

NEURAL TORQUE CONTROLLERS FOR TRAJECTORY TRACKING PROBLEM OF A NONHOLONOMIC MOBILE ROBOT

Nardênio Martins, Douglas Bertol, Warody Lombardi, Edson R. Pieri, Eugênio Castelan

DAS-CTC-UFSC, Caixa Postal 476, CEP 88040-900 - Florianópolis SC
{nardenio, douglas, warody, edson, eugenio}@das.ufsc.br

Abstract - In this paper, a trajectory tracking control for a nonholonomic mobile robot by the integration of a kinematic controller and torque controllers is investigated. The proposed neural torque controllers (PNTCs) are based on a Gaussian radial basis function neural network (RBFNN) modeling technique, which are used to compensate the mobile robot dynamics, and bounded unknown disturbances. Also, the PNTCs are not dependent of the robot dynamics neither requires the off-line training process. The stability analysis and the convergence of tracking errors, as well as the learning algorithm for weights are guaranteed with basis on Lyapunov theory. In addition, the simulations results shows the efficiency of the PNTCs.

Keywords: Trajectory tracking, nonholonomic mobile robot, torque control, neural networks, Lyapunov theory.

1 INTRODUCTION

Autonomous mobile robots, which can move with intelligence without any human intervention, have attracted the interest of many researchers due to their extensive field of applications. However, mobile robots have nonholonomic constraints, that is, they can move only in the direction normal to the axis of the driving wheels.

Several control methods have been proposed for motion control of a mobile robot under nonholonomic constraints [Kanayama *et al.*, 1990; Sarker *et al.*, 1994; Fierro and Lewis, 1998; Yang and Kim, 1999; Fukao *et al.*, 2000, Hu and Yang, 2001; Oh *et al.*, 2003]. Some nonlinear feedback controllers have been suggested to solve such problems [Kanayama *et al.*, 1990], which consider only the kinematic model of a mobile robot and suppose 'perfect velocity tracking'. But it is not easy to design a dynamic controller for the accomplishment of 'perfect velocity tracking'.

Some researches consider the dynamics of mobile robot to achieve 'perfect velocity tracking' [Sarker *et al.*, 1994; Yang and Kim, 1999]. However, in these methods the perfect knowledge about the parameter values of mobile robot is necessary. Usually, such requirement is unattainable. In practical situations, obtaining exact parameter values on a mobile robot is almost impossible. There are few results on the problems, regarding the integration of both, kinematic and neural dynamic controllers for a mobile robot.

Controlling a mobile robot efficiently, with unknown dynamics, and subjected to the uncertainties and/or unmodeled significant disturbances is a field that has been attracting the attention of several researchers. The computed torque control approach is able to accomplish the control of mobile robot, but it demands the exact dynamics model that, in fact, is impossible in practice. Adaptive controllers [Fukao *et al.*, 2000] can perform the control of mobile robots even with partially unknown dynamics, however, a complicated online estimation of such unknown dynamics is necessary.

There are several controllers, based on neural networks, which have good results in areas where the model-based approaches have failed. Fierro and Lewis [1998] developed a neural network based model by combining the backstepping tracking technique with a torque controller, using a multi-layer feedforward neural network (known as MLP), that can learn the mobile robot's dynamics, the bounded unmodeled disturbances and the unstructured dynamics, through its online learning. However, both, the control and neural network learning algorithms are very complicated and it is computationally expensive.

In this paper, the RBFNN is applied to control a dynamic system, since the structure of an RBFNN is simpler than a multi-layer perceptron (MLP), the learning rate of RBFNN is generally faster than a MLP, and a RBFNN is mathematically tractable [Zhihong *et al.*, 1998; Seshagiri and Khalil, 2000]. It is important emphasize that neural networks only have the static mapping capability [Rumelhart and McClelland, 1986], however, when they are used as controllers, they must be able to realize the dynamics.

Differently from other investigations with neural networks in mobile robots [Fierro and Lewis, 1998; Hu and Yang, 2001; Oh *et al.*, 2003], the contributions are:

- The implementation of the PNTCs based on the partitioning of the RBFNN into several smaller subnets in order to obtain more efficient computation;

- The modelling by RBFNNs of the centripetal and Coriolis matrix $C(q, \dot{q})$ through of the inertia matrix $H(q)$ of the mobile robot dynamics. As result, the obtained PNTC is modeled with static RBFNNs only, what makes possible the reduction of the size of the RBFNNs, of the computational load and the implementation in real time;
- The PNTCs neither requires the knowledge of the mobile robot dynamics nor the time-consuming training process;
- The stability analysis and convergence of the mobile robot control system, and the learning algorithm for weights are proved by using Lyapunov theory, considering the presence of bounded unknown disturbances.

The paper is organized as follows: In Section 2 the nonholonomic mobile robot kinematics and dynamics are presented. The Ge-Lee (GL) matrix and its product operator are introduced for the stability analysis of neural networks. The problem of neural network approximation is briefly described. The neural networks modeling for mobile robots are shown. In Section 3 the kinematic controller (KC) for a reference trajectory tracking, and the PNTCs are described. In Section 4 the results of numerical simulations are shown and, finally, in Section 5 the conclusions are presented.

2 PRELIMINARIES

In this section the nonholonomic mobile robot kinematics and dynamics, the GL matrix and its product operator, the problem of neural network approximation and the neural networks modeling for mobile robots are described.

2.1 Kinematics and Dynamics of a Mobile Robot

A nonholonomic mobile robot is shown in Figure 1, and its local frame $\{C, X_c, Y_c\}$ and the inertial frame $\{O, X_o, Y_o\}$. The robot has two driving wheels mounted on the same axis and a passive front wheel. The both driving wheels are independently driven by two actuators and they are responsible to the motion and orientation. The mobile robot's position is given by the posture vector $q = [x_c \ y_c \ \theta]^T$, containing the center of mass (guidance point) C coordinates and the heading angle θ , with D , d , r , and $2R$ being intersection of the axis of symmetry with the driving wheel axis, distance from the point C to the point D , radius of wheels, and distance between wheels, respectively.

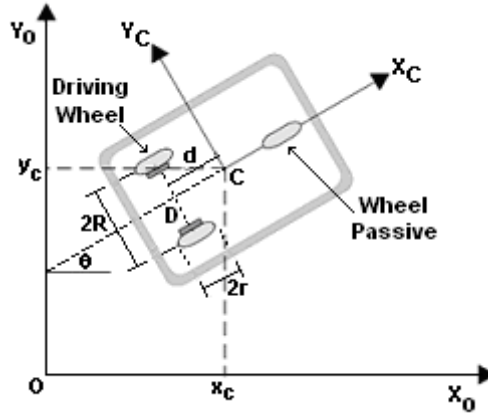


Figure 1: Model of a Nonholonomic Mobile Robot.

This mobile robot system has an n -dimensional configuration space C^n with generalized coordinates (q_1, \dots, q_n) and subject to p restrictions can be described by [Fierro and Lewis, 1998]:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (1)$$

where $H(q) \in \mathfrak{R}^{n \times n}$ is a symmetric, positive definite inertia matrix; $C(q, \dot{q}) \in \mathfrak{R}^{n \times n}$ is the centripetal and Coriolis matrix; $F(\dot{q}) \in \mathfrak{R}^{n \times 1}$ denotes the surface friction vector; $G(q) \in \mathfrak{R}^{n \times 1}$ is the gravitational vector; τ_d denotes the bounded unknown disturbances including unstructured and unmodeled dynamics; $B(q) \in \mathfrak{R}^{n \times (n-p)}$ is the input transformation matrix; $\tau \in \mathfrak{R}^{(n-p) \times 1}$ is the input vector; $A(q) \in \mathfrak{R}^{p \times n}$ is the matrix associated with the constraints; and $\lambda \in \mathfrak{R}^{p \times 1}$ is the vector of constraint forces.

The right side of equation (1) can be rewritten in the Lagrange-Euler formulation as:

$$\sum_{j=1}^n h_{kj}(q)\ddot{q}_j + \sum_{j=1}^n c_{kj}\dot{q}_j = B(q)\tau + A^T(q)\lambda - \tau_d \quad (2)$$

where:

$$c_{kj} = \sum_{i=1}^n c_{ijk}\dot{q}_i = \sum_{i=1}^n \frac{1}{2} \left(\frac{\partial h_{kj}}{\partial q_i} + \frac{\partial h_{ki}}{\partial q_j} - \frac{\partial h_{ij}}{\partial q_k} \right) \dot{q}_i \quad (3)$$

with the coefficient c_{ijk} is known as Christoffel symbols.

