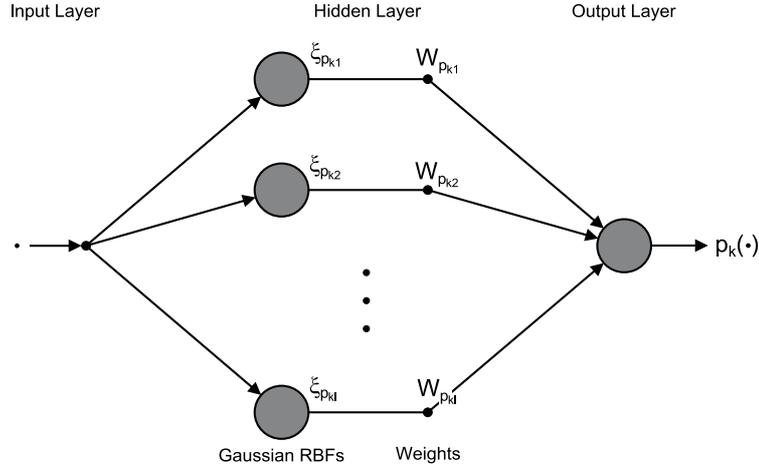
where  $z = [q^T \dot{q}^T]^T \in R^{2n}$ ,  $W_{\bar{c}_{kjl}} \in \mathfrak{R}$  is the weight of the neural network,  $\xi_{\bar{c}_{kjl}}(z) \in \mathfrak{R}$  is a GRBF with the respective input vector  $z$ , and  $\varepsilon_{\bar{c}_{kj}}(z) \in \mathfrak{R}$  is the modeling error of  $\bar{c}_{kj}(q, \dot{q})$  and is also assumed to be bounded.

In Fig. 5,  $W_{\bar{c}_{kj}}$  is the vector of weights of RBFNNs,  $\xi_{\bar{c}_{kj}}(q)$  is the vector of GRBFs expressed similarly as in (11), and the  $l$ -th element is defined as follows:

$$\xi_{\bar{c}_{kjl}}(z) = \exp\left(-\frac{\|z - m_{\bar{c}_{kjl}}\|^2}{2\sigma_{\bar{c}_{kjl}}^2}\right) = \exp\left(-\frac{(z - m_{\bar{c}_{kjl}})^T(z - m_{\bar{c}_{kjl}})}{2\sigma_{\bar{c}_{kjl}}^2}\right), \quad (18)$$

where  $m_{\bar{c}_{kjl}} \in \mathfrak{R}^n$  is the vector of centers, and  $\sigma_{\bar{c}_{kjl}}^2 \in \mathfrak{R}$  is the variance.

Previously grounded in (3), the matrices  $\bar{H}(q)$  and  $\bar{C}(q, \dot{q})$  of the wheeled

Figure 4. Implementation of  $p_k(\cdot)$ 

mobile robot dynamics can be expressed by the following:

$$\begin{aligned}\bar{H}(q) &= \left[ \{W_{\bar{H}}\}^T \bullet \{\xi_{\bar{H}}(q)\} \right] + E_{\bar{H}}(q), \\ \bar{C}(q, \dot{q}) = \bar{C}(z) &= \left[ \{W_{\bar{C}}\}^T \bullet \{\xi_{\bar{C}}(z)\} \right] + E_{\bar{C}}(z),\end{aligned}\quad (19)$$

where  $\{W_{\bar{H}}\}$ ,  $\{\xi_{\bar{H}}(q)\}$ ,  $\{W_{\bar{C}}\}$ , and  $\{\xi_{\bar{C}}(z)\}$  are GL matrices, and their respective elements are  $W_{\bar{h}_{kj}}$ ,  $\xi_{\bar{h}_{kj}}(q)$ ,  $W_{\bar{c}_{kj}}$ , and  $\xi_{\bar{c}_{kj}}(z)$ . The  $E_{\bar{H}}(q) \in R^{n \times n}$  and  $E_{\bar{C}}(z) \in R^{n \times n}$  are matrices, and their modeling error elements are  $\varepsilon_{\bar{h}_{kj}}(q)$  and  $\varepsilon_{\bar{c}_{kj}}(z)$ , respectively.

It is important to emphasize that a vector  $P(\cdot)$  can be modeled with static neural networks because it is a function of a variable only. Thus,  $P(\cdot)$  results from the following equation:

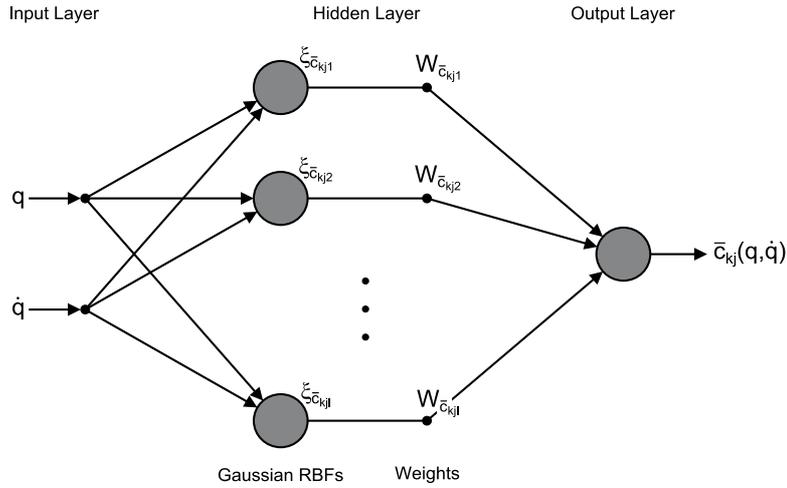
$$P(\cdot) = \left[ \{W_P\}^T \bullet \{\xi_P(\cdot)\} \right] + E_P(\cdot), \quad (20)$$

where  $\{W_P\}$  and  $\{\xi_P(\cdot)\}$  are GL vectors with their respective elements being  $W_{p_k}$  and  $\xi_{p_k}(\cdot)$ , and  $E_P(\cdot) \in R^n$  is a vector with the modeling error elements being  $\varepsilon_{p_k}(\cdot)$ . Based on (20),  $P_v(\sigma^*)$  and  $P_s(s)$  will be declared later.

## 2.5. Error dynamics for a wheeled mobile robot

To formulate the trajectory tracking control problem, a reference trajectory is generated by the following reference kinematic model:

$$\dot{q}_r = S(q_r)v_r, \quad \dot{x}_r = v_{l_r} \cos(\theta_r), \quad \dot{y}_r = v_{l_r} \sin(\theta_r), \quad \dot{\theta}_r = \omega_{a_r}, \quad (21)$$

Figure 5. Implementation of  $\bar{c}_{kj}(q, \dot{q})$ 

where  $q_r = [x_r \ y_r \ \theta_r]^T \in \mathfrak{R}^3$  denotes the reference posture of the wheeled mobile robot, the structure of  $S(q_r)$  is defined as in (1), and  $v_r = [v_{l_r} \ \omega_{a_r}]^T$  denotes the reference linear and angular velocities of the wheeled mobile robot, respectively. With respect to (21), it is assumed that the signal  $v_r(t)$  is chosen to produce the desired motion and that  $v_r(t)$ ,  $\dot{v}_r(t)$ ,  $q_r(t)$ , and  $\dot{q}_r(t)$  are bounded for all time.

The trajectory tracking control problem of a wheeled mobile robot is solved by designing a control input  $v(t) = v_c(t) = [v_l \ \omega_a]^T$  such that the system (1) follows reference (21) despite disturbances. In fact, the aim is to converge the posture tracking errors ( $e_x = x_r - x_c$ ,  $e_y = y_r - y_c$ ,  $e_\theta = \theta_r - \theta$ ) to zero while respecting the following constraints:

$$|v_l| \leq v_{l_{\max}}, \quad |\omega_a| \leq \omega_{a_{\max}}. \quad (22)$$

Converting the tracking errors in the inertial frame to the wheeled mobile robot frame, the posture error equation of the wheeled mobile robot can be denoted as follows:

$$\tilde{z} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}. \quad (23)$$

The error dynamics can be obtained from the time derivative of (8) as follows:

$$\dot{\tilde{z}} = \begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} v_{l_r} \cos(\tilde{\theta}) \\ v_{l_r} \sin(\tilde{\theta}) \\ \omega_{a_r} \end{bmatrix} + \begin{bmatrix} -1 & \tilde{y} \\ 0 & -\tilde{x} \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_l + \delta_{v_l} \\ \omega_a + \delta_{\omega_a} \end{bmatrix}. \quad (24)$$

### 3. Control design for trajectory tracking

In this section, the PANC (KNC plus TNC) is designed for kinematic and dynamic models with uncertainties and disturbances. The KNC is based on VSC theory, and the TNC consists of a DNC and a RNC, which are based on the inertia-related control and SMC theories. In the KNC and RNC, the RBFNNs are used as replacements for the discontinuous components of the classical VSC and SMC to avoid chattering and to suppress uncertainties and disturbances. For such a development, the selection for the sliding surfaces and a brief description of the generic modeling of nonlinear systems for the VSC and SMC designs are required (Utkin et al., 2009).

#### 3.1. Choice of sliding surfaces

The VSC provides feedback control with high-speed switching, whose action takes place in two phases: the reaching phase and the sliding phase. In the reaching phase, the state trajectories of the system (linear or nonlinear) are led to a location in the state space chosen by the designer. In general, this location is defined by the linear surfaces of the control errors ( $\tilde{z} = [\tilde{x} \ \tilde{y} \ \tilde{\theta}]^T$ ), known as the switching or sliding surfaces ( $\sigma$ ), and each one of them is described by the following:

$$\sigma_i(\tilde{z}, t) = c_i^T \tilde{z}_i = 0, \quad i = 1, 2, \quad (25)$$

In the sliding phase, the state trajectories are forced to remain on the sliding surfaces. By choosing the appropriate constants  $c_i^T$  in (25), the errors will tend exponentially to zero, according to the standard determined by these constants during the sliding phase.

Thus, to control the kinematic model (1), the sliding surfaces are selected as follows:

$$\sigma(\tilde{z}, t) = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} k_1 \tilde{x} \\ k_2 \tilde{y} + k_3 \tilde{\theta} \end{bmatrix}, \quad (26)$$

where  $k_1$ ,  $k_2$ , and  $k_3$  are positive constants. One sliding surface,  $\sigma_i$ ,  $i = 1, 2$ , is associated with each control input.

#### 3.2. Generic model for nonlinear systems

The derivation of the VSC and the associated properties are carried out directly for an important class of nonlinear systems, whose model in the form of state equations is given by the following:

$$\dot{\tilde{z}}(t) = A(\tilde{z}, \rho, t) + B(\tilde{z}, \rho, t)v_c(\tilde{z}, t) + d_b(t), \quad (27)$$

with  $A(\tilde{z}, \rho, t) = A_0(\tilde{z}, t) + \Delta A(\tilde{z}, \rho, t)$  and  $B(\tilde{z}, \rho, t) = B_0(\tilde{z}, t) + \Delta B(\tilde{z}, \rho, t)$ , where  $\tilde{z}(t)$  is the vector of states,  $A(\tilde{z}, \rho, t)$  is the vector of nonlinear functions,  $v_c(\tilde{z}, t)$  is the vector of control inputs,  $\rho(\tilde{z}, t)$  is the vector of parametric

uncertainties,  $B(\tilde{z}, \rho, t)$  is the matrix of nonlinear functions,  $\Delta A(\tilde{z}, \rho, t)$  and  $\Delta B(\tilde{z}, \rho, t)$  are the respective vector and the matrix representing the disturbances in the system arising from the parametric uncertainties,  $d_b(t)$  is the vector of external disturbances, and  $A_0(\tilde{z}, t)$  and  $B_0(\tilde{z}, t)$  refer to the vector and the matrix of nominal parameters, respectively.

The aim of this study is the derivation of a VSC that is robust to the disturbances present in the kinematic model (1). To ensure the robustness of the controller, the disturbances should be bounded, the matrix  $B(\tilde{z}, \rho, t)$  should be nonsingular, and the following matching conditions must be satisfied:

$$\Delta A(\tilde{z}, \rho, t) = B_0(\tilde{z}, t)\tilde{a}, \quad \Delta B(\tilde{z}, \rho, t) = B_0(\tilde{z}, t)\tilde{b}, \quad d_b(t) = B_0(\tilde{z}, t)\tilde{d}_0, \quad (28)$$

which means that  $\Delta A(\tilde{z}, \rho, t)$ ,  $\Delta B(\tilde{z}, \rho, t)$ , and  $d_b(t)$  must belong to the image of  $B_0(\tilde{z}, t)$ ,  $\tilde{a}$  and  $\tilde{b}$  are the respective vector and matrix that incorporate the parametric uncertainties, and  $\tilde{d}_0$  represents the external disturbances.

Thus, the error dynamics (24) can be rewritten based on (27) and (28) as follows:

$$\dot{\tilde{z}} = A_0(\tilde{z}, t) + B_0(\tilde{z}, t)v_c(\tilde{z}, t) + d_b(t), \quad (29)$$

because there are no parametric uncertainties ( $\Delta A = 0$ ,  $\Delta B = 0$ ), and  $d_b(t)$  is defined as follows:

$$d_b(t) = B_0(\tilde{z}, t)d_v(t), \quad (30)$$

for the case of the matched disturbance (24).

### 3.3. Variable structure control design

To provide influence also on the process of reaching the sliding surfaces, the control  $v_c(\tilde{z}, t)$  will be chosen in such a manner that  $\sigma(\tilde{z}, t)$  is imposed to have the dynamics given by the following first-order differential equation:

$$\dot{\sigma}(\tilde{z}, t) = -G\text{sign}(\sigma) - K_p h(\sigma), \quad (31)$$

where  $G = \text{diag}\{G_{11}, G_{22}\}$  and  $K_p = \text{diag}\{K_{p11}, K_{p22}\}$ ,  $h(\sigma) = \sigma$  (which could be another function because  $\sigma^T h(\sigma) > 0$ ), and  $\text{sign}(\sigma) = \frac{\sigma}{|\sigma|}$  is a discontinuous function.