Considering the time-independence of all kinematic equality constraints one can write:

$$A(q)\dot{q} = 0 \quad (4)$$

Let $S(q)$ be a full rank matrix $(n - p)$ belonging to the null space of $A(q)$, that is:

$$A(q)S(q) = 0 \quad (5)$$

Based on equations (4)-(5), it is possible to find out $v(t) \in \mathfrak{R}^{n-p}$ such that, for all t :

$$\dot{q} = S(q)v(t) \quad (6)$$

The system, equation (1), is now transformed into a more appropriate representation for control purposes. By differentiating equation (6) and replacing the result in equation (1), then pre-multiplying by $S^T(q)$ and using equations (4)-(5), it is possible to eliminate the constraint matrix $A^T(q)\lambda$, resulting in:

$$S^T(q)H(q)S(q)\dot{v} + S^T(q)[H(q)\dot{S}(q) + C(q, \dot{q})S(q)]v + S^T(q)F(\dot{q}) + S^T(q)G(q) + S^T(q)\tau_d = S^T(q)B(q)\tau \quad (7)$$

Disregarding surface friction $F(\dot{q})$ and gravitational torques $G(q)$, one can rewritten equation (7) as follows:

$$\textbf{Dynamics 1} \quad S^T(q)H(q)S(q)\dot{v} + S^T(q)[H(q)\dot{S}(q) + C(q, \dot{q})S(q)]v = \bar{\tau} - \bar{\tau}_d \quad (8)$$

or

$$\textbf{Dynamics 2} \quad \bar{H}(q)\dot{v} + \bar{C}(q, \dot{q})v + \bar{\tau}_d = \bar{B}(q)\tau = \bar{\tau} \quad (9)$$

where $\bar{H}(q) = S^T(q)H(q)S(q)$, $\bar{C}(q, \dot{q}) = S^T(q)[H(q)\dot{S}(q) + C(q, \dot{q})S(q)]$, $\bar{\tau} = S^T(q)B(q)\tau = \bar{B}(q)\tau$, and $\bar{\tau}_d = S^T(q)\tau_d$.

Equations (6), (8)-(9) represent the nonholonomic mobile robot system in a local coordination system of the mobile robot. In fact $S(q)$ is a Jacobian matrix that transforms the independent velocities v in coordinates attached to the mobile base into constrained velocities \dot{q} in cartesian coordinates. Therefore, the properties of the original dynamics are maintained for the new set of coordinates [Fierro and Lewis, 1998]. Thus, the following pattern properties must be emphasized:

Property 1: Boundedness $\rightarrow \bar{H}(q)$, the norm of the $\bar{C}(q, \dot{q})$ and $\bar{\tau}_d$ are bounded.

Property 2: Skew-symmetry \rightarrow The matrix $\dot{\bar{H}}(q) - 2\bar{C}(q, \dot{q})$ is skew symmetric. This property is particularly important in the stability analysis of the control system.

To avoid the estimation of positions and orientation (posture vector q) as well as the use of velocities \dot{q} , due to the nonholonomic constraints pertinent of mobile robots, the equation (9) can be rewritten as:

$$\textbf{Dynamics 3} \quad \bar{\tau} = \bar{B}(q)\tau = \bar{H}(q, \dot{v}) + \bar{C}(q, \dot{q}, v) + \bar{\tau}_d = \hat{H}(\dot{v}) + \hat{C}(v) + \bar{\tau}_d \quad (10)$$

In this case, the dynamics of the equation (10) is described in a vectorial form, where the variables q and \dot{q} are implicitly in the formulation of this dynamics, but these variables are not used in the control design and stability proof.

Mobile robot's dynamics, equation (10), also it can be rewritten in a linear form,

$$\textbf{Dynamics 4} \quad \bar{\tau} = \hat{H}(\dot{v}) + \hat{C}(v) + \bar{\tau}_d = \Psi(v, \dot{v})\rho + \bar{\tau}_d \quad (11)$$

where $\Psi(v, \dot{v})$ is a coefficient matrix consisting of the known functions of robot velocity v and acceleration \dot{v} , which is referred as the robot regressor; and ρ is a vector consisting of the known and unknown robot dynamic parameters, such as geometric size, mass, moments of inertias, etc.

2.2 GL Matrix and Operator

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

Assuming that I_0 be 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 way:

$$\{\partial_k\} = \{\partial_{k1} \quad \partial_{k2} \quad \dots \quad \partial_{kn}\} \quad , \quad \{\partial_k\}^T = \{\partial_{k1}^T \quad \partial_{k2}^T \quad \dots \quad \partial_{kn}^T\} \quad (12)$$

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

$$\{\Theta\} = \begin{Bmatrix} \partial_{11} & \partial_{12} & \cdot & \partial_{1n} \\ \partial_{21} & \partial_{22} & \cdot & \partial_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \partial_{n1} & \partial_{n2} & \cdot & \partial_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\partial_1\} \\ \{\partial_2\} \\ \{\cdot\} \\ \{\partial_n\} \end{Bmatrix} \quad (13)$$

$$\{\Theta\}^T = \begin{Bmatrix} \partial_{11}^T & \partial_{12}^T & \cdot & \partial_{1n}^T \\ \partial_{21}^T & \partial_{22}^T & \cdot & \partial_{2n}^T \\ \cdot & \cdot & \cdot & \cdot \\ \partial_{n1}^T & \partial_{n2}^T & \cdot & \partial_{nn}^T \end{Bmatrix} \quad (14)$$

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

$$\{\Xi\} = \begin{Bmatrix} \zeta_{11} & \zeta_{12} & \cdot & \zeta_{1n} \\ \zeta_{21} & \zeta_{22} & \cdot & \zeta_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \zeta_{n1} & \zeta_{n2} & \cdot & \zeta_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\zeta_1\} \\ \{\zeta_2\} \\ \{\cdot\} \\ \{\zeta_n\} \end{Bmatrix} \quad (15)$$

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

$$[\{\Theta\}^T \bullet \{\Xi\}] = \begin{Bmatrix} \partial_{11}^T \zeta_{11} & \partial_{12}^T \zeta_{12} & \cdot & \partial_{1n}^T \zeta_{1n} \\ \partial_{21}^T \zeta_{21} & \partial_{22}^T \zeta_{22} & \cdot & \partial_{2n}^T \zeta_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \partial_{n1}^T \zeta_{n1} & \partial_{n2}^T \zeta_{n2} & \cdot & \partial_{nn}^T \zeta_{nn} \end{Bmatrix} \quad (16)$$

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

$$\Lambda_k \bullet \{\zeta_k\} = \{\Lambda_k\} \bullet \{\zeta_k\} = [\pi_{k1} \zeta_{k1} \quad \pi_{k2} \zeta_{k2} \quad \dots \quad \pi_{kn} \zeta_{kn}] \in R^{m \times n} \quad (17)$$

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, and then followed by the multiplication of $[\{A\} \bullet \{B\}]$ with matrix C .

2.3 Radial Basis Functions Neural Networks (RBFNNs)

In the field of control engineering, neural network is often used to approximate a given nonlinear function $f(y)$ up to a small error tolerance. The function approximation problem can be stated formally as follows.

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

$$d_f(\hat{f}(W^*, y), f(y)) \leq \varepsilon \quad (18)$$

for an acceptable small ε [Rice, 1964].

In this paper, Gaussian radial basis function (RBF) neural network is considered. It is a particular network architecture which uses l numbers of Gaussian function of the form:

$$a_i(y) = \exp\left(-\frac{(y - m_i)^T (y - m_i)}{\sigma^2}\right), i = 1, 2, \dots, l \quad (19)$$

where $m_i \in \mathfrak{R}^n$ is the center vector and $\sigma^2 \in \mathfrak{R}$ is the width. As shown in figure 2, each Gaussian RBF network 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 $\hat{f}(W, y)$ is given by:

$$\hat{f}(W, y) = W^T a(y) \quad (20)$$

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

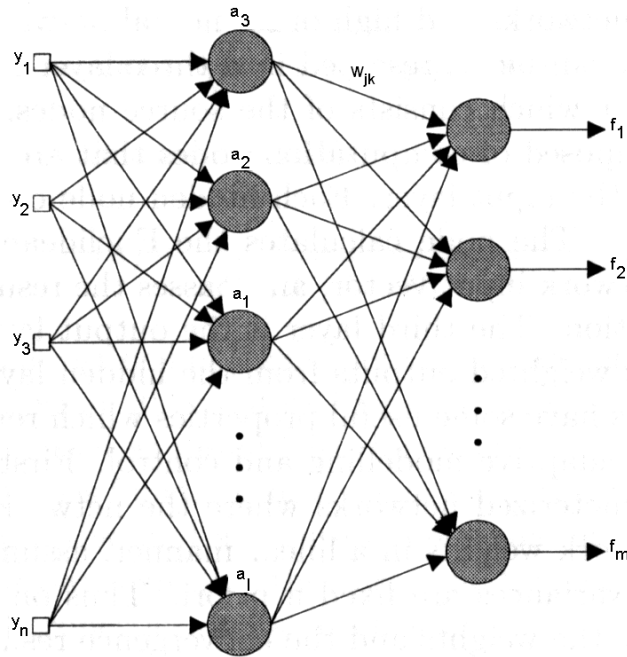


Figure 2: Schematic Diagram of RBF Neural Network.