By rewriting (31) for the  $i$ -th sliding surface, one obtains the following:

$$\dot{\sigma}_i(\tilde{z}, t) + k_{p_i}\sigma_i = -g_i\text{sign}(\sigma_i). \quad (32)$$

Now, returning to (31) and taking into account (29) results in the following:

$$\begin{aligned} \dot{\sigma}(\tilde{z}, t) &= \frac{\partial \sigma(\tilde{z}, t)}{\partial \tilde{z}} \dot{\tilde{z}} + \frac{\partial \sigma(\tilde{z}, t)}{\partial t} \\ &= \frac{\partial \sigma}{\partial \tilde{z}} (A_0 + B_0 v_c + d_b) + \frac{\partial \sigma}{\partial t} \\ &= \frac{\partial \sigma}{\partial \tilde{z}} (A_0 + B_0 v_c) + \frac{\partial \sigma}{\partial \tilde{z}} d_b + \frac{\partial \sigma}{\partial t}, \end{aligned} \quad (33)$$

with

$$\frac{\partial \sigma(\tilde{z}, t)}{\partial \tilde{z}} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & k_3 \end{bmatrix}, \quad (34)$$

from which the following control law is derived:

$$v_c = -B_{0\sigma}^{-1} \left( A_{0\sigma} + \frac{\partial \sigma}{\partial t} + G \text{sign}(\sigma) + K_p \sigma \right), \quad (35)$$

in which

$$A_{0\sigma} = \frac{\partial \sigma}{\partial \tilde{z}} A_0 = \begin{bmatrix} k_1 v_{l_r} \cos(\tilde{\theta}) \\ k_2 v_{l_r} \sin(\tilde{\theta}) + k_3 \omega_{a_r} \end{bmatrix}, \quad (36)$$

$$B_{0\sigma} = \frac{\partial \sigma}{\partial \tilde{z}} B_0 = \begin{bmatrix} -k_1 & k_1 \tilde{y} \\ 0 & -k_2 \tilde{x} - k_3 \end{bmatrix}, \quad (37)$$

$$B_{0\sigma}^{-1} = \begin{bmatrix} -\frac{1}{k_1} & -\frac{\tilde{y}}{k_2 \tilde{x} + k_3} \\ 0 & -\frac{1}{k_2 \tilde{x} + k_3} \end{bmatrix}, \quad (38)$$

and  $k_2 = k_3 \alpha$ ,  $0 \leq \alpha \leq \frac{1}{\|\tilde{x}\|+1}$ , similarly as in the work of Cheng and Tsai (2005).

Defining

$$v_c^* = -(G \text{sign}(\sigma) + K_p \sigma), \quad (39)$$

and replacing (35) in (33) results in the following:

$$\dot{\sigma} = A_{0\sigma} - B_{0\sigma} B_{0\sigma}^{-1} \left( A_{0\sigma} + \frac{\partial \sigma}{\partial t} - v_c^* \right) + d_\sigma + \frac{\partial \sigma}{\partial t} = -G \text{sign}(\sigma) - K_p \sigma + \psi, \quad (40)$$

where using (30), one obtains:

$$d_\sigma = \frac{\partial \sigma}{\partial \tilde{z}} d_b = \frac{\partial \sigma}{\partial \tilde{z}} B_0 d_v = \begin{bmatrix} -k_1 \left( \delta_{v_i} - \tilde{y} \delta_{\omega_a} \right) \\ -(k_2 \tilde{x} + k_3) \delta_{\omega_a} \end{bmatrix}, \quad (41)$$

with  $B_{0\sigma} B_{0\sigma}^{-1} = I_n$  and  $\psi = d_\sigma$  being the disturbances in the system.

### 3.4. Stability analysis

Upon choosing the Lyapunov function candidate in the form

$$V_1 = \frac{1}{2} \sigma^T \sigma, \quad (42)$$

which is positive definite, the sliding surface will be attractive, because the control law (35) ensures that  $\dot{V}_1$  is negative definite. Using the result described by (40), an expression for  $\dot{V}_1$  is immediately obtained, i.e.,

$$\dot{V}_1 = \sigma^T \dot{\sigma} = -\sigma^T G \text{sign}(\sigma) - \sigma^T K_p \sigma + \sigma^T \psi. \quad (43)$$

Because  $\sigma^T K_p \sigma \geq 0$ , the condition  $\dot{V}_1 \leq 0$  can be expressed as follows:

$$\sigma^T G \text{sign}(\sigma) \geq \sigma^T \psi, \quad (44)$$

and this is satisfied, if the diagonal elements of  $G$  meet the following restriction:

$$g_i > |\bar{\psi}_i|, \quad \forall i. \quad (45)$$

If  $g_i > \bar{\psi}_i$ , then  $\dot{V}_1 \leq 0$  ( $\dot{V}_1 = 0$  only when  $V_1 = 0$ ), which implies that  $V_1$  may decrease to  $V_1 = 0$  exponentially; however, if  $g_i < \bar{\psi}_i$ , there is a value of  $V_1 = V_{1ss} > 0$ , for which  $\dot{V}_1 = 0$  can lead to nonzero errors. Therefore, it is possible to affirm that if the disturbances are better estimated, the results will be better.

However, for the existence and reachability of a sliding mode, it is sufficient to select  $V_1 > 0$  such that the sliding surface will be attractive, because the control law (35) ensures that  $\dot{V}_1 < 0$ . Therefore, in the derivation of (35), it is necessary that matrix  $B_{0\sigma}$  be nonsingular. Because  $G$  is a positive definite diagonal matrix in (35), the sliding mode can be enforced under the condition that the matrix  $B_{0\sigma}$  is positive definite, and the elements of the matrix  $G$  are sufficiently large. However, in this control law (35), the matrix  $B_{0\sigma}$  is nonsingular only. To solve this problem, a diagonalization method is used, which is based on the fact that the equivalent system is invariant to a nonsingular sliding surface transformation, as verified in Theorem 2, and the proof is described in DeCarlo et al. (1988). Loosely stated, Theorem 2 says that the motion in the sliding mode is independent of a nonsingular and possibly time-varying transformation of the sliding surfaces and that any nonsingular transformation with bounded derivatives will produce the same equivalent system.

In particular, consider the new sliding surfaces as follows:

$$\sigma^*(\tilde{z}, t) = \Psi(\tilde{z}, t)\sigma(\tilde{z}, t), \quad (46)$$

for an adequate nonsingular transformation  $\Psi(\tilde{z}, t) \in \mathfrak{R}^{m \times m}$ , which is defined as follows:

$$\Psi(\tilde{z}, t) = \left( \frac{\partial \sigma}{\partial \tilde{z}} B_0 \right)^T = (B_{0\sigma})^T. \quad (47)$$

Differentiating  $V_1$  and replacing (33) and (41) results in the following:

$$\dot{V}_1 = \sigma^T \dot{\sigma} = \sigma^T \frac{\partial \sigma}{\partial \tilde{z}} A_0 + \sigma^T \frac{\partial \sigma}{\partial \tilde{z}} B_0 v_c + \sigma^T \frac{\partial \sigma}{\partial t} + \sigma^T \frac{\partial \sigma}{\partial \tilde{z}} B_0 d_v, \quad (48)$$

and in the sequence, by performing the necessary algebraic manipulations and using (46) and (47), one obtains the following:

$$\begin{aligned} \dot{V}_1 &= \sigma^T B_{0\sigma} (B_{0\sigma})^{-1} A_{0\sigma} + \sigma^T B_{0\sigma} (B_{0\sigma})^{-1} \frac{\partial \sigma}{\partial t} + \sigma^T B_{0\sigma} v_c + \sigma^T B_{0\sigma} d_v \\ &= \left( (B_{0\sigma})^T \sigma \right)^T (B_{0\sigma})^{-1} \left( A_{0\sigma} + \frac{\partial \sigma}{\partial t} \right) + \left( (B_{0\sigma})^T \sigma \right)^T (v_c + d_v) \\ &= \sigma^{*T} (B_{0\sigma})^{-1} \left( A_{0\sigma} + \frac{\partial \sigma}{\partial t} \right) + \sigma^{*T} (v_c + d_v). \end{aligned} \quad (49)$$

Selecting the control law  $v_c$ ,

$$v_c = -(B_{0_\sigma})^{-1} \left( A_{0_\sigma} + \frac{\partial \sigma}{\partial t} \right) - (Gsign(\sigma^*) + K_p \sigma^*). \quad (50)$$

By replacing (50) in (49),  $\dot{V}_1$  becomes:

$$\dot{V}_1 = -\sigma^{*T} (Gsign(\sigma^*) + K_p \sigma^*) + \sigma^{*T} d_v, \quad (51)$$

with  $\sigma^* = \left[ -k_1^2 \tilde{x} \quad k_1 \tilde{y} k_1 \tilde{x} - (k_2 \tilde{x} + k_3) (k_2 \tilde{y} + k_3 \tilde{\theta}) \right]^T$ . Equation (51) is similar to (43), and therefore, the same conclusions on the stability analysis are valid, considering (44) and (45). Moreover, the sliding mode occurs in the manifold  $\sigma^*(\tilde{z}, t) = 0$ . The transformation in (46) and (47) is nonsingular, and therefore, the manifolds  $\sigma(\tilde{z}, t) = 0$  and  $\sigma^*(\tilde{z}, t) = 0$  coincide, and the sliding mode takes place in the manifold  $\sigma(\tilde{z}, t) = 0$ , which was selected to design the sliding motion with the desired properties.

### 3.5. KNC

The design of the classical sliding mode controllers faces major disadvantages. First, because of control actions that are discontinuous across  $\sigma$  and  $\sigma^*$ , chattering occurs at a boundary of the surfaces  $\sigma$  and  $\sigma^*$ . This sort of high frequency switching (chattering) might excite unmodeled dynamics and impose undue wear and tear on the actuators, such that the control law would not be considered acceptable. Second, prior knowledge of the boundary of uncertainty is required in the compensators. If the boundary is unknown, a large value must be applied to the gain of the discontinuous components of the control signals, and this large control gain might intensify the chattering on the sliding surface. Thus, due to the delays, physical limitations of actuators, and imperfections of switching, it is not possible to switch the control from one value to another instantaneously. Because of this situation, the state trajectories vary in the vicinity around the sliding surface instead of sliding over it. This phenomenon, known as chattering, can be avoided or at least reduced with the use of RBFNNs (expressed similarly in (20)), which are nonlinear and continuous functions, to approximate  $Gsgn(\sigma^*)$  in (50) (Martins et al., 2012). Therefore,  $v_c$  becomes,

$$\begin{aligned} v_c &= -B_{0_\sigma}^{-1} \left( A_{0_\sigma} + \frac{\partial \sigma}{\partial t} \right) - \hat{P}_v(\sigma^*) - K_p \sigma^* \\ &= -B_{0_\sigma}^{-1} \left( A_{0_\sigma} + \frac{\partial \sigma}{\partial t} \right) - \left[ \left\{ \hat{W}_{\sigma^*} \right\}^T \bullet \left\{ \xi_{\sigma^*}(\sigma^*) \right\} \right] - K_p \sigma^*, \end{aligned} \quad (52)$$

where  $\left\{ \hat{W}_{\sigma^*} \right\}$ ,  $\left\{ \xi_{\sigma^*}(\sigma^*) \right\}$  are the Ge-Lee (GL) vectors (Ge, 1996) with their respective elements being  $\hat{W}_{\sigma_k^*}$  and  $\xi_{\sigma_k^*}(\sigma^*)$ , and  $\hat{P}_v(\sigma^*)$  is an  $n \times 1$  output vector of the RBFNNs.

Thus, for the stability analysis, one can choose the Lyapunov function candidate as follows:

$$V_1 = \frac{1}{2} \left( \sigma^T \sigma + \sum_{k=1}^n \tilde{W}_{\sigma_k^*}^T \Gamma_{\sigma_k^*}^{-1} \tilde{W}_{\sigma_k^*} \right), \quad (53)$$

where  $\Gamma_{\sigma_k^*}$  is a dimensionally compatible symmetric positive definite matrix, and  $\{\tilde{W}_{\sigma_k^*}\} = \{W_{\sigma_k^*}\} - \{\hat{W}_{\sigma_k^*}\}$ .

By differentiating (53), making the necessary mathematical manipulations, and replacing (52),  $\dot{V}_1$  is obtained as follows:

$$\dot{V}_1 = -\sigma^{*T} \left[ \{\hat{W}_{\sigma^*}\}^T \bullet \{\xi_{\sigma^*}(\sigma^*)\} \right] - \sigma^{*T} K_p \sigma^* + \sigma^{*T} d_v - \sum_{k=1}^n \tilde{W}_{\sigma_k^*}^T \Gamma_{\sigma_k^*}^{-1} \dot{\tilde{W}}_{\sigma_k^*}. \quad (54)$$

Recall that

$$\sigma^{*T} \left[ \{\tilde{W}_{\sigma^*}\}^T \bullet \{\xi_{\sigma^*}(\sigma^*)\} \right] = \sum_{k=1}^n \tilde{W}_{\sigma_k^*}^T \xi_{\sigma_k^*}(\sigma^*) \sigma_k^*. \quad (55)$$

Choosing the learning law of RBFNNs to be

$$\dot{\tilde{W}}_{\sigma_k^*} = \Gamma_{\sigma_k^*} \xi_{\sigma_k^*}(\sigma^*) \sigma_k^*, \quad (56)$$

and substituting (55) and (56) into (54),  $\dot{V}_1$  becomes as follows:

$$\dot{V}_1 \leq -K_{p_{\min}} |\sigma^*|^2 + \sigma^{*T} d_v - \sigma^{*T} \left[ \{W_{\sigma^*}\}^T \bullet \{\xi_{\sigma^*}(\sigma^*)\} \right], \quad (57)$$

where  $K_{p_{\min}}$  is the minimum singular value of  $K_p$ .

The expression for  $\dot{V}_1$  can be rewritten as follows:

$$\dot{V}_1 \leq -K_{p_{\min}} |\sigma^*|^2 + |\Delta f_v - P_v| |\sigma^*|, \quad (58)$$

with  $P_v = \left[ \{W_{\sigma^*}\}^T \bullet \{\xi_{\sigma^*}(\sigma^*)\} \right]$  being the optimal compensation for  $\Delta f_v = d_v$ . According to the property of universal approximation of RBFNNs (Li et al., 2004), there exists  $\mu > 0$  satisfying  $|\Delta f_v - P_v| \leq \mu$ , where  $\mu$  is arbitrary and can be chosen to be as small as possible. Assuming that  $\mu \leq \beta |\bar{\sigma}|$  with  $0 < \beta < 1$ , one obtains  $|\Delta f_v - P_v| |\sigma^*| \leq \beta |\sigma^*|^2 = \beta \sigma^{*2}$ ; therefore, the following inequality for  $\dot{V}_1$  results:

$$\dot{V}_1 \leq -(K_{p_{\min}} - \beta) \sigma^{*2}. \quad (59)$$

Because of  $K_{p_{\min}} > \beta$ ,  $\dot{V}_1$  is guaranteed to be negative definite.

### 3.6. TNC: DNC plus RNC

Given the desired control velocity  $v_c$  (52), one defines now the auxiliary velocity tracking error  $e_c$  as follows:

$$e_c = v_c - v = \begin{bmatrix} v_{c1} - v_l \\ v_{c2} - \omega_a \end{bmatrix}. \quad (60)$$

Let  $\Lambda$  be a symmetric diagonal positive definite matrix (Lewis et al., 2004), one introduces the following definitions:

$$\begin{aligned} v_r &= v_c + \Lambda_s \int_0^t e_c dt, & \dot{v}_r &= \dot{v}_c + \Lambda_s e_c, \\ s &= v_r - v = e_c + \Lambda_s \int_0^t e_c dt, & \dot{s} &= \dot{v}_r - \dot{v} = \dot{e}_c + \Lambda_s e_c, \end{aligned} \quad (61)$$

where  $v_r$  is the reference velocity vector,  $\dot{v}_r$  is the reference acceleration vector,  $s$  is the filtered tracking error vector, and  $\int_0^t e_c dt$  is an auxiliary position tracking error that does not reflect the position tracking error  $\tilde{z}$  directly (23) and does not have a physical meaning.

Let  $\{\hat{W}_{\bar{H}}\}$ , and  $\{\hat{W}_{\bar{C}}\}$  be the estimation of the true parameters of matrices  $\{W_{\bar{H}}\}$ , and  $\{W_{\bar{C}}\}$  of (2.4), and one defines the control input (TNC controller) as follows:

$$\begin{aligned} \bar{\tau} &= \hat{\tau} - \gamma_s = \hat{H}(q)\dot{v}_r + \hat{C}(q, \dot{q})v_r + (K_s + I_n)s - \gamma_s \\ &= \underbrace{\left[ \left\{ \hat{W}_{\bar{H}} \right\}^T \bullet \{ \xi_{\bar{H}}(q) \} \right] \dot{v}_r + \left[ \left\{ \hat{W}_{\bar{C}} \right\}^T \bullet \{ \xi_{\bar{C}}(z) \} \right] v_r}_{DNC} + (K_s + I_n)s - \gamma_s, \end{aligned} \quad (62)$$

where  $(K_s + I_n)^T = (K_s + I_n) > 0$ ,  $I_n$  is the identity matrix, and  $\gamma_s$  is the constant plus proportional rate reaching law with the aim of compensating the bounded unknown disturbances (Utkin et al., 2009), which is defined as follows:

$$\gamma_s = -G_s \operatorname{sgn}(s), \quad (63)$$

with  $G_s^T = G_s > 0$ .

It is necessary to emphasize that in (63), the discontinuous control signal will cause a significant chattering problem, which will excite the high-frequency dynamics of the nonlinear system. Because this outcome is highly undesirable, to eliminate or minimize the chattering, a smooth approximation (proper continuous function, shifted sigmoid function, and hyperbolic tangent function) is recommended instead of the sign function. Then,

$$\operatorname{sgn}(s) = s/(|s| + \delta),$$

$$\begin{aligned} \operatorname{sgn}(s) &= (1 - e^{-\delta s}) / (1 + e^{-\delta s}), \\ \operatorname{sgn}(s) &= \tanh(\delta s), \end{aligned} \quad (64)$$

where  $\delta$  is a positive parameter.

The present control design proposes an RBFNN expressed in a manner similar to (20) as a continuous approximation of  $G_s \operatorname{sgn}(s)$  in  $\gamma_s$  (63). Then,

$$\gamma_s = -\hat{P}_s(s) = - \underbrace{\left[ \left\{ \hat{W}_{P_s} \right\}^T \bullet \left\{ \xi_{P_s}(s) \right\} \right]}_{RNC}, \quad (65)$$

where  $\hat{P}_s(s)$  is an  $n \times 1$  vector, in which  $\hat{p}_{s_k}$  is the output of the  $k$ -th RBFNN.

Substituting  $v = v_r - s$ ,  $\dot{v} = \dot{v}_r - \dot{s}$  of (61), (62), and (65) into the dynamics (3), one obtains the closed-loop system error dynamics:

$$\begin{aligned} \bar{H}(q)\dot{s} &= \left[ \left\{ \tilde{W}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(\cdot) \right\} \right] \dot{v}_r + \left[ \left\{ \tilde{W}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(\cdot) \right\} \right] v_r + \bar{\tau}_{cp} \\ &\quad + E - \bar{C}(q, \dot{q})s - (K_s + I_n)s - \left[ \left\{ \hat{W}_{P_s} \right\}^T \bullet \left\{ \xi_{P_s}(s) \right\} \right], \end{aligned} \quad (66)$$

where  $\left\{ \tilde{W} \right\} = \left\{ W \right\} - \left\{ \hat{W} \right\}$  are parameter errors, and  $E = E_{\bar{H}}(q) \dot{v}_r + E_{\bar{C}}(z) v_r$  defines the vector of neural network modeling errors.

For the TNC, (62) and (65), the learning algorithms for the neural networks must be developed such that the control system will be stable, and both the velocity and position tracking errors converge to zero.

Let us consider the following Lyapunov function candidate:

$$\begin{aligned} V &= V_1 + V_2 \\ V_2 &= \frac{1}{2} \left( s^T \bar{H}(q)s + \sum_{k=1}^n \tilde{W}_{\bar{H}_k}^T \Gamma_{\bar{H}_k}^{-1} \tilde{W}_{\bar{H}_k} + \sum_{k=1}^n \tilde{W}_{\bar{C}_k}^T \Gamma_{\bar{C}_k}^{-1} \tilde{W}_{\bar{C}_k} \right. \\ &\quad \left. + \sum_{k=1}^n \tilde{W}_{P_{s_k}}^T \Gamma_{P_{s_k}}^{-1} \tilde{W}_{P_{s_k}} \right) + \left( \int_0^t e_c dt \right)^T \Lambda_s \int_0^t e_c dt, \end{aligned} \quad (67)$$

where  $V_1$  corresponds to the function dependent on the KNC, as described in Subsection 3.5, and designed for the posture kinematic model (1), which assumes that  $V_1 > 0$  and  $\dot{V}_1 \leq 0, \forall t > 0$ . In (67),  $\Gamma_{\bar{H}_k}$ ,  $\Gamma_{\bar{C}_k}$ , and  $\Gamma_{P_{s_k}}$  are dimensionally compatible symmetric positive definite matrices, and clearly,  $V_2 > 0$  if and only if  $e_c \neq 0$ ,  $\int_0^t e_c dt \neq 0$ ,  $s \neq 0$ ,  $\left\{ \tilde{W}_{\bar{H}} \right\} \neq 0$ ,  $\left\{ \tilde{W}_{\bar{C}} \right\} \neq 0$ , and  $\left\{ \tilde{W}_{P_s} \right\} \neq 0$ .

By differentiating  $V_2$  of (67), and substituting the error dynamics (66),  $\dot{V}_2$  is obtained as follows:

$$\begin{aligned} \dot{V}_2 &= s^T \left[ \left\{ \tilde{W}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(q) \right\} \right] \dot{v}_r + s^T \left[ \left\{ \tilde{W}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(z) \right\} \right] v_r \\ &\quad + s^T \bar{\tau}_{cp} + s^T E - s^T \left[ \left\{ \hat{W}_{P_s} \right\}^T \bullet \left\{ \xi_{P_s}(s) \right\} \right] - s^T (K_s + I_n) s \\ &\quad - \sum_{k=1}^n \tilde{W}_{\bar{H}_k}^T \Gamma_{\bar{H}_k}^{-1} \dot{\tilde{W}}_{\bar{H}_k} - \sum_{k=1}^n \tilde{W}_{\bar{C}_k}^T \Gamma_{\bar{C}_k}^{-1} \dot{\tilde{W}}_{\bar{C}_k} - \sum_{k=1}^n \tilde{W}_{P_{s_k}}^T \Gamma_{P_{s_k}}^{-1} \dot{\tilde{W}}_{P_{s_k}} \\ &\quad + 2e_c^T \Lambda_s \int_0^t e_c dt, \end{aligned} \quad (68)$$

where the skew-symmetry property of  $(\dot{\hat{H}} - 2\bar{C})$  has been used.

Recall that

$$\begin{aligned} s^T \left[ \left\{ \tilde{W}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(q) \right\} \right] \dot{v}_r &= \sum_{k=1}^n \left\{ \tilde{W}_{\bar{H}_k} \right\}^T \bullet \left\{ \xi_{\bar{H}_k}(q) \right\} \dot{v}_r s_k, \\ s^T \left[ \left\{ \tilde{W}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(z) \right\} \right] v_r &= \sum_{k=1}^n \left\{ \tilde{W}_{\bar{C}_k} \right\}^T \bullet \left\{ \xi_{\bar{C}_k}(z) \right\} v_r s_k, \\ s^T \left[ \left\{ \tilde{W}_P \right\}^T \bullet \left\{ \xi_P(s) \right\} \right] &= \sum_{k=1}^n \tilde{W}_{P_{s_k}}^T \xi_{P_{s_k}}(s) s_k, \end{aligned} \quad (69)$$

and choose the weight learning laws of the neural networks to be

$$\begin{aligned} \dot{\hat{W}}_{\bar{H}_k} &= \Gamma_{\bar{H}_k} \bullet \left\{ \xi_{\bar{H}_k}(q) \right\} \dot{v}_r s_k - K_{\bar{H}_k} \Gamma_{\bar{H}_k} \|s\| \hat{W}_{\bar{H}_k}, \\ \dot{\hat{W}}_{\bar{C}_k} &= \Gamma_{\bar{C}_k} \bullet \left\{ \xi_{\bar{C}_k}(z) \right\} v_r s_k - K_{\bar{C}_k} \Gamma_{\bar{C}_k} \|s\| \hat{W}_{\bar{C}_k}, \\ \dot{\hat{W}}_{P_k} &= \Gamma_{P_{s_k}} \xi_{P_{s_k}}(s) s_k - K_{P_{s_k}} \Gamma_{P_{s_k}} \|s\| \hat{W}_{P_{s_k}}, \end{aligned} \quad (70)$$

where  $K_{\cdot k} = K_{\cdot} > 0$  are positive constants. The terms  $K_{\cdot k} \Gamma_{\cdot k} \|s\| \hat{W}_{\cdot k}$  of (70) correspond to  $\varepsilon$  modification (Lewis et al., 2004) from the adaptive control theory. Therefore, they must be added to eliminate the condition of persistent excitation and to ensure the bounded neural network weight estimates.

Substituting (69), and (70) into (68),  $\dot{V}_2$  results in the following:

$$\begin{aligned} \dot{V}_2 &\leq -K_{s_{min}} \|s\|^2 - s^T I_n s + \|\bar{\tau}_{cp}\| \|s\| + \|E\| \|s\| - s^T \left[ \{W_{P_s}\}^T \bullet \left\{ \xi_{P_s}(s) \right\} \right] \\ &\quad + 2e_c^T \Lambda_s \int_0^t e_c dt + k_{\bar{H}} \|s\| \sum_{k=1}^n \tilde{W}_{\bar{H}_k}^T \hat{W}_{\bar{H}_k} + k_{\bar{C}} \|s\| \sum_{k=1}^n \tilde{W}_{\bar{C}_k}^T \hat{W}_{\bar{C}_k} \\ &\quad + k_{P_s} \|s\| \sum_{k=1}^n \tilde{W}_{P_{s_k}}^T \hat{W}_{P_{s_k}}, \end{aligned} \quad (71)$$

where  $K_{s_{min}}$  are the minimum singular values of  $K_s$ , and  $K_{\cdot k} = k_{\cdot} > 0$  are the positive constants.

Observing that  $tr \left( \tilde{W}_{\cdot}^T \hat{W}_{\cdot} \right) = \sum_{k=1}^n \tilde{W}_{\cdot k}^T \hat{W}_{\cdot k}$ , with  $tr(\cdot)$  being trace function, and assuming that the unmodeled and unstructured disturbances are bounded as well as the neural networks modeling errors such that  $\|\bar{\tau}_{cp}\| \leq b_{cp}$  and  $\|E\| \leq \varepsilon_{NN}$ , one obtains  $\dot{V}_2$  as follows:

$$\begin{aligned} \dot{V}_2 &\leq -K_{s_{min}} \|s\|^2 - \|e_c\|^2 + b_{cp} \|s\| + \varepsilon_{NN} \|s\| - s^T \left[ \{W_{P_s}\}^T \bullet \left\{ \xi_{P_s}(s) \right\} \right] \\ &\quad - \left( \int_0^t e_c dt \right)^T \Lambda_s^T \Lambda_s \int_0^t e_c dt + k_{\bar{H}} \|s\| tr \left( \tilde{W}_{\bar{H}}^T \hat{W}_{\bar{H}} \right) \\ &\quad + k_{\bar{C}} \|s\| tr \left( \tilde{W}_{\bar{C}}^T \hat{W}_{\bar{C}} \right) + k_{P_s} \|s\| tr \left( \tilde{W}_{P_s}^T \hat{W}_{P_s} \right). \end{aligned} \quad (72)$$

Using the Schwartz inequality (Li et al., 2004), the trace function can be written as follows:

- For matrices:

$$\begin{aligned} tr\left(\tilde{W}_{\bar{H}}^T \hat{W}_{\bar{H}}\right) &= tr\left(\tilde{W}_{\bar{H}}^T (W_{\bar{H}} - \tilde{W}_{\bar{H}})\right) = \left\langle \tilde{W}_{\bar{H}}, W_{\bar{H}} \right\rangle - \left\| \tilde{W}_{\bar{H}} \right\|_F^2 \\ &\leq \left\| \tilde{W}_{\bar{H}} \right\|_F \|W_{\bar{H}}\|_F - \left\| \tilde{W}_{\bar{H}} \right\|_F^2 \leq \left\| \tilde{W}_{\bar{H}} \right\|_F w_{\bar{H}_{max}} - \left\| \tilde{W}_{\bar{H}} \right\|_F^2 \\ tr\left(\tilde{W}_{\bar{C}}^T \hat{W}_{\bar{C}}\right) &= tr\left(\tilde{W}_{\bar{C}}^T (W_{\bar{C}} - \tilde{W}_{\bar{C}})\right) = \left\langle \tilde{W}_{\bar{C}}, W_{\bar{C}} \right\rangle - \left\| \tilde{W}_{\bar{C}} \right\|_F^2 \\ &\leq \left\| \tilde{W}_{\bar{C}} \right\|_F \|W_{\bar{C}}\|_F - \left\| \tilde{W}_{\bar{C}} \right\|_F^2 \leq \left\| \tilde{W}_{\bar{C}} \right\|_F w_{\bar{C}_{max}} - \left\| \tilde{W}_{\bar{C}} \right\|_F^2; \end{aligned} \quad (73)$$

- For vectors:

$$\begin{aligned} tr\left(\tilde{W}_{P_s}^T \hat{W}_{P_s}\right) &= tr\left(\tilde{W}_{P_s}^T (W_{P_s} - \tilde{W}_{P_s})\right) = \left\langle \tilde{W}_{P_s}, W_{P_s} \right\rangle - \left\| \tilde{W}_{P_s} \right\|^2 \\ &\leq \left\| \tilde{W}_{P_s} \right\| \|W_{P_s}\| - \left\| \tilde{W}_{P_s} \right\|^2 \leq \left\| \tilde{W}_{P_s} \right\| w_{P_{s_{max}}} - \left\| \tilde{W}_{P_s} \right\|^2; \end{aligned} \quad (74)$$

where  $w_{max}$  are positive constants.