Gaussian RBF network has been quite successful in representing the complex nonlinear function. It has been shown that a linear superposition of Gaussian radial basis function gives an optimal mean square approximation to an unknown function which is infinitely differentiable and 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 [Poggio and Girosi, 1990].

2.4 Modeling by RBFNNs

A $P(\cdot)$ vector or matrix can be modeled with static neural networks, since it is a function of a variable only. Thus, $P(\cdot)$ results in:

$$P(\cdot) = \left[\{W_P\}^T \bullet \{\xi_P(\cdot)\} \right] + E_P(\cdot) \quad (21)$$

where $\{W_P\}$ and $\{\xi_P(\cdot)\}$ are GL vectors or matrices, and their respective elements are W_{P_k} and $\xi_{P_k}(\cdot)$. $E_P(\cdot)$ is a vector or matrix, and their modeling error elements $\varepsilon_{P_k}(\cdot)$.

Based on equation (8), it can be verified that $H(q)$ is function of q only, thus, static neural networks are enough to model them. As a consequence, the size of the network can be much smaller compared with its dynamic counterparts.

It is well known that $h_{kj}(q)$ of equation (8) is infinite differentiable. Considering that the exact expression for $h_{kj}(q)$ is:

$$h_{kj}(q) = \sum_l W_{h_{kjl}} \xi_{h_{kjl}}(q) = W_{h_{kj}}^T \xi_{h_{kj}}(q) \quad (22)$$

where l denotes the number of hidden neurons, $W_{h_{kjl}}$ is the weight, and:

$$\xi_{h_{kjl}}(q, m_h, \sigma_h) = e^{\left(\frac{-\|q-m_h\|^2}{\sigma_h^2} \right)} = e^{\left(\frac{-(q-m_h)^T (q-m_h)}{\sigma_h^2} \right)} \quad (23)$$

Since

$$\frac{\partial h_{ij}}{\partial q_k} = -2 \frac{1}{\sigma_{h_k}^2} W_{h_{ij}}^T \xi_{h_{ij}}(q_k - m_{h_k}), \quad (24)$$

one has:

$$c_{ijk} = -\frac{1}{\sigma_h^2} \left(W_{h_{kj}}^T \xi_{h_{kj}}(q_i - m_{h_i}) + W_{h_{ki}}^T \xi_{h_{ki}}(q_j - m_{h_j}) - W_{h_{ij}}^T \xi_{h_{ij}}(q_k - m_{h_k}) \right) \quad (25)$$

or

$$c_{kj}(q, \dot{q}_i) = \sum_{i=1}^n c_{ijk} \dot{q}_i = -\frac{1}{\sigma_h^2} \sum_{i=1}^n W_{h_{kj}}^T \xi_{h_{kj}}(q_i - m_{h_i}) \dot{q}_i - \frac{1}{\sigma_h^2} \sum_{i=1}^n W_{h_{ki}}^T \xi_{h_{ki}}(q_j - m_{h_j}) \dot{q}_i + \frac{1}{\sigma_h^2} \sum_{i=1}^n W_{h_{ij}}^T \xi_{h_{ij}}(q_k - m_{h_k}) \dot{q}_i \quad (26)$$

where $\sigma_h = \sigma_{h_i} = \sigma_{h_j} = \sigma_{h_k}$.

It can be seen that the dynamics of mobile robots, equation (8), can be constructed by using the subnets for $H(q)$, because $C(q, \dot{q})$ can be constructed based on the parameters of $H(q)$. Note that since the $H(q)$ is function of q only, the subnets are static instead of dynamic, the size of the network is much smaller by introducing deterministic factors into the NN model. Thus, the matrix $C(q, \dot{q})$ is a function of $H(q)$, i. e.,

$$C(q, \dot{q}) = -\frac{1}{\sigma_H^2} H(q)(q - m_H)^T \dot{q} - \frac{1}{\sigma_H^2} H(q) \dot{q}(q - m_H)^T + \frac{1}{\sigma_H^2} (q - m_H) \dot{q}^T H(q) \quad (27)$$

or

$$C(q, \dot{q}) = -\frac{1}{\sigma_H^2} \left[\{W_H\}^T \bullet \{\xi_H(q)\} \right] (q - m_H)^T \dot{q} - \frac{1}{\sigma_H^2} \left[\{W_H\}^T \bullet \{\xi_H(q)\} \right] \dot{q} (q - m_H)^T + \frac{1}{\sigma_H^2} (q - m_H) \dot{q}^T \left[\{W_H\}^T \bullet \{\xi_H(q)\} \right] \quad (28)$$

In summary, the dynamics (**Dynamics 1**), equation (8), results in:

$$\bar{\tau} = S^T H(q) S \dot{v} + S^T (H \dot{S} + CS) v + S^T \tau_d = S^T \left[\{W_H\}^T \bullet \{\xi_H(q)\} \right] z_r + S^T \tau_d \quad (29)$$

with:

$$z_r = S \dot{v} + \dot{S} v - \underbrace{\left(-\frac{1}{\sigma_H^2} (q - m_H)^T \dot{q} - \frac{1}{\sigma_H^2} \dot{q} (q - m_H)^T \right)}_x S v - \underbrace{\left(\frac{1}{\sigma_H^2} (q - m_H) \dot{q}^T \right)}_y S v \quad (30)$$

In the dynamics, equations (9)-(10), it can be verified that $\bar{H}(q)$, $\hat{H}(\dot{v})$, and $\hat{C}(v)$ are functions of q , \dot{v} , and v only, respectively, thus, static neural networks are enough to model them. Assuming that $\bar{h}_{kj}(q)$, $\hat{h}_k(\dot{v})$, and $\hat{c}_k(v)$ can be modeled as:

$$\bar{h}_{kj}(q) = \sum_l X_{\bar{h}_{kjl}}^T \xi_{\bar{h}_{kjl}}(q) + \varepsilon_{\bar{h}_{kj}}(q) = X_{\bar{h}_{kj}}^T \xi_{\bar{h}_{kj}}(q) + \varepsilon_{\bar{h}_{kj}}(q) \quad (31)$$

$$\hat{h}_k(\dot{v}) = \sum_l W_{\hat{h}_{kl}}^T \xi_{\hat{h}_{kl}}(\dot{v}) + \varepsilon_{\hat{h}_k}(\dot{v}) = W_{\hat{h}_k}^T \xi_{\hat{h}_k}(\dot{v}) + \varepsilon_{\hat{h}_k}(\dot{v}) \quad (32)$$

$$\hat{c}_k(v) = \sum_l W_{\hat{c}_{kl}}^T \xi_{\hat{c}_{kl}}(v) + \varepsilon_{\hat{c}_k}(v) = W_{\hat{c}_k}^T \xi_{\hat{c}_k}(v) + \varepsilon_{\hat{c}_k}(v) \quad (33)$$

where $X_{\bar{h}_{kjl}}$, $W_{\hat{h}_{kl}}$, $W_{\hat{c}_{kl}} \in \mathfrak{R}$ are weights of the neural networks; $\xi_{\bar{h}_{kjl}}(q)$, $\xi_{\hat{h}_{kl}}(\dot{v})$, $\xi_{\hat{c}_{kl}}(v) \in \mathfrak{R}$ are Gaussian radial basis functions with their respective input vectors q , \dot{v} , and v only, as well as $\varepsilon_{\bar{h}_{kj}}(q)$, $\varepsilon_{\hat{h}_k}(\dot{v})$, $\varepsilon_{\hat{c}_k}(v) \in \mathfrak{R}$ are modeling errors of $\bar{h}_{kj}(q)$, $\hat{h}_k(\dot{v})$ and $\hat{c}_k(v)$, respectively, and are assumed to be bounded. Bearing in mind that $\bar{C}(q, \dot{q})$, equation (9), and $\bar{N}(v, \dot{v}) = \Psi(v, \dot{v})\rho$, equation (11), are dynamic neural networks, since they are functions of q and \dot{q} , v and \dot{v} , respectively, its modeling are required. Assuming that $\bar{c}_{kj}(q, \dot{q})$ and $\bar{n}_k(v, \dot{v})$ can be modeled as:

$$\bar{c}_{kj}(q, \dot{q}) = \sum_l X_{\bar{c}_{kjl}}^T \xi_{\bar{c}_{kjl}}(z) + \varepsilon_{\bar{c}_{kj}}(z) = X_{\bar{c}_{kj}}^T \xi_{\bar{c}_{kj}}(z) + \varepsilon_{\bar{c}_{kj}}(z) \quad (34)$$

$$\bar{n}_k(v, \dot{v}) = \sum_l W_{\bar{n}_{kl}}^T \xi_{\bar{n}_{kl}}(v, \dot{v}) + \varepsilon_{\bar{n}_k}(v, \dot{v}) = W_{\bar{n}_k}^T \xi_{\bar{n}_k}(v, \dot{v}) + \varepsilon_{\bar{n}_k}(v, \dot{v}) \quad (35)$$

where $z = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T \in R^{2n}$, $X_{\bar{c}_{kjl}}$, $W_{\bar{n}_{kl}} \in \mathfrak{R}$ are weights vectors; $\xi_{\bar{c}_{kjl}}(z)$, $\xi_{\bar{n}_{kl}}(v, \dot{v}) \in \mathfrak{R}$ are Gaussian radial basis functions with their respective input vectors z , v , and \dot{v} ; $\varepsilon_{\bar{c}_{kj}}(z)$, $\varepsilon_{\bar{n}_k}(v, \dot{v}) \in \mathfrak{R}$ are modeling errors of $\bar{c}_{kj}(q, \dot{q})$ and $\bar{n}_k(v, \dot{v})$, which are also assumed to be bounded.