Substituting (73), and (74) into (72), leads to the following expression regarding  $\dot{V}_2$ :

$$\begin{aligned} \dot{V}_2 &\leq -K_{s_{min}} \|s\|^2 - \|e_c\|^2 + (b_{cp} + \varepsilon_{NN}) \|s\| - s^T \left[ \{W_{P_s}\}^T \bullet \{\xi_{P_s}(s)\} \right] \\ &\quad - \left( \int_0^t e_c dt \right)^T \eta_s \int_0^t e_c dt + k_{\bar{H}} \|s\| \left( \left\| \tilde{W}_{\bar{H}} \right\|_F w_{\bar{H}_{max}} - \left\| \tilde{W}_{\bar{H}} \right\|_F^2 \right) \\ &\quad + k_{\bar{C}} \|s\| \left( \left\| \tilde{W}_{\bar{C}} \right\|_F w_{\bar{C}_{max}} - \left\| \tilde{W}_{\bar{C}} \right\|_F^2 \right) + k_{P_s} \|s\| \left( \left\| \tilde{W}_{P_s} \right\| w_{P_{s_{max}}} - \left\| \tilde{W}_{P_s} \right\|^2 \right), \end{aligned} \quad (75)$$

or

$$\begin{aligned} \dot{V}_2 &\leq -K_{s_{min}} \|s\|^2 - \|e_c\|^2 + |\Delta f_s - P_s| \|s\| - \eta_{s_{min}} \left\| \int_0^t e_c dt \right\|^2 \\ &\quad + k_{\bar{H}} \|s\| \left( \left\| \tilde{W}_{\bar{H}} \right\|_F w_{\bar{H}_{max}} - \left\| \tilde{W}_{\bar{H}} \right\|_F^2 \right) \\ &\quad + k_{\bar{C}} \|s\| \left( \left\| \tilde{W}_{\bar{C}} \right\|_F w_{\bar{C}_{max}} - \left\| \tilde{W}_{\bar{C}} \right\|_F^2 \right) \\ &\quad + k_{P_s} \|s\| \left( \left\| \tilde{W}_{P_s} \right\| w_{P_{s_{max}}} - \left\| \tilde{W}_{P_s} \right\|^2 \right), \end{aligned} \quad (76)$$

where  $\eta_s = \Lambda_s^T \Lambda_s$ , with  $\eta_{s_{min}}$  being the minimum singular values of  $\eta_s$ , and  $P_s = \left[ \{W_{P_s}\}^T \bullet \{\xi_{P_s}(s)\} \right]$  being the optimal compensation for  $\Delta f_s = b_{cp} + \varepsilon_{NN}$ .

According to the property of universal approximation of RBFNNs (Li et al., 2004), there exists  $\mu_s > 0$  satisfying  $|\Delta f_s - P_s| \leq \mu_s$ , where  $\mu_s$  is arbitrary and can be chosen as small as possible. Therefore, by making the necessary mathematical manipulations, one obtains for  $\dot{V}_2$  the following expression:

$$\begin{aligned} \dot{V}_2 \leq & -\|e_c\|^2 - \eta_{s_{\min}} \left\| \int_0^t e_c dt \right\|^2 - \|s\| (K_{s_{\min}} \|s\| - \mu_s \\ & + k_{\bar{H}} \left( \left\| \tilde{W}_{\bar{H}} \right\|_F - \frac{w_{\bar{H}_{\max}}}{2} \right)^2 - k_{\bar{H}} \frac{w_{\bar{H}_{\max}}^2}{4} \\ & + k_{\bar{C}} \left( \left\| \tilde{W}_{\bar{C}} \right\|_F - \frac{w_{\bar{C}_{\max}}}{2} \right)^2 - k_{\bar{C}} \frac{w_{\bar{C}_{\max}}^2}{4} \\ & + k_{P_s} \left( \left\| \tilde{W}_{P_s} \right\| - \frac{w_{P_s_{\max}}}{2} \right)^2 - k_{P_s} \frac{w_{P_s_{\max}}^2}{4} \Big). \end{aligned} \quad (77)$$

Thus,  $\dot{V}_2$  is guaranteed negative as long as the term in parenthesis in (77) is positive, and either

$$\|s\| > \frac{\mu_s + k_{\bar{H}} \frac{w_{\bar{H}_{\max}}^2}{4} + k_{\bar{C}} \frac{w_{\bar{C}_{\max}}^2}{4} + k_{P_s} \frac{w_{P_s_{\max}}^2}{4}}{K_{s_{\min}}}, \quad (78)$$

or

$$\begin{aligned} \left\| \tilde{W}_{\bar{H}} \right\|_F & > \frac{w_{\bar{H}_{\max}}}{2} + \sqrt{\frac{w_{\bar{H}_{\max}}^2}{4} + \frac{\mu_s + k_{\bar{C}} \frac{w_{\bar{C}_{\max}}^2}{4} + k_{P_s} \frac{w_{P_s_{\max}}^2}{4}}{k_{\bar{H}}}} \\ \left\| \tilde{W}_{\bar{C}} \right\|_F & > \frac{w_{\bar{C}_{\max}}}{2} + \sqrt{\frac{w_{\bar{C}_{\max}}^2}{4} + \frac{\mu_s + k_{\bar{H}} \frac{w_{\bar{H}_{\max}}^2}{4} + k_{P_s} \frac{w_{P_s_{\max}}^2}{4}}{k_{\bar{C}}}} \\ \left\| \tilde{W}_{P_s} \right\| & > \frac{w_{P_s_{\max}}}{2} + \sqrt{\frac{w_{P_s_{\max}}^2}{4} + \frac{\mu_s + k_{\bar{H}} \frac{w_{\bar{H}_{\max}}^2}{4} + k_{\bar{C}} \frac{w_{\bar{C}_{\max}}^2}{4}}{k_{P_s}}}. \end{aligned} \quad (79)$$

Therefore,  $\dot{V}_2$  is negative definite within a particular compact set and is negative semidefinite outside this set, as defined by (78) and (79). The stability of the global system is ensured because  $\dot{V}_1$  and  $\dot{V}_2$  are guaranteed to be negative definite; thus,  $\dot{V}$  is also guaranteed negative definite. According to a standard Lyapunov theory and LaSalle's Theorem (Lewis et al., 2004), all signals of  $\|\tilde{z}\|$ ,  $\|e_c\|$ ,  $\left\| \int_0^t e_c dt \right\|$ ,  $\|s\|$ ,  $\left\| \tilde{W}_{\bar{H}} \right\|_F$ , and  $\left\| \tilde{W}_{\bar{C}} \right\|_F$  and  $\left\| \tilde{W}_{P_s} \right\|$  are uniformly ultimate bounded (u.u.b.).

A representation of the proposed control system (PANC: KNC plus TNC) is shown in the block diagram of Fig. 6, which consists of two control loops:

- Kinematics  $\rightarrow$  Uses the KNC controller in rectangular coordinates (KNC block) to control the wheeled mobile robot kinematics (WMR Kinematics block) to achieve posture tracking;
- Dynamics  $\rightarrow$  Uses the TNC controller to control the wheeled mobile robot dynamics (WMR Dynamics block) to achieve velocity tracking.

The function of the PANC controller is to perform a mapping between the known information pieces (e.g., information on the reference position, reference velocity and sensors) and the actuator commands, designed for a wheeled mobile

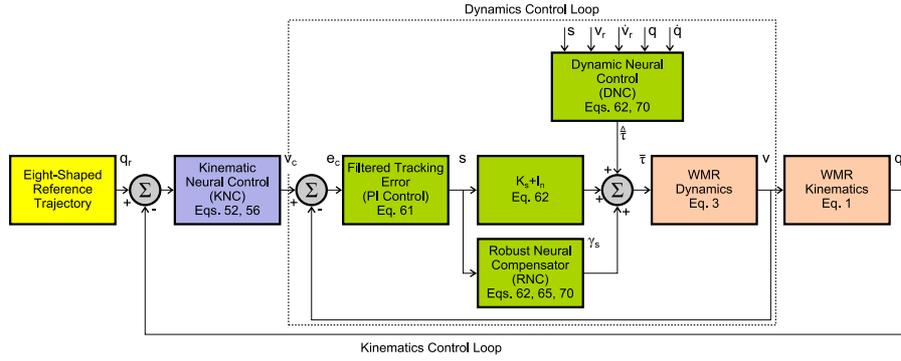


Figure 6. Block diagram of the proposed control structure – the PANC controller

robot that performs a given task. Thus, the problem of control design for wheeled mobile robots can be described as follows: given the reference positions  $q_r(t)$  and the reference velocities  $\dot{q}_r(t)$ , design a control law to generate the torques of the actuators  $\bar{\tau}$ , which make the wheeled mobile robot move with velocity  $v$  to perform the action of a smooth control velocity input  $v_c$  ( $v \rightarrow v_c$  when  $t \rightarrow \infty$ ), and as a result,  $\lim_{t \rightarrow \infty} (q_r - q) = 0$  through the use of the KNC controller. In short, the TNC controller contains the function that corrects the auxiliary velocity tracking error  $e_c$ , whereas the KNC controller aims to correct only the posture tracking error ( $q_r - q$ ).

#### 4. Simulation results

The simulations were carried out in the MATLAB/Simulink software using Euler's method with an integration size of 0.001 s.

In the simulations, the dynamic model was made use of, described in Souza Junior et al. (2002), whose parameters of the Magellan PRO ISR wheeled mobile robot (Silveira Junior and Hemerly, 2004) are:  $m_c = 22.9644$  kg, and  $I_c = 0.4732$  kgm<sup>2</sup>.

Beyond the disturbances in the kinematic model,

$$d_v = \begin{bmatrix} \delta_{v_l} \\ \delta_{\omega_a} \end{bmatrix} = \begin{bmatrix} 0.5 + 0.1 \sin(0.01t) \\ 0.8 + 0.1 \cos(0.01t) \end{bmatrix}, \quad (80)$$

a Coulomb friction vector is added to the wheeled mobile robot dynamics as a disturbance (Hu et al., 2002), i.e.,

$$\tau_p = \begin{bmatrix} f_{s_1} \operatorname{sgn}(v_l) \\ f_{s_2} \operatorname{sgn}(\omega_a) \end{bmatrix}, \quad (81)$$

where  $f_{s_1} = f_{s_2} = 1.5$  N.

Considering that the guidance point of the wheeled mobile robot is null ( $d = 0$ ) for the wheeled mobile robot dynamics (3), it reduces to

$$\bar{H}(q)\dot{v} + \bar{\tau}_{cp} = \bar{\tau}, \quad (82)$$

and because of the matrix  $\bar{C}(q, \dot{q})$ , the result is

$$\bar{C}(q, \dot{q}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (83)$$

and the disturbance vector  $\bar{\tau}_{cp}$  becomes as follows:

$$\bar{\tau}_{cp} = \bar{H}(q)\dot{d}_v + \bar{\tau}_p. \quad (84)$$

For the trajectory tracking control problem, an eight-shaped trajectory is considered and is given by  $x_r = \sin(t/10)$ ,  $y_r = \sin(t/20)$  and  $\theta_r = \text{Atan2}(\dot{y}_r, \dot{x}_r) + k\pi$  with  $k = 0, 1$  (Oriolo et al., 2002). The reference angular and linear velocities are given by  $v_{l_r} = \sqrt{\dot{x}_r^2 + \dot{y}_r^2}$  and  $\omega_{a_r} = \frac{\ddot{y}_r\dot{x}_r - \ddot{x}_r\dot{y}_r}{\dot{x}_r^2 + \dot{y}_r^2}$ . The trajectory initiates with  $[x_r, y_r, \theta_r]^T = [0, 0, \pi/6]^T$  and the initial posture of the wheeled mobile robot is taken as  $[x_c, y_c, \theta]^T = [0.2, -0.3, \pi/3]^T$ . Additionally, the reference initial velocities are  $v_{l_r}(0) = 0.1118$  m/s and  $\omega_{a_r}(0) = 0.0$  rad/s, and a full cycle is completed in approximately 125.0 s. This trajectory is more general, because its curvature is not constant; thus, the angular velocity will not display a steady value throughout the trajectory tracking; this situation may result in possible position tracking errors oscillating near zero during the movement of the wheeled mobile robot, as shown later in the simulations.

In the simulations, integration of the kinematic controller is considered in rectangular coordinates (KNC) with a torque controller. This torque controller can be of the type of computed-torque (Oh et al., the 2004), neural network controller (Hu and Yang, 2001), or the proposed TNC, in accordance with the wheeled mobile robot dynamics (82) (Martins et al., 2012), which are related by convenience.