Foregrounded in equations (9)-(11), the mobile robot's dynamics can be expressed by equations (31) and (34) for the **Dynamics 2**; equations (32) and (33) for the **Dynamics 3**; equation (35) for the **Dynamics 4**, respectively. Thus, $\bar{H}(q)$, $\bar{C}(q, \dot{q})$, $\hat{H}(\dot{v})$, $\hat{C}(v)$ and $\bar{N}(v, \dot{v})$ can be expressed as:

$$\textbf{Dynamics 2} \quad \bar{H}(q) = \left[\{X_{\bar{H}}\}^T \bullet \{\xi_{\bar{H}}(q)\} \right] + E_{\bar{H}}(q), \quad \bar{C}(q, \dot{q}) = \left[\{X_{\bar{C}}\}^T \bullet \{\xi_{\bar{C}}(z)\} \right] + E_{\bar{C}}(z) \quad (36)$$

$$\textbf{Dynamics 3} \quad \hat{H}(\dot{v}) = \left[\{W_{\hat{H}}\}^T \bullet \{\xi_{\hat{H}}(\dot{v})\} \right] + E_{\hat{H}}(\dot{v}), \quad \hat{C}(v) = \left[\{W_{\hat{C}}\}^T \bullet \{\xi_{\hat{C}}(v)\} \right] + E_{\hat{C}}(v) \quad (37)$$

$$\textbf{Dynamics 4} \quad \bar{N}(v, \dot{v}) = \left[\{W_{\bar{N}}\}^T \bullet \{\xi_{\bar{N}}(v, \dot{v})\} \right] + E_{\bar{N}}(v, \dot{v}) \quad (38)$$

where $\{X_{\bar{H}}\}$, $\{\xi_{\bar{H}}(q)\}$, $\{X_{\bar{C}}\}$, and $\{\xi_{\bar{C}}(z)\}$ are GL matrices, whereas $\{W_{\hat{H}}\}$, $\{\xi_{\hat{H}}(\dot{v})\}$, $\{W_{\hat{C}}\}$, $\{\xi_{\hat{C}}(v)\}$, $\{W_{\bar{N}}\}$, and $\{\xi_{\bar{N}}(v, \dot{v})\}$ are GL vectors; and their respective elements are $X_{\bar{h}_{kj}}$, $\xi_{\bar{h}_{kj}}(q)$, $X_{\bar{c}_{kj}}$, $\xi_{\bar{c}_{kj}}(z)$, $W_{\hat{h}_k}$, $\xi_{\hat{h}_k}(\dot{v})$, $W_{\hat{c}_k}$, $\xi_{\hat{c}_k}(v)$, $W_{\bar{n}_k}$, and $\xi_{\bar{n}_k}(v, \dot{v})$. $E_{\bar{H}}(q) \in R^{n \times n}$, and $E_{\bar{C}}(z) \in R^{n \times n}$ are matrices; $E_{\hat{H}}(\dot{v}) \in R^n$, $E_{\hat{C}}(v) \in R^n$, and $E_{\bar{N}}(v, \dot{v}) \in R^n$ are vectors, and their modeling error elements $\varepsilon_{\bar{h}_{kj}}(q)$, $\varepsilon_{\bar{c}_{kj}}(z)$, $\varepsilon_{\hat{h}_k}(\dot{v})$, $\varepsilon_{\hat{c}_k}(v)$, and $\varepsilon_{\bar{n}_k}(v, \dot{v})$, respectively.

3 CONTROL DESIGN

The function of a controller is to implement a mapping between the known information (e. g., reference position, velocity and sensor informations) and the actuator commands designed to achieve the robot's task. Thus, the controller design problem for the mobile robot can be described as: given the reference position $q_r(t)$ and the velocity $\dot{q}_r(t)$, design a control laws (PNTCs \rightarrow C1, C2, C3, and C4) for the actuator torques, which drives the mobile robot to move, so the mobile robot velocity carries out the tracking a smooth velocity control input and the reference position.

The following mild bounding assumptions always hold in practical applications and the development of the control design, and are required in order to proceed.

Assumption 2: On any compact subset of \mathfrak{R}^n , the ideal NN weights, centers, and widths are bounded by:

$$\|W\| \leq w_{\max}, \|X\| \leq x_{\max} \quad (39)$$

with w_{\max} , and x_{\max} being positive constants.

Assumption 3: On any compact subset of \mathfrak{R}^n , the NN approximation error is bounded by:

$$\|E_{NN}\| \leq e_{NN} \quad (40)$$

with e_{NN} being a positive constant.

Assumption 4: The desired reference trajectory is bounded so that $\|q_r\| \leq q_d$ with q_d a known scalar bound, and the disturbances are bounded so that $\|\bar{\tau}_d\| \leq b_d$ with b_d a known scalar bound.

Assumption 5: The reference linear velocity is nonzero, bounded, and $v_r > 0$ for all t . The angular velocity ω_r is bounded.

3.1 Kinematic Control (KC)

Let velocity and position of a reference robot be given as:

$$q_r = [x_r \quad y_r \quad \theta_r]^T, v_{ref} = [v_r \quad \omega_r]^T, \dot{x}_r = v_r \cos(\theta_r), \dot{y}_r = v_r \sin(\theta_r), \dot{\theta}_r = \omega_r \quad (41)$$

where $v_r > 0$ for all t is the reference linear velocity, and ω_r is the reference angular velocity. Thus, the position tracking error vector is expressed in the basis of a frame linked to the mobile robot platform as:

$$e_q = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (42)$$

The position error dynamics can be obtained from the time derivative of equation (42) as:

$$\dot{e}_q = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - v_1 + v_r \cos(e_3) \\ -\omega e_1 + v_r \sin(e_3) \\ \omega_r - \omega \end{bmatrix} \quad (43)$$

An auxiliary velocity control input v_c that achieves tracking for equation (6) that is given by [Kanayama *et al.*, 1990]:

$$v_c = \begin{bmatrix} v_r \cos(e_3) + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin(e_3) \end{bmatrix} \quad (44)$$

where k_1 , k_2 and k_3 are positive parameters.

Given the desired velocity v_c , one defines now the auxiliary velocity tracking error as:

$$e_c = v_c - v = \begin{bmatrix} v_{c1} - v_1 \\ v_{c2} - \omega \end{bmatrix} = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} \quad (45)$$

where v is the actual velocity of the mobile robot.

Stability and convergence analysis of this controller will be described later, through of the choice of a Lyapunov function candidate V_1 , but further details in Kanayama *et al.* [1990].

3.2 Dynamic Control (PNTCs)

Differentiating equation (45) and substituting the result in equations (8)-(11), respectively, the mobile robot dynamics using the velocity tracking error can be rewritten as:

$$\bar{H}(q)\dot{e}_c = -\bar{C}(z)e_c - \bar{\tau} + \Omega + \bar{\tau}_d \quad (46)$$

where the important nonlinear mobile robot function is:

$$\mathbf{C1} \quad \Omega = S^T H(q)S\dot{v}_c + S^T (H\dot{S} + CS)v_c \quad (47)$$

$$\mathbf{C2} \quad \Omega = \bar{H}(q)\dot{v}_c + \bar{C}(z)v_c \quad (48)$$

$$\mathbf{C3} \quad \Omega = \hat{H}(\dot{v}_c) + \hat{C}(v_c) \quad (49)$$

$$\mathbf{C4} \quad \Omega = \Psi(v_c, \dot{v}_c)\rho \quad (50)$$

Function Ω contains all the mobile robot parameters, such as masses, moments of inertias, etc, which are quantities often imperfectly known and difficult to determine.