The computed-torque controller (CTC) considers the precise knowledge of all parameters of the wheeled mobile robot dynamics (82) and ignores disturbances of any nature, i.e.,  $\bar{\tau}_{cp} = 0$ . Thus, the law of control of the type computed-torque is given as follows:

$$\bar{\tau} = \bar{H}(q)u_r, \quad (85)$$

with

$$u_r = \dot{v}_c + K_r e_c, \quad (86)$$

where  $e_c$  is the auxiliary velocity tracking error,  $K_r^T = K_r > 0$ , and  $\dot{v}_c$  is the time derivative of the desired control velocity  $v_c$  (60). Therefore, substituting (85) and (86) into (82) produces the following:

$$\bar{H}(q)(\dot{e}_c + K_r e_c) = 0, \quad (87)$$

which implies that  $v$  converges to  $v_c$  with exponential rate  $-K_r$ . In Fierro and Lewis (1995), the stability of the control law (85) is established locally. It is important to note that the test requires that the parameters of the system be determined with precision and that the system be free of disturbances.

The neural network controller (NNC) (Yu and Yang, 2001) considers the completely unknown wheeled mobile robot dynamics (82) under unmodeled disturbances (81).

Thus, the law of the neural network control is given as follows:

$$\bar{\tau} = \bar{H}(q) \dot{v}_c + \bar{K}_v e_c = \bar{Y}(\dot{v}_{c_c}) \bar{\varphi} + \bar{K}_v e_c = \bar{Y}(\dot{v}_{c_c}) \hat{W} + \bar{K}_v e_c, \quad (88)$$

where  $e_c$  is the auxiliary velocity tracking error,  $\bar{K}_v$  is a diagonal positive defined design matrix, as well as a robustness term to compensate for the unmodeled disturbances,  $\hat{W}$  is a vector that represents the connection weights of the neural network and is also the approximation of  $\bar{\varphi}$ ,  $\dot{v}_c$  is the time derivative of the desired control velocity  $v_c$  (60),  $\bar{Y}(\dot{v}_{c_c})$  is the wheeled mobile robot regressor (i.e., a coefficient matrix consisting of the known functions of wheeled mobile robot, or, in this case, the control acceleration  $\dot{v}_{c_c}$ ), and  $\bar{\varphi}$  is a vector consisting of the known and unknown wheeled mobile robot dynamics (in this case, mass  $m$  and moment of inertia  $I$ ). Therefore,  $\bar{Y}(\dot{v}_{c_c})$  and  $\bar{\varphi}$  are defined as follows:

$$\begin{aligned} \bar{Y}(\dot{v}_{c_c}) &= \begin{bmatrix} \dot{v}_{c_1} & 0 \\ 0 & \dot{v}_{c_2} \end{bmatrix} \\ \bar{\varphi} &= \begin{bmatrix} m & I \end{bmatrix}. \end{aligned} \quad (89)$$

In Yu and Yang (2001), the vector of estimation error of  $\bar{\varphi}$  is given as  $\tilde{W} = \bar{\varphi} - \hat{W} = \bar{W} - \hat{W}$ , and  $\dot{\tilde{W}} = -\dot{\hat{W}}$ . Therefore, the learning law for the neural network is obtained as follows:

$$\dot{\tilde{W}} = -\bar{\Gamma} \bar{Y}^T e_c, \quad (90)$$

where  $\bar{\Gamma}$  is a positive constant design matrix, and the weights  $\hat{W}$  of the neural network are initialized to zero without any prior knowledge of the system uncertainties and disturbances. The stability of the proposed control system and the convergence of tracking errors to zero are rigorously proved using Lyapunov theory (Yu and Yang, 2001).

In the simulations, the gains of the controllers are empirically chosen to obtain acceptable tracking errors and control efforts, thus preventing damage to the actuators. To view the influence of the disturbances on the dynamic model of the wheeled mobile robot, the same gains as in the nominal case (free of disturbances) are considered for the CTC, NNC and TNC controllers, as well as for the KNC controller. It should be noted that for the KNC and TNC controllers, the centers of the localized Gaussian radial basis functions are evenly distributed to span the input space of the neural networks (Passold, 2009). The weights of the RBFNNs were initialized to zero without any prior knowledge of

the system uncertainties and disturbances. It is important to emphasize that different tracking performances can be achieved by adjusting the parameter gains and other factors, i.e., the size of the RBFNNs, centers, and variances of the Gaussian radial basis functions.

The parameters of the controllers are as follows:

- KNC  $\rightarrow k_1 = 1, k_2 = 1, k_3 = 0.25, K_{p_{11}} = 7.5, K_{p_{22}} = 15, \Gamma_{\sigma_k^*} = 0.37, \sigma_{\sigma_k^*} = \sqrt{1.5}$ , and the number of hidden neurons = 25;
- CTC  $\rightarrow K_r = 5$ ;
- NNC  $\rightarrow \bar{\Gamma}_{11} = 120, \bar{\Gamma}_{22} = 25, \bar{K}_{v_{11}} = 45, \bar{K}_{v_{22}} = 25$ ;
- TNC  $\rightarrow \Lambda_{s_{11}} = 0.5, \Lambda_{s_{22}} = 0.5, K_{s_{11}} = 20, K_{s_{22}} = 12, \Gamma_{\bar{H}_{11}} = 3.75, \Gamma_{\bar{H}_{22}} = 10, \Gamma_{P_{s_k}} = \text{diag}\{0.5, 0.5\}, k_{\bar{H}_k} = 0.01, k_{p_{s_k}} = 0.01, \sigma_{\bar{H}_{kj}} = 4, \sigma_{p_{s_k}} = \sqrt{1.5}$ , and the number of hidden neurons = 25.

The simulation results are obtained by taking into account the kinematic and dynamic models with disturbances and without disturbances (free of disturbances) and the following analysis, considering the respective control strategies:

- Control 1: Integration of the KNC controller (with or without the neural term) (52) or (56) with the CTC controller;
- Control 2: Integration of the KNC controller (with or without the neural term) (52) or (56) with the NNC controller;
- Control 3: Integration of the KNC controller (with or without the neural term) (52) or (56) with the TNC controller.

#### 4.1. Nominal case: use of KNC controller with or without the neural term

In this case, the kinematic and dynamic models are free of disturbances, i.e., there are no disturbances.

For the results of simulations, Figs. 7 and 8 show that with the use of Controls 1, 2, and 3, the RMR satisfactorily follows the desired trajectory.

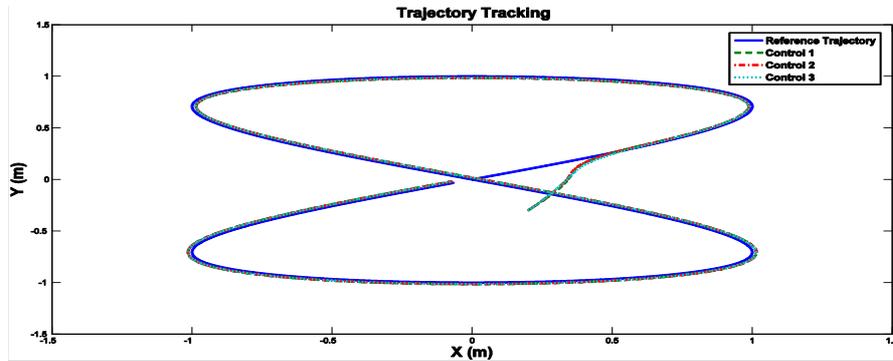


Figure 7. Trajectory tracking without the neural term of the KNC controller

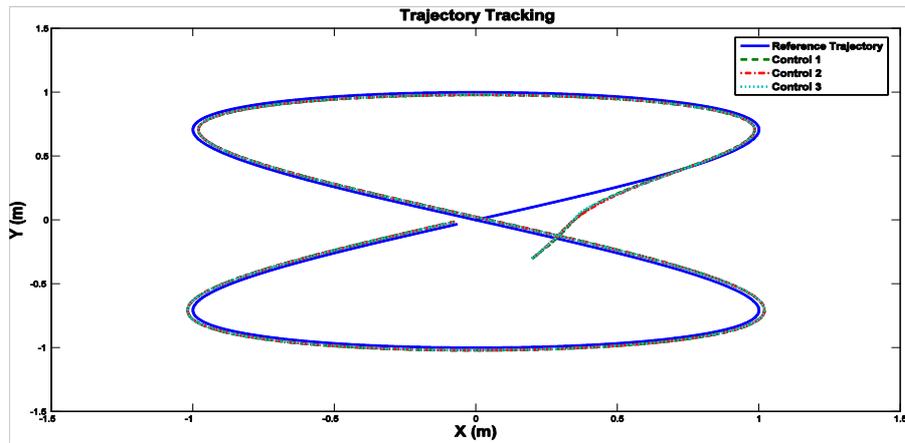
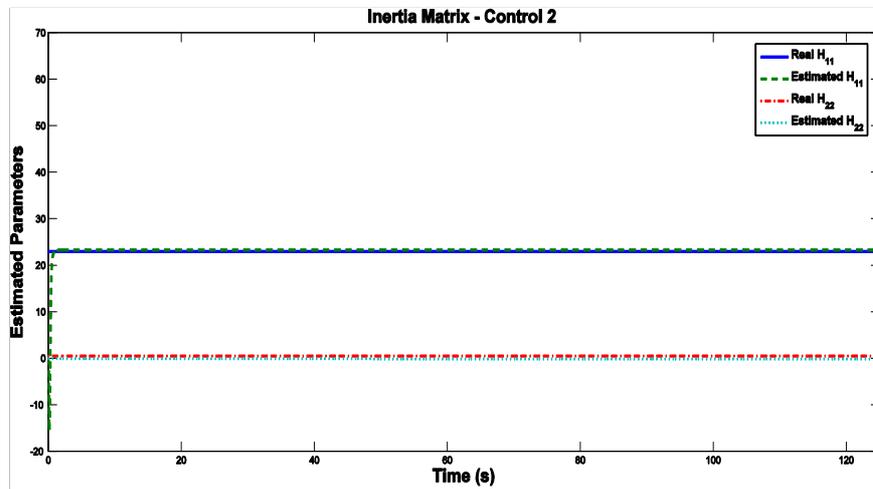


Figure 8. Trajectory tracking without the neural term of the KNC controller

Figure 9. Estimated parameters for the inertia matrix  $\bar{H}(q)$  by NNC controller

Then, Figs. 9, 10, 11 and 12 verify the estimation of parameters of the inertia matrix  $\bar{H}(q)$  for Controls 2 and 3. Considering a suitable choice of gains for NNC controller of Control 2, without the neural term of the KNC controller (Fig. 9), the estimated parameters tend to the true parameters of the inertia matrix ( $\bar{H}_{11} = 22.9644$ , and  $\bar{H}_{22} = 0.4732$ ), whereas with the neural term of the KNC controller (Fig. 11), the estimated parameter  $\hat{H}_{11}$  is located significantly away from its true parameter, and the estimated parameter  $\hat{H}_{22}$  is closer to its true parameter. By observation, for TNC controller of Control 3, the estimated parameters tend to stay close to the true parameters of the inertia matrix  $\bar{H}(q)$ .

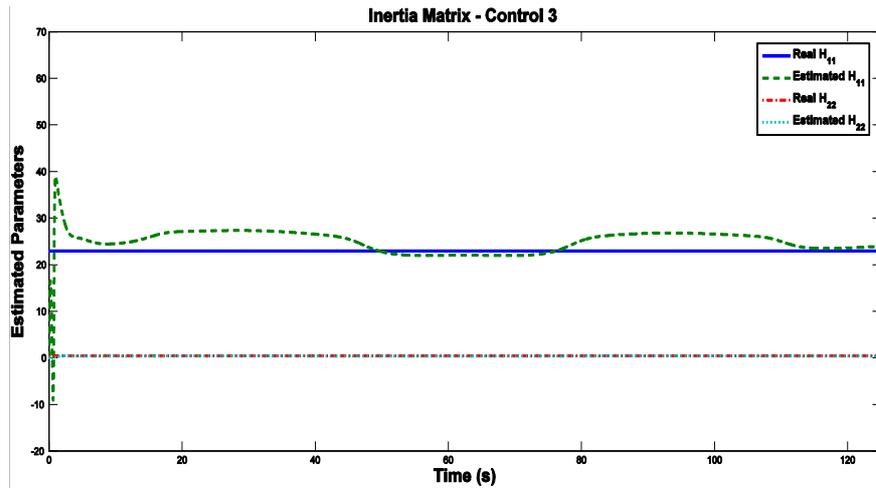


Figure 10. Estimated parameters for the inertia matrix  $\bar{H}(q)$  by TNC controller using the KNC controller without the neural term

This result is obtained with a suitable choice of gains, with or without the neural term of the KNC controller (Figs. 10 and 12).