Thus, the suitable control input for equations (8)-(11) respectively is given as:

$$\bar{\tau} = \hat{\tau}_{NN} + k_4 e_c - \gamma \quad (51)$$

where k_4 is a diagonal positive definite design matrix, and γ a robustifying term to compensate the unmodeled and unstructured disturbances, $\hat{\tau}_{NN}$ is the generated torque from the sum of output torque of the neural networks, that is given by:

$$\mathbf{C1} \quad \hat{\tau}_{NN} = S^T \left[\left\{ \hat{W}_H \right\}^T \bullet \left\{ \xi_H(q) \right\} \right] z_r \quad (52)$$

$$\mathbf{C2} \quad \hat{\tau}_{NN} = \left[\left\{ \hat{X}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(q) \right\} \right] \dot{v}_c + \left[\left\{ \hat{X}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(z) \right\} \right] v_c \quad (53)$$

$$\mathbf{C3} \quad \hat{\tau}_{NN} = \left[\left\{ \hat{W}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(\dot{v}_c) \right\} \right] + \left[\left\{ \hat{W}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(v_c) \right\} \right] \quad (54)$$

$$\mathbf{C4} \quad \hat{\tau}_{NN} = \left[\left\{ \hat{W}_{\bar{N}} \right\}^T \bullet \left\{ \xi_{\bar{N}}(v_c, \dot{v}_c) \right\} \right] \quad (55)$$

where $\{\hat{W}_H\}$, $\{\hat{X}_{\bar{H}}\}$, $\{\hat{X}_{\bar{C}}\}$, $\{\hat{W}_{\bar{H}}\}$, $\{\hat{W}_{\bar{C}}\}$, and $\{\hat{W}_{\bar{N}}\}$ represent estimates of true parameters of $\{W_H\}$, $\{X_{\bar{H}}\}$, $\{X_{\bar{C}}\}$, $\{W_{\bar{H}}\}$, $\{W_{\bar{C}}\}$, and $\{W_{\bar{N}}\}$ of equations (29), (36)-(38), respectively. Four controllers, in terms of input vectors of the RBFNN, are formed by: static neural networks (C1 and C3), static and dynamic neural networks (C2), and dynamic neural networks (C4).

Substituting equation (51) into equation (46), the closed-loop system error dynamics can be expressed as:

$$\bar{H}(q)\dot{e}_c = -(k_4 + \bar{C})e_c + \delta + \bar{\tau}_d + E_{NN} + \gamma \quad (56)$$

where:

$$\mathbf{C1} \quad \delta = S^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H(q) \right\} \right] z_r, E_{NN} = 0 \quad (57)$$

$$\mathbf{C2} \quad \delta = \left[\left\{ \tilde{X}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(q) \right\} \right] \dot{v}_c + \left[\left\{ \tilde{X}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(z) \right\} \right] v_c, E_{NN} = E_{\bar{H}}(q)\dot{v}_c + E_{\bar{C}}(z)v_c \quad (58)$$

$$\mathbf{C3} \quad \delta = \left[\left\{ \tilde{W}_{\bar{H}} \right\}^T \bullet \left\{ \xi_{\bar{H}}(\dot{v}_c) \right\} \right] + \left[\left\{ \tilde{W}_{\bar{C}} \right\}^T \bullet \left\{ \xi_{\bar{C}}(v_c) \right\} \right], E_{NN} = E_{\bar{H}}(\dot{v}) + E_{\bar{C}}(v) \quad (59)$$

$$\mathbf{C4} \quad \delta = \left[\left\{ \tilde{W}_{\bar{N}} \right\}^T \bullet \left\{ \xi_{\bar{N}}(v_c, \dot{v}_c) \right\} \right], E_{NN} = E_{\bar{N}}(v, \dot{v}) \quad (60)$$

being that $(\tilde{\cdot}) = (\cdot) - (\hat{\cdot})$ define the error vector in the parameters, and E_{NN} define the neural network modeling error.

By using the control input, equation (51), there is no guarantee that the torque $\bar{\tau}$ will make the velocity tracking error small. Thus, the control design problem is to specify a method of selecting the matrix gain k_4 , and the estimate $\hat{\tau}_{NN}$ so that the velocity error vector e_c is bounded. In addition, the estimate of $\hat{\tau}_{NN}$ should be bounded so that the control signals are also bounded.

Assuming that the unmodeled and unstructured disturbances are bounded, as well as the neural networks modeling errors, such that $\|\bar{\tau}_d\| \leq b_d$ and $\|E_{NN}\| \leq e_{NN}$, the robustifying term is defined as:

$$\gamma = -(k_d + I_n)e_c \quad (61)$$

where k_d is a diagonal positive definite matrix, and I_n is the identity matrix.

For the controllers (PNTCs \rightarrow C1, C2 C3, and C4), equations (51)-(55) and (61), learning algorithms for the neural networks should be developed, so that the control system can be stable, and both the position and velocity tracking errors converge to zero.

Let us consider Lyapunov's candidate function:

$$V = V_1 + k_1(e_1^2 + e_2^2) + 2\frac{k_1}{k_2}[1 - \cos(e_3)], \quad V_1 = \frac{1}{2}(e_c^T \bar{H}(q)e_c + V_2) \quad (62)$$

where the choice of V_2 depends on the controller used:

$$\mathbf{C1} \quad V_2 = \sum_{k=1}^n \tilde{W}_{Hk}^T \Gamma_{Hk}^{-1} \tilde{W}_{Hk} \quad (63)$$

$$\mathbf{C2} \quad V_2 = \sum_{k=1}^n \tilde{X}_{\bar{H}k}^T \Gamma_{\bar{H}k}^{-1} \tilde{X}_{\bar{H}k} + \sum_{k=1}^n \tilde{X}_{\bar{C}k}^T \Gamma_{\bar{C}k}^{-1} \tilde{X}_{\bar{C}k} \quad (64)$$

$$\mathbf{C3} \quad V_2 = \sum_{k=1}^n \tilde{W}_{\bar{H}k}^T \Gamma_{\bar{H}k}^{-1} \tilde{W}_{\bar{H}k} + \sum_{k=1}^n \tilde{W}_{\bar{C}k}^T \Gamma_{\bar{C}k}^{-1} \tilde{W}_{\bar{C}k} \quad (65)$$

$$\mathbf{C4} \quad V_2 = \sum_{k=1}^n \tilde{W}_{\bar{N}k}^T \Gamma_{\bar{N}k}^{-1} \tilde{W}_{\bar{N}k} \quad (66)$$

being Γ_k are dimensional compatible symmetric positive definite matrices. Clearly, $V \geq 0$, and $V = 0$ if only if $e_q = 0$, $e_c = 0$, $\{\tilde{W}_H\} = 0$, $\{\tilde{X}_{\bar{H}}\} = 0$, $\{\tilde{X}_{\bar{C}}\} = 0$, $\{\tilde{W}_{\bar{H}}\} = 0$, $\{\tilde{W}_{\bar{C}}\} = 0$, and $\{\tilde{W}_{\bar{N}}\} = 0$.

In the sequel the stability analysis will be only performed to the controller C1. Similar results can be obtained to C2, C3 and C4. Differentiating V , equation (62), and substituting the error dynamics, equation (56), \dot{V} is given as:

$$\dot{V} = 2k_1 e_1 \dot{e}_1 + 2k_1 e_2 \dot{e}_2 + 2\frac{k_1}{k_2} \dot{e}_3 \sin(e_3) + \dot{V}_1, \quad \dot{V}_1 = -e_c^T k_4 e_c + e_c^T \bar{\tau}_d + e_c^T E_{NN} + e_c^T \gamma + \dot{V}_2 \quad (67)$$

where, using the property 2, $\dot{\bar{H}} - 2\dot{\bar{C}}$, \dot{V}_2 stays:

$$\dot{V}_2 = e_c^T \left(S^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H(q) \right\} \right] \right) z_r - \sum_{k=1}^n \tilde{W}_{Hk}^T \Gamma_{Hk}^{-1} \dot{\tilde{W}}_{Hk} \quad (68)$$

It can be seen that $\left\{ \tilde{W}_{Hk} \right\} = \left\{ W_{Hk} \right\} - \left\{ \hat{W}_{Hk} \right\}$, then $\left\{ \dot{\tilde{W}}_{Hk} \right\} = -\left\{ \dot{\hat{W}}_{Hk} \right\}$.

Recall that:

$$\begin{aligned} e_c^T \left(S^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] \right) z_r &= e_c^T \left(S^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] \right) (S \dot{v}_c + \dot{S} v_c - (x + y) S v_c) \\ e_c^T \left(S^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] \right) (S \dot{v}_c + \dot{S} v_c - x S v_c) &= r_r^T \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] (S \dot{v}_c + \dot{S} v_c - x S v_c) = \sum_{k=1}^n \left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} (S \dot{v}_c + \dot{S} v_c - x S v_c) r_{rk} \\ e_c^T S^T \left(-\frac{1}{\sigma_H^2} (q - m_H) \dot{q}^T \right) \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] S v_c &= r_r^T \left(-\frac{1}{\sigma_H^2} (q - m_H) \dot{q}^T \right) \left[\left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} \right] S v_c = \sum_{k=1}^n r_r^T \left(-\frac{1}{\sigma_H^2} (q - m_H) \right) \left\{ \tilde{W}_H \right\}^T \bullet \left\{ \xi_H \right\} S v_c \dot{q}_k \end{aligned} \quad (69)$$

and the learning laws for the neural networks are obtained as:

$$\dot{\hat{W}}_{H_k} = \Gamma_{H_k} \bullet \left\{ \xi_{H_k} \right\} \left((S\dot{v}_c + \dot{S}v_c - xSv_c)r_{r_k} - r_r \left(\frac{1}{\sigma_H} (q - m_H) \right) Sv_c \dot{q}_k \right) - K_{H_k} \Gamma_{H_k} \|e_c\| \hat{W}_{H_k} \quad (70)$$

where $K_{H_k} > 0$. The term $K_{H_k} \Gamma_{H_k} \|e_c\| \hat{W}_{H_k}$, equation (70), correspond to ε modification [Narendra and Annaswamy, 1987] from the adaptive control theory; they must be added to eliminate the condition of persistent excitation and to ensure bounded of neural networks weights estimates.

Then \dot{V}_1 of equation (67) can be simplified as:

$$\dot{V} \leq 2k_1 e_1 \dot{e}_1 + 2k_1 e_1 \dot{e}_2 + 2 \frac{k_1}{k_2} \dot{e}_3 \sin(e_3) + \dot{V}_1, \quad \dot{V}_1 \leq -e_c^T k_4 e_c + e_c^T \bar{\tau}_d + e_c^T E_{NN} + e_c^T \gamma + \dot{V}_2 \quad (71)$$

where:

$$\dot{V}_2 \leq K_H \|e_c\| \sum_{k=1}^n \tilde{W}_{H_k}^T \hat{W}_{H_k} \quad (72)$$

being that $K_H = K_{H_k}$ of equation (72). Observing that:

$$tr(\tilde{W}_H^T \hat{W}_H) = \sum_{k=1}^n \tilde{W}_{H_k}^T \hat{W}_{H_k} \quad (73)$$

with $tr(\cdot)$ being trace function, and replacing the robustifying term, equation (61), in \dot{V}_1 of equation (71), one obtains:

$$\dot{V} = 2k_1 e_1 \dot{e}_1 + 2k_1 e_1 \dot{e}_2 + 2 \frac{k_1}{k_2} \dot{e}_3 \sin(e_3) + \dot{V}_1, \quad \dot{V}_1 \leq -e_c^T e_c - e_c^T k_4 e_c - e_c^T k_d e_c + e_c^T \bar{\tau}_d + e_c^T E_{NN} + \dot{V}_2 \quad (74)$$

where:

$$\dot{V}_2 \leq \|e_c\| K_H tr(\tilde{W}_H^T \hat{W}_H) \quad (75)$$

or

$$\dot{V}_2 \leq \|e_c\| K_H tr(\tilde{W}_H^T (W_H - \tilde{W}_H)) \quad (76)$$

Since:

$$tr(\tilde{W}_H^T (W_H - \tilde{W}_H)) = \langle \tilde{W}_H, W_H \rangle - \|\tilde{W}_H\|_F^2 \leq \|\tilde{W}_H\|_F \|W_H\|_F - \|\tilde{W}_H\|_F^2 \quad (77)$$

the equation (74) becomes:

$$\dot{V} = 2k_1 e_1 \dot{e}_1 + 2k_1 e_1 \dot{e}_2 + 2 \frac{k_1}{k_2} \dot{e}_3 \sin(e_3) + \dot{V}_1, \quad \dot{V}_1 = -e_c^T e_c - k_{4\min} \|e_c\|^2 - k_{d\min} \|e_c\|^2 + (b_d + e_{NN}) \|e_c\| + \dot{V}_2 \quad (78)$$

where $k_{4\min}$ and $k_{d\min}$ are the minimum singular values of k_4 and k_d , respectively; and:

$$\dot{V}_2 \leq \|e_c\| K_H (\|\tilde{W}_H\|_F \|W_H\|_F - \|\tilde{W}_H\|_F^2) \quad (79)$$

Substituting the position error dynamics, equation (43), the time derivative of V results in:

$$\dot{V} \leq 2k_1 e_1 [\omega e_2 - v_1 + v_r \cos(e_3)] + 2k_1 e_2 [v_r \sin(e_3) - \omega e_1] + 2k_3 v_r (\omega_r - \omega) \sin(e_3) + \dot{V}_1 \quad (80)$$

$$\dot{V}_1 = -e_c^T e_c - k_{4\min} \|e_c\|^2 - k_{d\min} \|e_c\|^2 + (b_d + e_{NN}) \|e_c\| + \dot{V}_2$$

where $k_3 v_r = \frac{k_1}{k_2}$. Substituting e_c and $v = v_c - e_c$, equation (45), \dot{V} becomes:

$$\dot{V} \leq -k_1^2 e_1^2 - \frac{k_1 k_3}{k_2} v_r \sin^2(e_3) - (k_1 e_1 - e_4)^2 - \frac{k_1}{k_2 k_3 v_r} [k_3 v_r \sin(e_3) - e_5]^2 - k_{4\min} \|e_c\|^2 - k_{d\min} \|e_c\|^2 + (b_d + e_{NN}) \|e_c\| + \dot{V}_2 \quad (81)$$

Since the first four terms in equation (81) are negative, there results:

$$\dot{V} \leq -\|e_c\| (k_{4\min} \|e_c\| + k_{d\min} \|e_c\| - (b_d + e_{NN}) + K_H (\|\tilde{W}_H\|_F - \frac{W_{H\max}}{2})^2 - K_H \frac{W_{H\max}^2}{4}) \quad (82)$$

Thus, \dot{V} is guaranteed negative as long as the term in parentheses in equation (82) is positive, either:

$$\|e_c\| > \frac{\eta + b_d + e_{NN}}{k_{4\min} + k_{d\min}}, \quad \eta = K_H \frac{W_{H\max}^2}{4} \quad (83)$$

or

$$\|\tilde{W}_H\|_F > \frac{W_{H\max}}{2} + \sqrt{\frac{W_{H\max}^2}{4} + \frac{b_d + e_{NN}}{K_H}} \quad (84)$$

Therefore, \dot{V} is guaranteed negative. According to a standard *Lyapunov* theory, all signals $\|e_q\|$, $\|e_c\|$, and $\|\tilde{W}_H\|_F$ are bounded.

4 SIMULATIONS RESULTS

For accomplishing the simulations, the dynamic model described in Fierro and Lewis [1998] is used, where the parameters of the mobile robot are as follows: $m = 10\text{kg}$, $I = 5\text{kgm}^2$, $R = 0.5\text{m}$, $r = 0.05\text{m}$, $d = 0.2\text{m}$, $v_r = 0.5\text{m/s}$, and $\omega_r = 0.0\text{rad/s}$.

To illustrate the performance of the four PNTCs (C1, C2, C3, and C4 or equations (51)-(55), and (61)), the robot should to track a straight line with the disturbances that subject to sudden changes, as well as payload influence.

The reference trajectory is a straight line with initial coordinates $[x_r(0), y_r(0), \theta_r(0)] = [1, 2, 25.56^\circ]$. The initial position of the robot is $[x_c(0), y_c(0), \theta(0)] = [2, 1, 10^\circ]$. The design parameters of controllers are chosen as: KC $\rightarrow k_1 = 1$, $k_2 = 3$, $k_3 = 2$, and PNTCs:

C1 $\rightarrow k_4 = \text{diag}[7]$, $\Gamma_{H_k} = 10$, $\sigma_H^2 = 9$, $K_H = 0.001$, and $k_d = \text{diag}[17]$;

C2, C3, and C4 $\rightarrow k_4 = \text{diag}[7]$, $\Gamma_{\cdot k} = 10$, $\sigma_{\cdot}^2 = 49$, $K_{\cdot} = 0.001$, and $k_d = \text{diag}[7]$.

In this case, the number of hidden neurons is: 4 for C1, 20 for C2, 8 for C3 and 16 for C4. The values of weights (W_{\cdot} , and X_{\cdot}) of the RBFNNs are adjusted on-line and all initialised to zero. The centers (m_{\cdot}) of the localised Gaussian RBFs are evenly distributed to span the input space. Different tracking performance can be achieved by adjusting parameters gains and others factors, such as the size of the RBFNNs, centers and widths of the Gaussian RBFs.