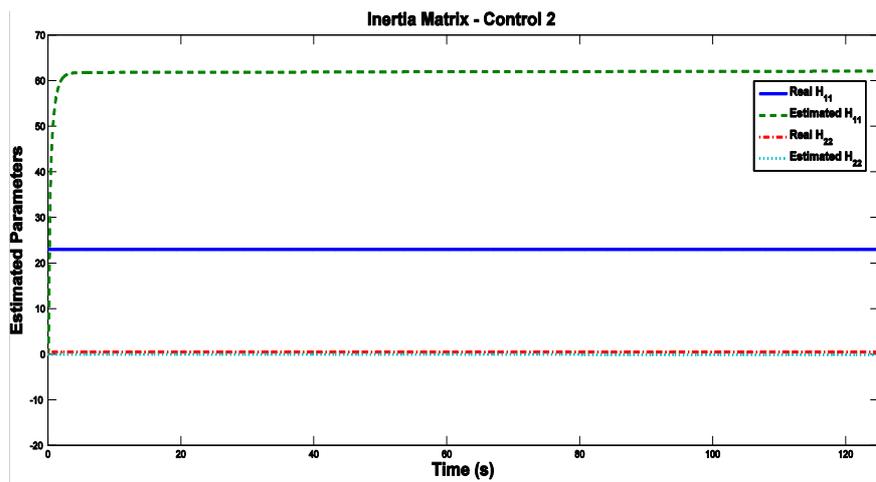


Figure 11. Estimated parameters for the inertia matrix  $\bar{H}(q)$  by NNC controller

Moreover, after simulations for this case with or without the neural term of the KNC controller, the results obtained were the following:

- The posture tracking errors ( $e_x$ ,  $e_y$ ,  $e_\theta$ ) tend to converge to zero with the use of Controls 1, 2, and 3;

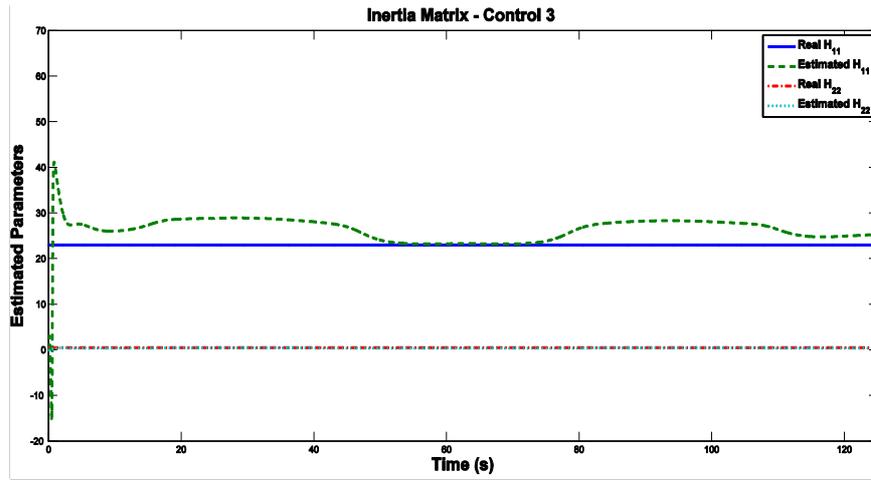


Figure 12. Estimated parameters for the inertia matrix  $\bar{H}(q)$  by TNC controller using the KNC controller with the neural term

- The linear and angular velocities  $(v(t) = v_c(t) = [v_l \ \omega_a]^T)$  do not display chattering and these velocities tend to the desired values  $(v_r = [v_{l_r} \ \omega_{a_r}]^T)$ , which represent the kinematic control signals (KNC controller) for the use of Controls 1, 2, and 3;
- The control torques  $\tau$  (dynamic control signals) using Controls 1, 2, and 3 also do not display chattering, and their magnitudes are within the allowable ranges at the beginning of the transient behavior, these torques showing steady-state behaviors that converge to zero;
- The auxiliary velocity tracking errors  $e_c$  tend to converge to zero, and the RBFNN outputs  $(\hat{P}_v(\sigma^*))$  of the KNC controller also tend to converge to zero with the absence or presence of the neural term (neural compensator) of the KNC controller by use of Controls 1, 2, and 3;
- Both the sliding surfaces  $\sigma$ , new sliding surfaces  $\sigma^*$ , and their derivatives  $(\dot{\sigma}, \dot{\sigma}^*)$  converge to zero, and the chattering is eliminated for the use of Controls 1, 2, and 3;
- The values of the RBFNN outputs  $(\hat{P}_s(s))$  of the robustness term  $\gamma_s$  of the TNC controller of Control 3 show that if there are no disturbances  $\tau_p$ , these outputs tend to converge to zero, i.e., they present behaviors similar to the disturbances (magnitudes in absolute values) in the steady-state;
- The filtered tracking errors  $s$  converge to zero, and the chattering is eliminated by the use of Control 3.

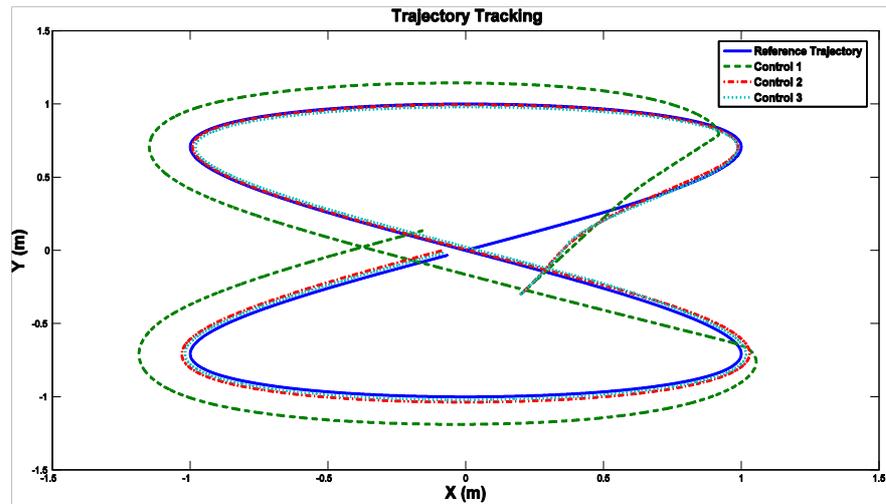


Figure 13. Trajectory tracking using the KNC controller without the neural term

#### 4.2. Case with disturbances: use of the KNC controller without the neural term

In Fig. 13, it appears that Control 1 does not show good performance in the reference trajectory tracking because of the disturbances that are not considered in the CTC controller design. The wheeled mobile robot tracks the reference trajectory by the use of the NNC controller of Control 2. With Control 3, which uses the TNC controller designed with neural terms to estimate the parameters of dynamics (parametric uncertainties) and compensate for the disturbances, the wheeled mobile robot is guided over a satisfactory reference trajectory.

Regarding the posture tracking errors shown in Fig. 14, it is noticed that the influence of the disturbances visibly affects the performance of Control 1, which (for reasons described previously) cannot compensate for the disturbances, thus showing significant posture tracking errors with behaviors alternating near zero. With the use of Controls 2 and 3, these errors converge quickly to zero, which is justified for the same reasons as described previously.

With respect to the control torques in Fig. 15, the behavior of the effects produced by Control 1 is similar to the behavior generated for Controls 2 and 3, except at the beginning of the trajectory and in the change of its direction. Furthermore, these effects, produced by Control 1 are not sufficient to compensate for the disturbances, resulting in auxiliary velocity tracking errors (Fig. 16) and consequent posture tracking errors, as well (Fig. 14). Therefore, the wheeled mobile robot does not adequately follow the reference trajectory (Fig. 13). The effects produced for Controls 2 and 3 compensate for the disturbances and allow the wheeled mobile robot to appropriately follow the reference trajectory. On

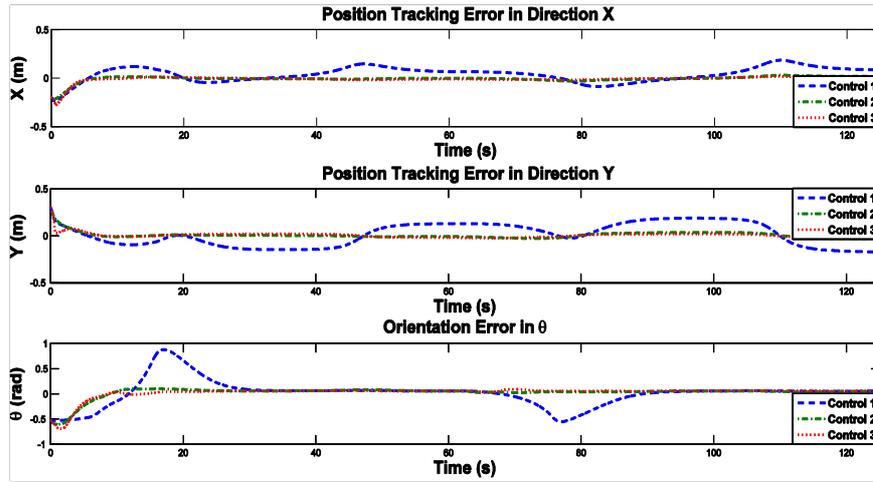


Figure 14. Posture tracking errors using the KNC controller without the neural term

the other hand, Fig. 15 verifies that in both Controls 1 and 3, the chattering effect does not appear in the behaviors of the control torques, whereas Control 2 presents some chattering at the beginning of the trajectory and in the change of its direction.

It is known that the KNC controller contains a function to correct the posture tracking errors, whereas the CTC, NNC or TNC controller aims to correct the auxiliary velocity tracking errors. With the integration of these controllers, which results in Controls 1, 2 and 3, perfect velocity tracking is not maintained. Thus, these auxiliary velocity tracking errors begin to be viewed as disturbances to the kinematic model.

Given the above statements, with the use of Control 1 (Figs. 16, 20, 21, 22 and 23), the first auxiliary velocity tracking error  $e_{c_1}$ , the first sliding surface  $\sigma_1$  and the derivative  $\dot{\sigma}_1$ , as well as the first new sliding surface  $\sigma_1^*$  and the derivative  $\dot{\sigma}_1^*$  tend to converge to zero. However, the second auxiliary velocity tracking error  $e_{c_2}$ , the second sliding surface  $\sigma_2$ , and the second new sliding surface  $\sigma_2^*$  do not converge to zero, whereas their derivatives  $\dot{\sigma}_2$ ,  $\dot{\sigma}_2^*$  converge to zero. This observation is due to inadequate compensation of the disturbances (Fig. 24), affecting the wheeled mobile robot, because these disturbances primarily influence the behavior of the control angular velocity (Fig. 25), thereby causing the related auxiliary velocity tracking error and the consequent posture tracking errors.

Regarding Control 2 (Figs. 17, 20, 21, 22 and 23), there is a behavior similar to that for Control 1 regarding the auxiliary velocity tracking errors  $e_c$ , the sliding surfaces  $\sigma$ , the new sliding surface  $\sigma^*$ , and their derivatives  $\dot{\sigma}$ ,  $\dot{\sigma}^*$ ; the disturbances (Fig. 24) have a small influence on the behavior of the control

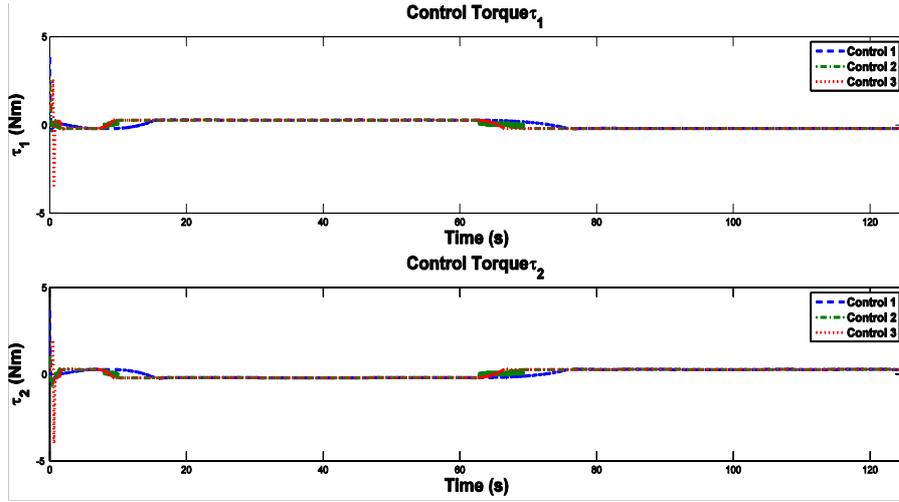


Figure 15. Control torques using the KNC controller without the neural term

angular velocity (Fig. 25), but the control torques were sufficient to compensate for these disturbances and make the wheeled mobile robot track the reference trajectory.

In the case of Control 3 (Figs. 18, 19, 20, 21, 22 and 23), the auxiliary velocity tracking errors  $e_c$ , the sliding surfaces  $\sigma$ , the new sliding surfaces  $\sigma^*$ , and the filtered tracking errors  $s$  tend to converge to zero, from which it can be said that the control velocities (Fig. 25) converge to their expected values, and the disturbances (Fig. 24) affecting the wheeled mobile robot are properly compensated.

It must be emphasized that for Controls 1, 2 and 3, the RBFNN outputs of the KNC controller (Figs. 16, 17 and 18) are zero at all times due to the absence of the neural term (neural compensator) of the KNC controller ( $\hat{W}_{\bar{\sigma}_k}$  is null). With the incidence of disturbances in the wheeled mobile robot, the RBFNN outputs of the robustness term  $\gamma_s$  of the TNC controller of Control 3 attempt to compensate for them and exhibit behaviors with magnitudes similar to the disturbances, as shown in Fig. 24. Additionally, in examining the amount of compensation of these disturbances, the auxiliary velocity tracking errors  $e_c$  and filtered tracking errors  $s$  are minor. Already the NNC controller of Control 2 attempts to compensate for them, by generating control torques through the estimation of the parameters, which justifies the parameter  $\hat{H}_{11}$  to be so far away from its true value. Furthermore, in examining the amount of compensation of these disturbances, the auxiliary velocity tracking errors  $e_c$  are minor.

Looking at the estimation of the parameters of the inertia matrix  $\bar{H}(q)$  by the NNC controller of Control 2, it is possible to observe that according to Fig. 26, the estimated parameter  $\hat{H}_{11}$  tends to converge to a value away

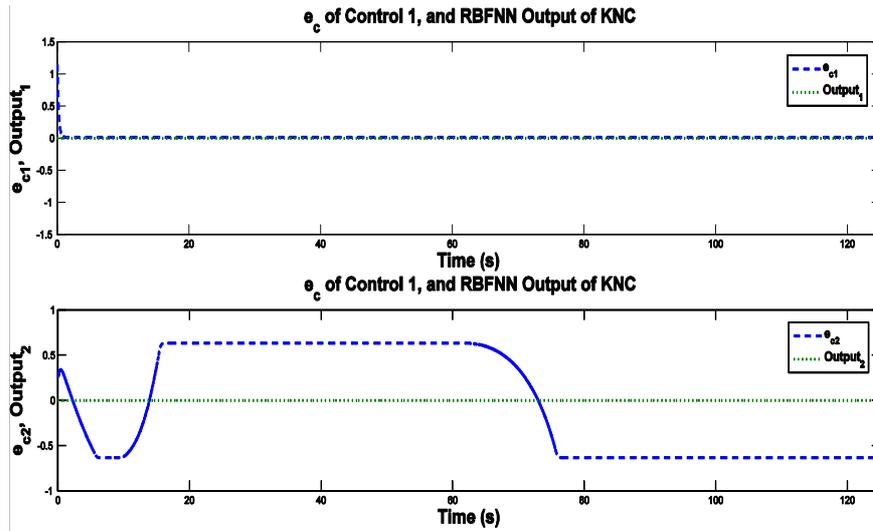


Figure 16. Auxiliary velocity tracking errors of Control 1 and RBFNN outputs ( $\hat{P}_v(\sigma^*)$ , (52)) of the KNC controller without the neural term

from its true value, whereas the estimated parameter  $\hat{H}_{22}$  tends to approach its true value. This is due to the fact that the NNC controller of Control 2 attempts to compensate for the disturbances generating control torques through this estimation of the parameters.