A Coulomb friction term as unmodeled disturbance, as well as bounded periodic disturbance are added to the robot system as,

$$\bar{F} = \begin{bmatrix} (f_1 + f_1(t))\text{sgn}(v_1) + 0.1\sin(2t) \\ (f_2 + f_2(t))\text{sgn}(\omega) + 0.1\cos(2t) \end{bmatrix}$$

where $f_1 = 0.3$, and $f_2 = 0.5$. Function $f(t)$ is nonlinear, defined as: $[f_1(t) \ f_2(t)] = [0.0 \ 0.0]^T$ if $t < 8$; $[f_1(t) \ f_2(t)] = [0.2 \ 0.2]^T$ if $t \geq 8$, respectively. Thus, disturbance is subject to a sudden change at time goes to 8 sec. Moreover, in 8 sec, the mobile robot suddenly dropped of an object of 2.5kg , that is, a quarter of its original mass.

The trajectory tracking obtained by the controllers (only $\bar{\tau} = k_4 e_c$ of equation (51), and PNTCs \rightarrow C1, C2, C3, C4) can be quantified (see Table 1) using the mean square quadratic error (MSE).

Table 1: Mean Square Quadratic Error – MSE.

Controller	MSE		
	X Direction	Y Direction	θ Orientation
$\bar{\tau} = k_4 e_c$	0.0609	0.0930	0.1397
C1	0.0486	0.0519	0.0886
C2	0.0464	0.0437	0.0812
C3	0.0512	0.0582	0.1166
C4	0.0395	0.0564	0.0884

In addition, the tracking performance of the PNTCs (C1, C2, C3, and C4) are verified in the tracking of the reference trajectory as shown in Figure 3. It can be noted that the both actual trajectories converge to the reference trajectory.

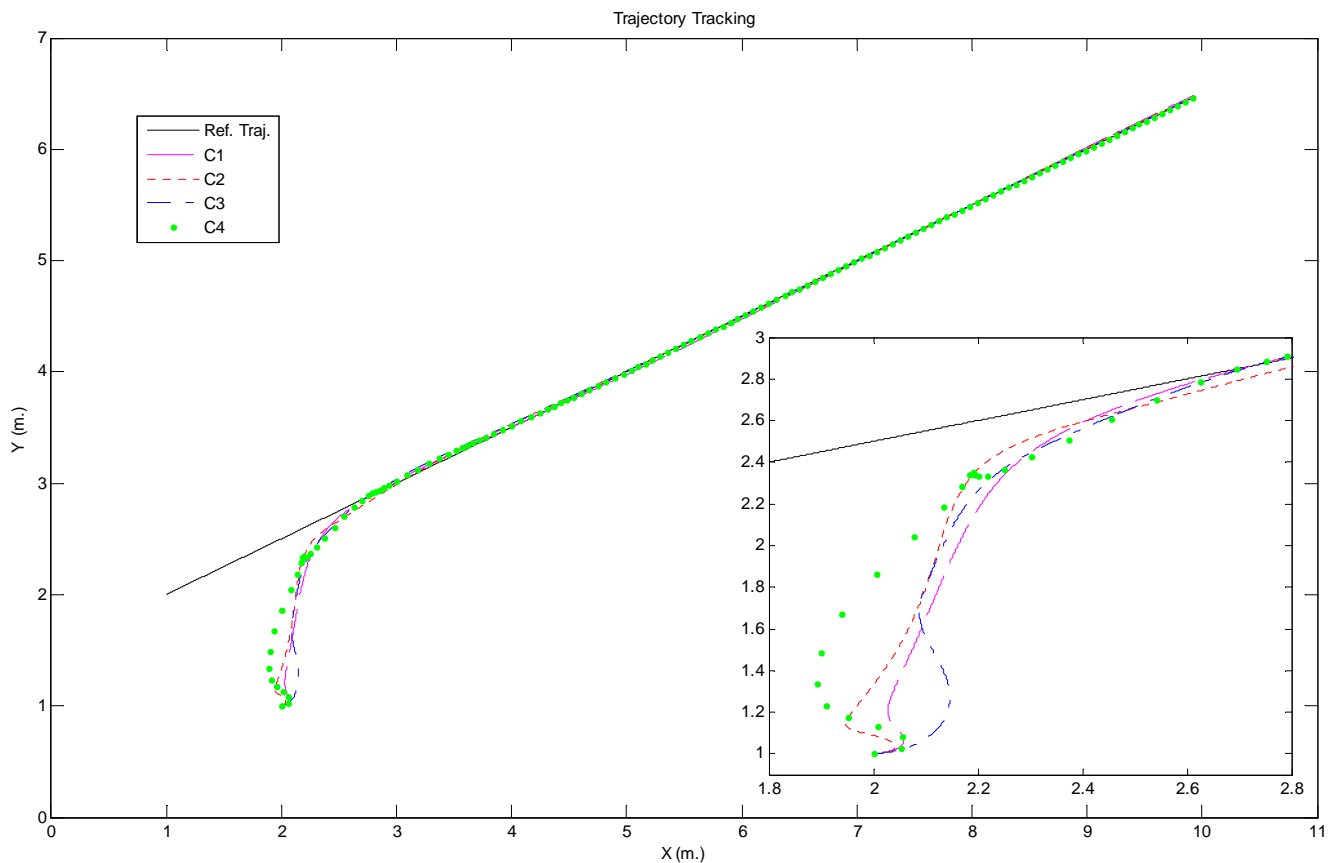


Figure 3: Tracking Trajectory – PNTCs.

Similar conclusions can be obtained in Figure 4, the tracking errors in the X and Y directions, and in the orientation converge to zero.

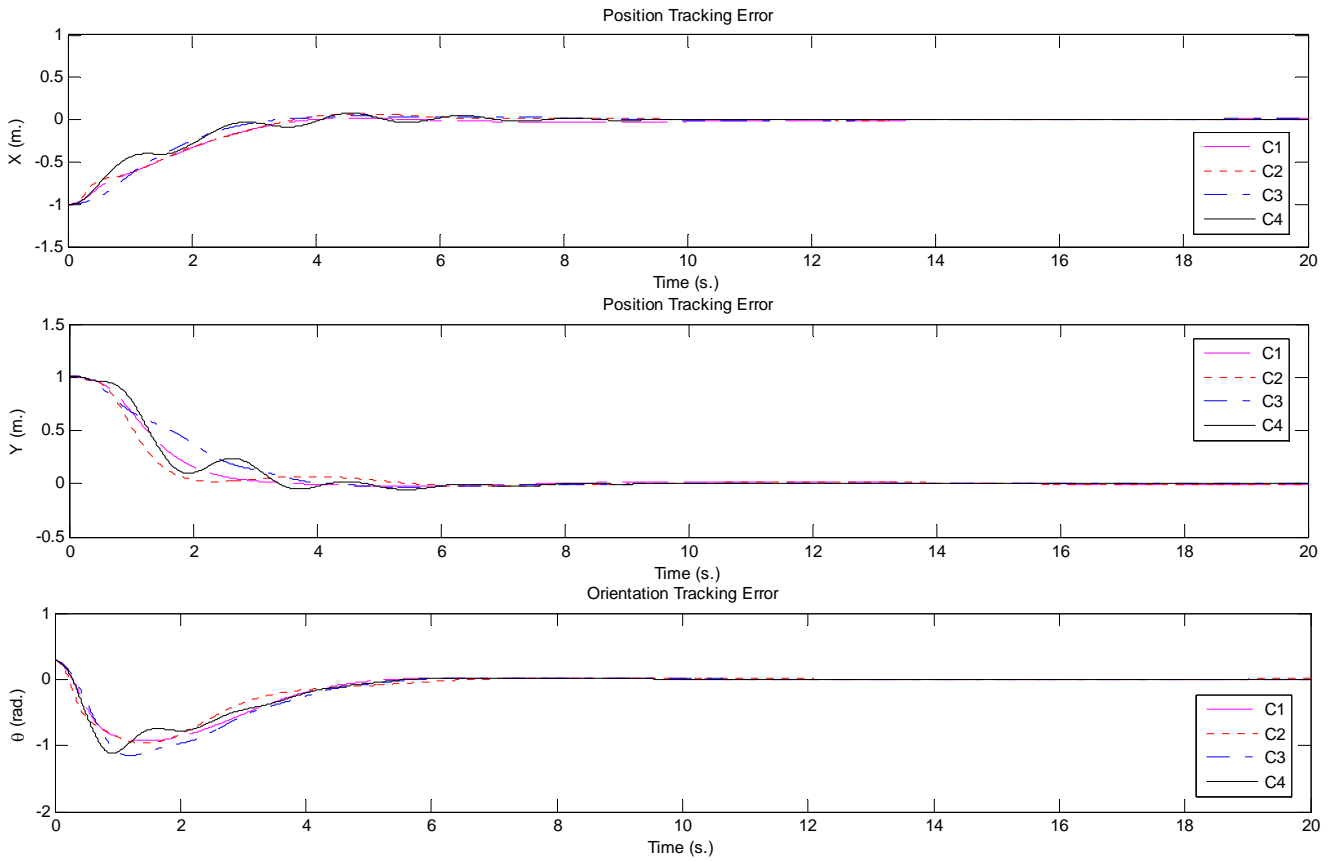


Figure 4: Tracking Errors – PNTCs.