With respect to the estimation of the parameters of the inertia matrix  $\bar{H}(q)$  by the TNC controller of Control 3, it is possible to observe that according to Fig. 27, the estimated parameter  $\hat{H}_{11}$  tends to approach its true value. However, the estimated parameter  $\hat{H}_{22}$  tends to converge to a value away from its true value. This observation can be explained by the fact that the choice of adaptation gains  $\Gamma_{s_k}$  and the robustness term  $\gamma_s$  (65) of the TNC controller of Control 3 for compensation of the disturbances affecting the wheeled mobile robot will influence the behavior of the estimated parameters.

Moreover, for Controls 1, 2 and 3, no chattering occurs on the sliding surfaces and new sliding surfaces, as shown in Figs. 20 and 22, respectively. In Fig. 19, for Control 3, the chattering in the filtered tracking errors is eliminated.

By observing Figs. 15 and 25, it can be verified that no chattering occurs in the control torques (except in the Control 2) as well as in the linear and angular velocities.

Finally, one significant difference must be noted: Control 1 suffers only from the influence of the disturbances; Control 2 suffers from and compensates for the influence of disturbances through the estimation of the parameters, which may result in the imposition of unnecessary torques to the wheeled mobile robot; Control 3 suffers from and compensates for such influence through the RNC

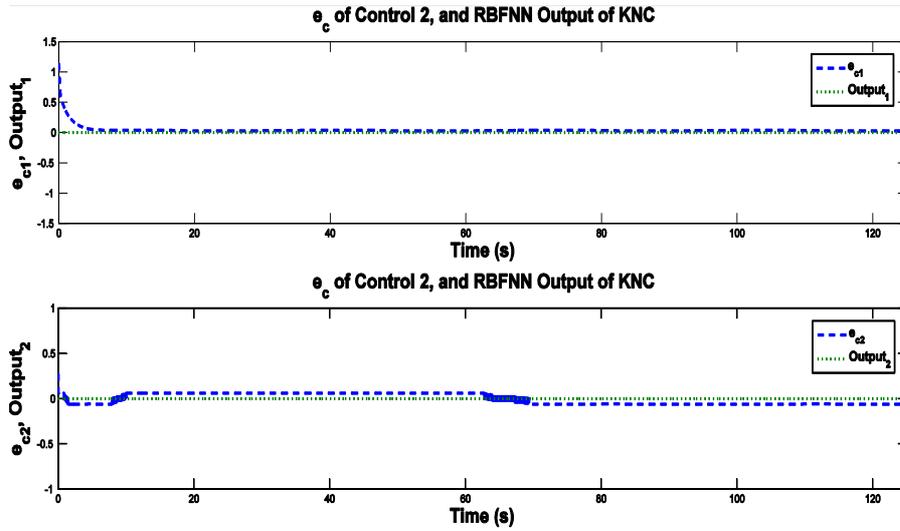


Figure 17. Auxiliary velocity tracking errors of Control 2 and RBFNN outputs ( $\hat{P}_v(\sigma^*)$ , (52)) of the KNC controller without neural term

of the TNC controller, and the DNC of the TNC controller only compensates for the dynamics (i.e., the inertia matrix), allowing the application of torques required for wheeled mobile robot.

#### 4.3. Case with disturbances: use of the KNC controller with the neural term

For the use of Controls 2 and 3, Fig. 28 verifies that the wheeled mobile robot presents good performance in reference trajectory tracking. The reasons for such performance are the same as the ones mentioned in Subsections 4.1 and 4.2. In the use of Control 1, a significant improvement is evidenced in the reference trajectory tracking compared with that of Control 1 of Subsection 4.2. Because the disturbances are not considered in the CTC controller, these events cause auxiliary velocity tracking errors and consequent posture tracking errors. However, the auxiliary velocity tracking errors (Fig. 31) are viewed as disturbances for the kinematic model and are compensated by the neural term (neural compensator) of the KNC controller, thus ensuring that the posture tracking errors tend to converge to zero (Fig. 29).

Observing Fig. 29, with the use of Controls 2 and 3, we see that the posture tracking errors converge quickly to zero, and the behaviors are similar to those of Controls 2 and 3 of Subsections 4.1 and 4.2. For the use of Control 1 and for reasons described previously, the posture tracking errors tend to converge to zero.

With respect to the control torques of Fig. 30, the comments of Subsection

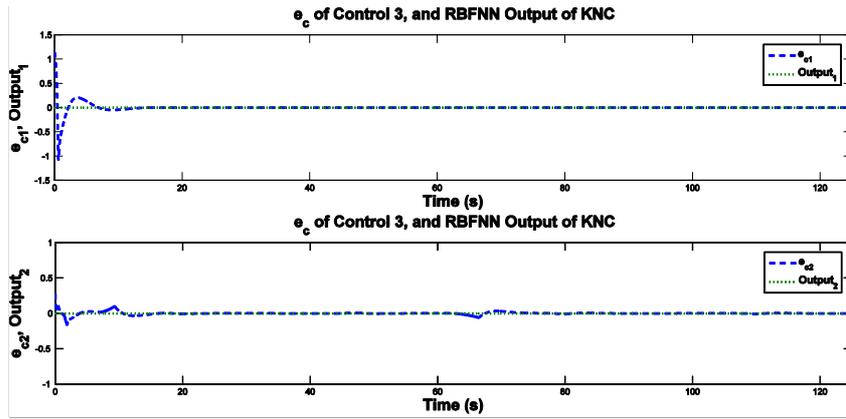


Figure 18. Auxiliary velocity tracking errors of Control 3 and RBFNN outputs ( $\hat{P}_v(\sigma^*)$ , (52)) of the KNC controller without neural term

4.2 are also valid for Controls 1, 2 and 3.

For Control 1, Fig. 31 shows that the existence of auxiliary velocity tracking errors must be due to disturbances that were not duly compensated by the CTC controller, which contains the function to correct these errors. However, because the function of the KNC controller is to correct the posture tracking errors, the RBFNN outputs of this controller attempt to compensate for the auxiliary velocity tracking errors, which are viewed as disturbances in the kinematic model. With this approach, the posture tracking errors tend to converge to zero and therefore show satisfactory improvement in trajectory tracking.

In Controls 2 and 3 (Figs. 32 and 33), the NNC and TNC controllers correct the auxiliary velocity tracking errors, because this controller has the ability to estimate the unknown parameters of the wheeled mobile robot dynamics and to compensate for the disturbances acting upon the wheeled mobile robot. Moreover, the RBFNN outputs of the KNC controller compensate for the possible auxiliary velocity tracking errors, thus correcting the posture tracking errors and producing more satisfactory reference trajectory tracking.

It is known that the KNC controller contains the function that corrects the posture tracking errors, whereas the CTC, NNC or TNC controller aims to correct the auxiliary velocity tracking errors. With the integration of these controllers, which results in Controls 1, 2 and 3, perfect velocity tracking is not maintained. Thus, these auxiliary velocity tracking errors begin to be viewed as disturbances for the kinematic model.

It should be noted that for Control 2, the filtered tracking errors (Fig. 34) converge to zero, and the chattering is eliminated. Additionally, for the Controls 1, 2 and 3, the sliding surfaces (Fig. 35) and new sliding surfaces (Fig. 37) converge to zero, and the chattering is eliminated.

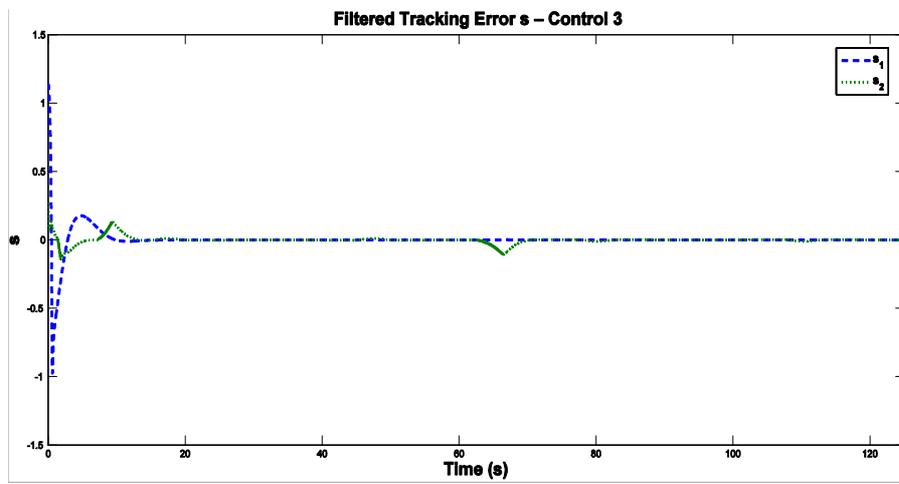


Figure 19. Filtered tracking errors using the KNC controller without the neural term

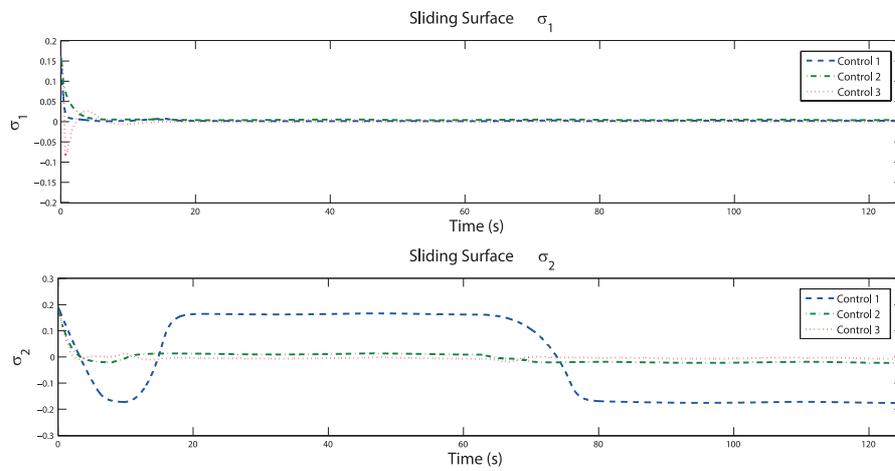


Figure 20. Sliding surfaces using the KNC controller without the neural term

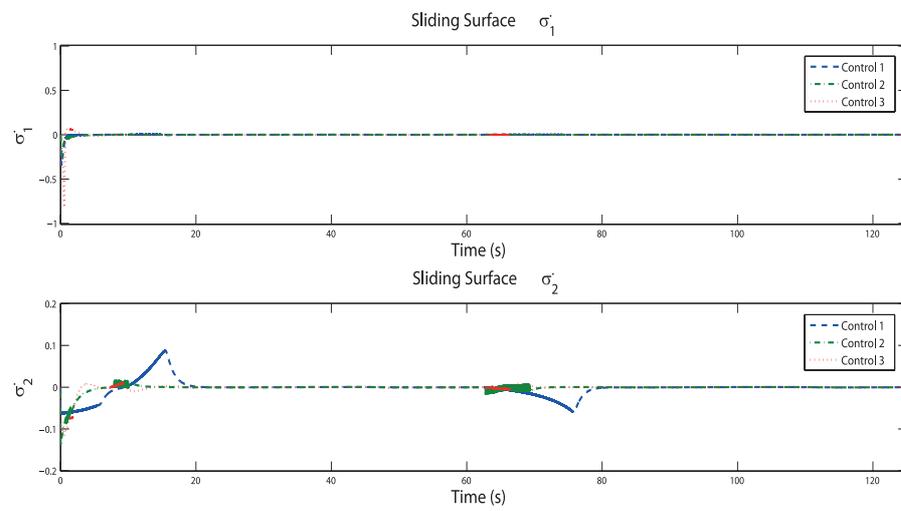


Figure 21. Derivatives of sliding surfaces using the KNC controller without the neural term

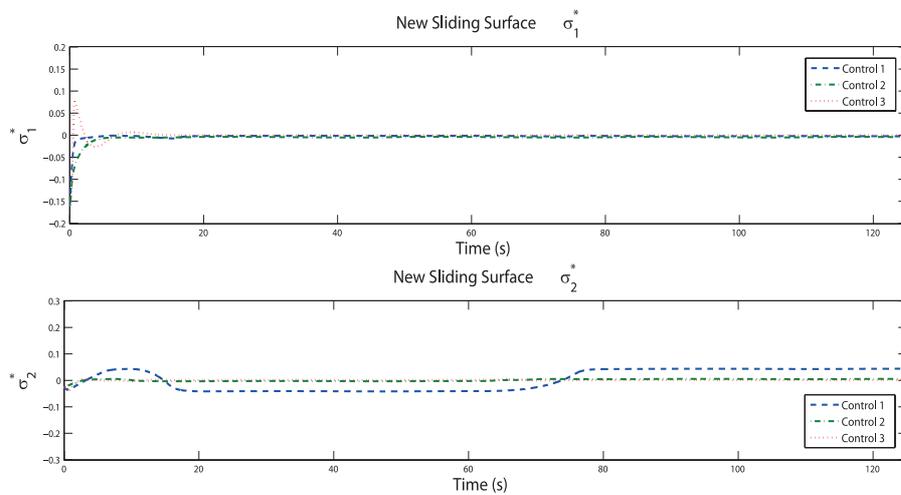


Figure 22. New sliding surfaces using the KNC controller without the neural term

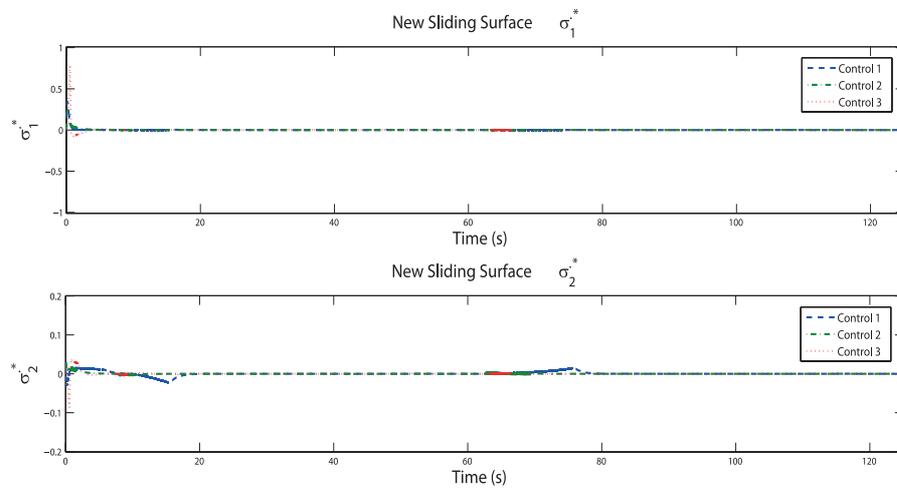


Figure 23. Derivatives of the new sliding surfaces using the KNC controller without the neural term

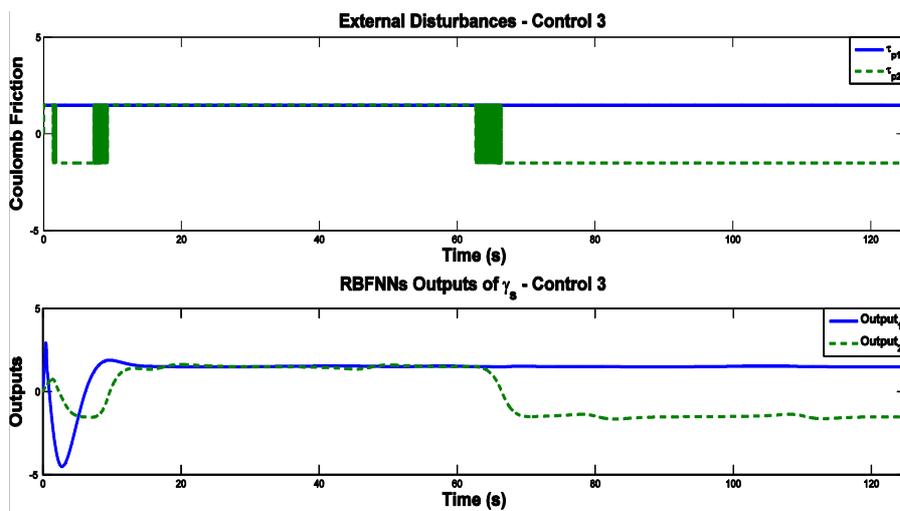


Figure 24. Disturbances (80) and (81) and RBFNN outputs ( $\hat{P}_s(s)$ ) of the robustness term  $\gamma_s$  (65) of the TNC controller of Control 3 using the KNC controller without the neural term

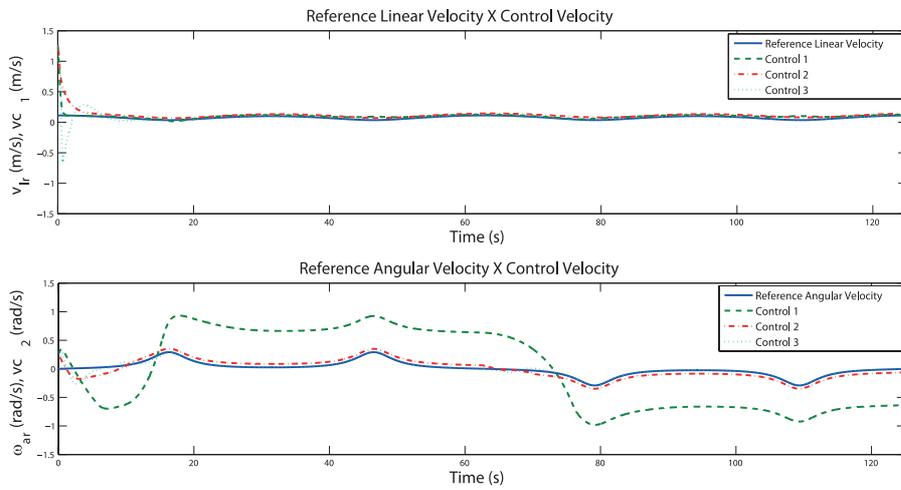


Figure 25. Profile of the velocities using the KNC controller without the neural term

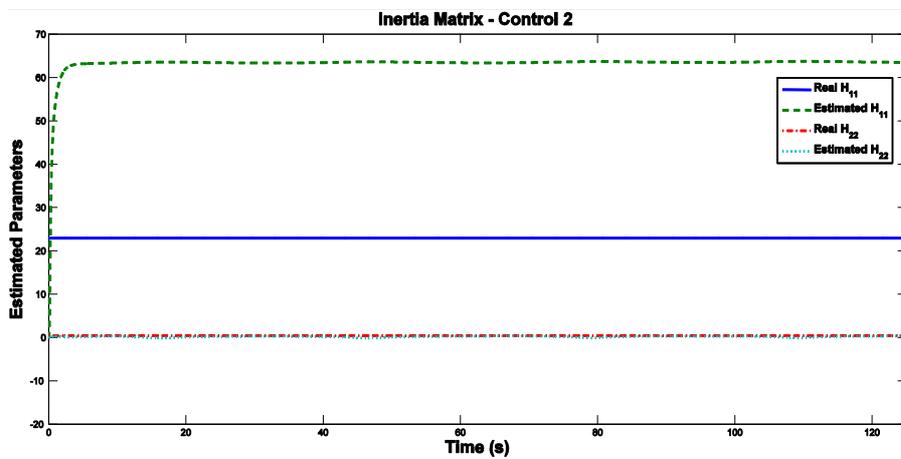


Figure 26. Estimated parameters for the inertia matrix  $\bar{H}(q)$  (62) by NNC controller using the KNC controller without the neural term

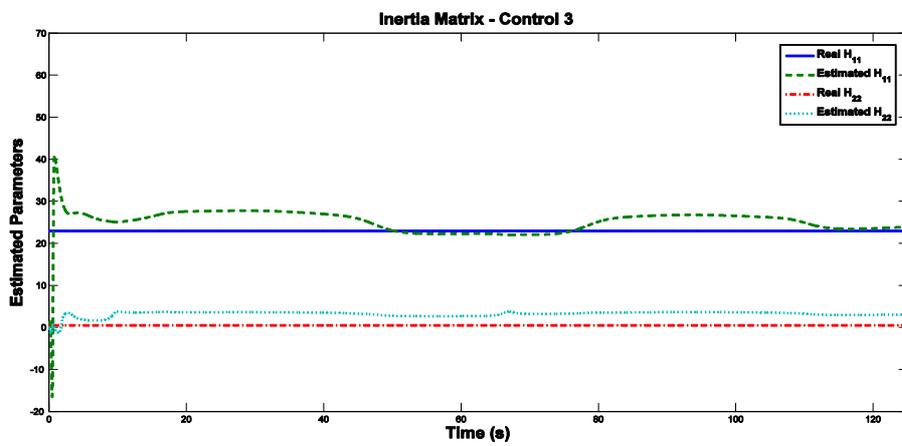


Figure 27. Estimated parameters for the inertia matrix  $\bar{H}(q)$  (62) by TNC controller using the KNC controller without the neural term

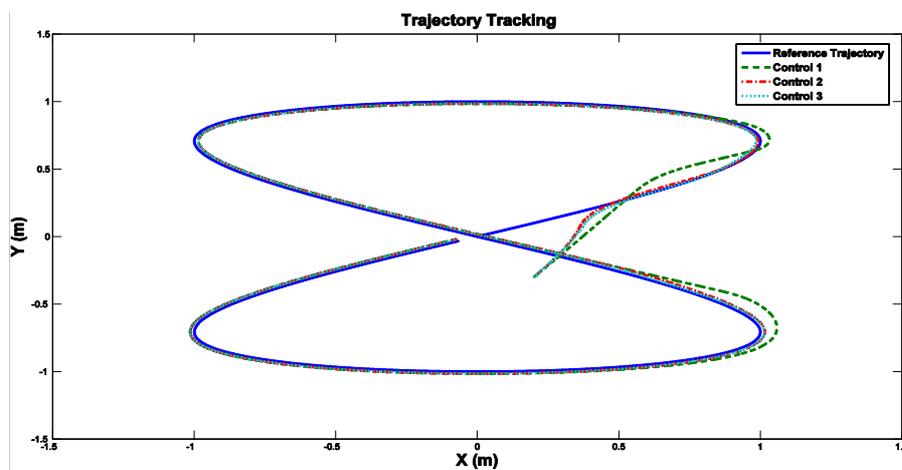


Figure 28. Trajectory tracking using the KNC controller with the neural term

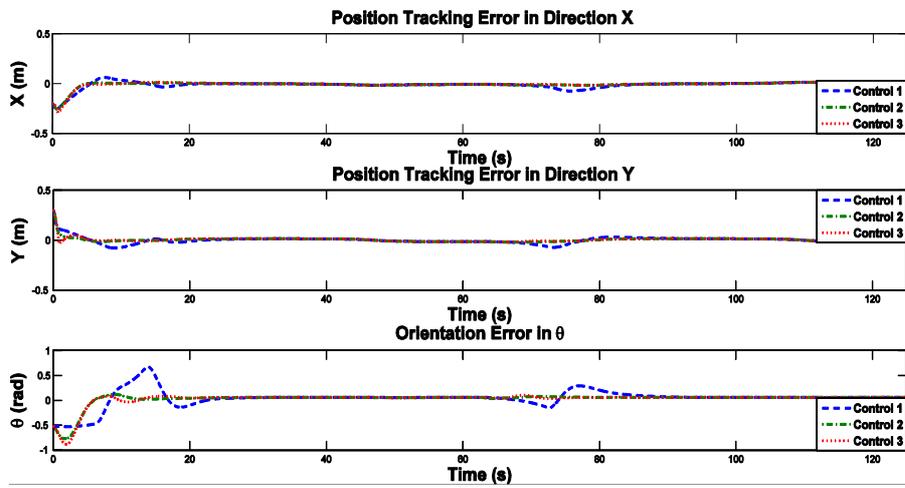


Figure 29. Posture tracking errors using the KNC controller with the neural term

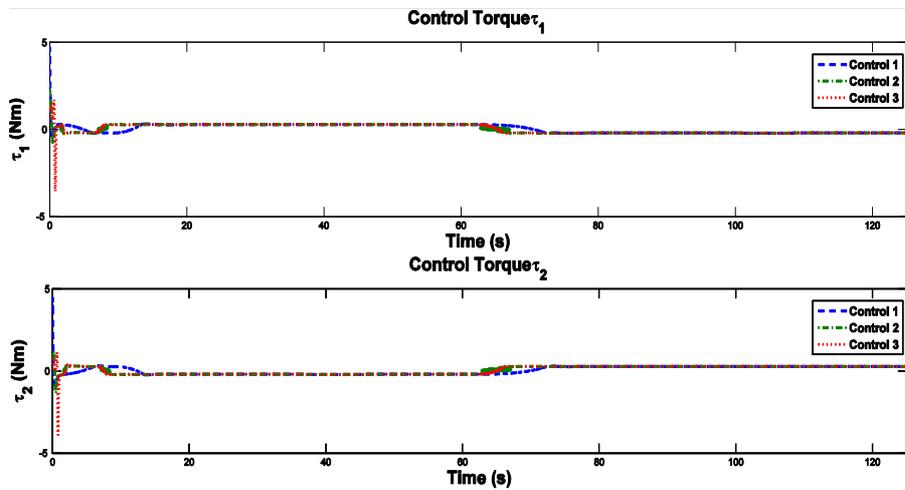


Figure 30. Control torques using the KNC controller with the neural term

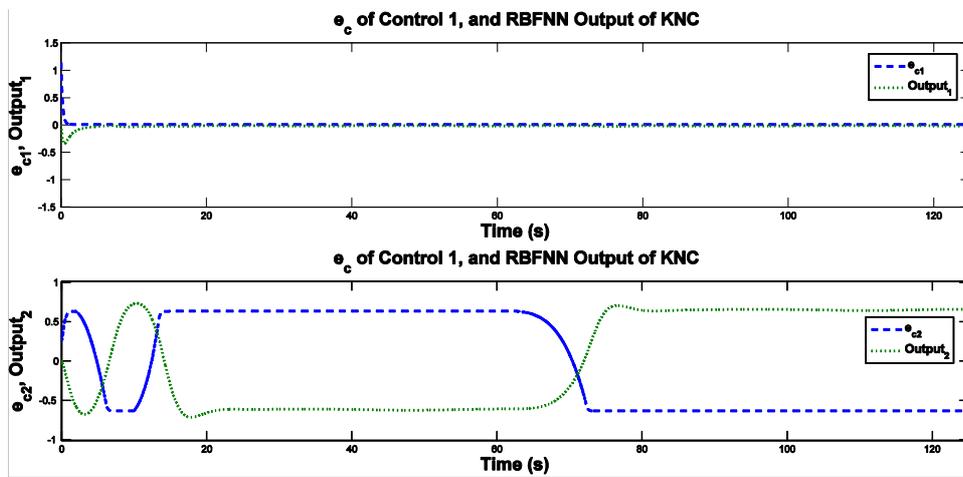


Figure 31. Auxiliary velocity tracking errors of Control 1 and RBFNN outputs  $(\hat{P}_v(\sigma^*), (52))$  of the KNC controller with the neural term

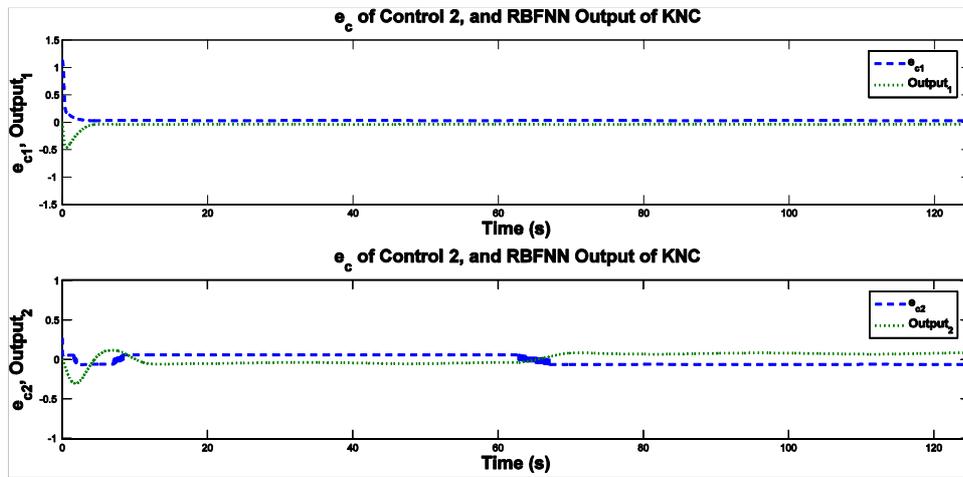


Figure 32. Auxiliary velocity tracking errors of Control 2 and RBFNN outputs  $(\hat{P}_v(\sigma^*), (52))$  of the KNC controller with the neural term