Figure 5 shows the total control torques and in Figure 6 the actual linear and angular velocities of the mobile robot are presented.

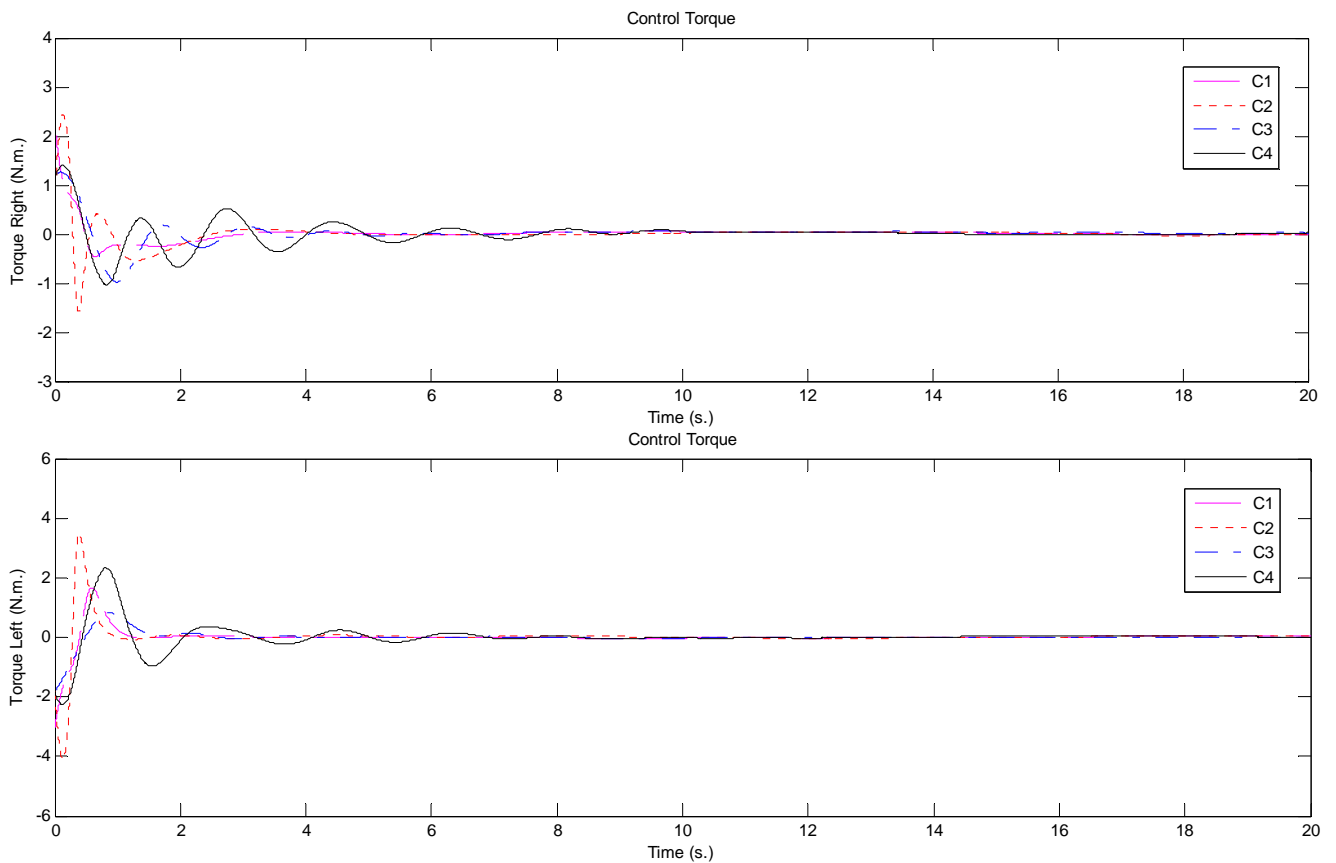


Figure 5: Control Torques – PNTCs.

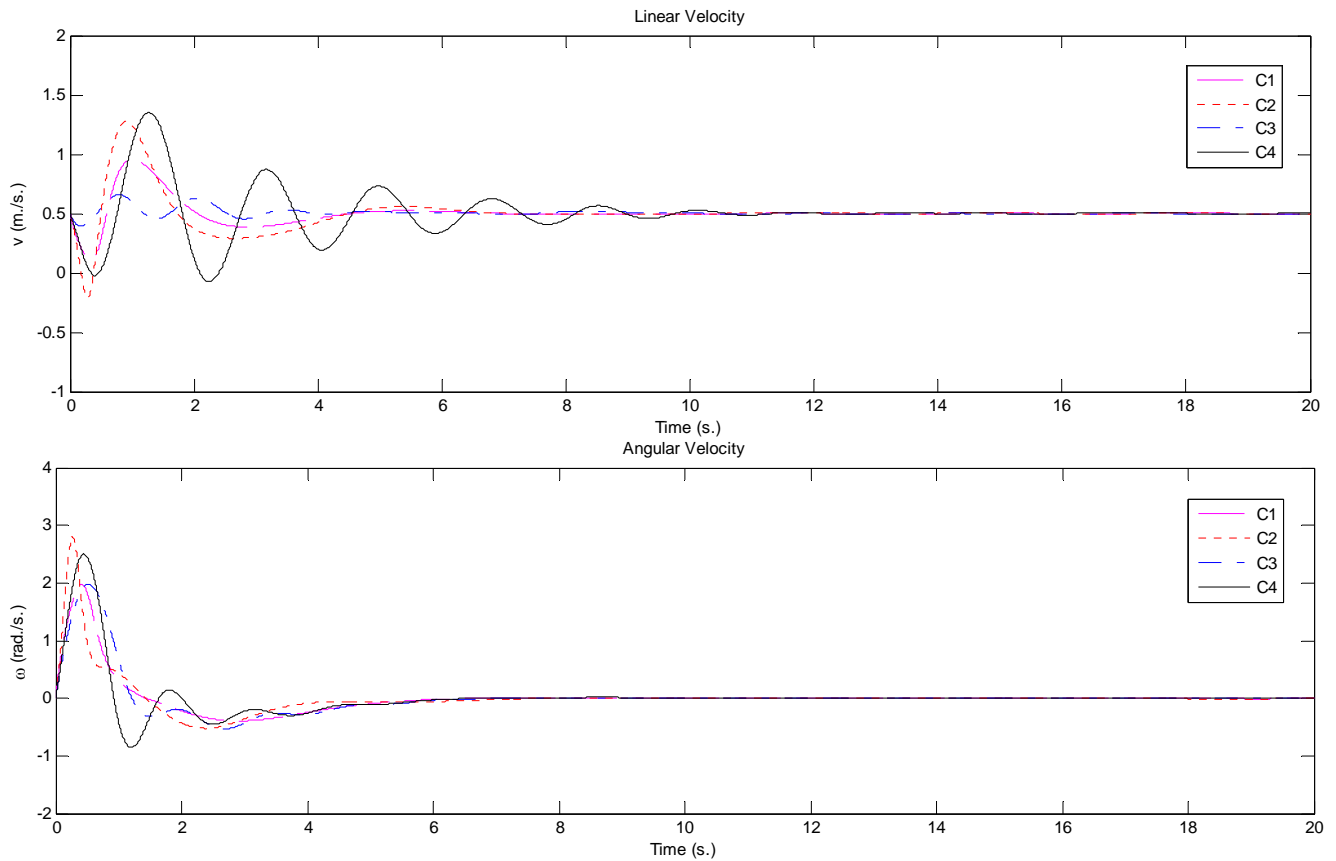


Figure 6: Velocities of the Mobile Robot – PNTCs.

It can be observed that the tracking errors tend to zero, as well as the robot velocities and the control torques converge to its steady state. Moreover, when the sudden change of friction and load variation occurs, is observed that the tracking errors tend to zero, because the PNTCs (C1, C2, C3, and C4) are able to compensate the sudden changes of the robot dynamics through learning mechanism of the RBFNN.

5 CONCLUSIONS

This paper suggests neural control algorithms (PNTCs) for a nonholonomic mobile robot, with a completely unknown robot dynamics, and subject to bounded unknown disturbance including unmodeled and unstructured dynamics.

Since the PNTCs have different structures in terms of input vectors of the RBFNN, the implementation of these controllers is based on the partitioning of the RBFNNs into several smaller subnets in order to obtain more efficient computation, which simplifies the design, gives added controller structure, and also leads to contributes to faster weight tuning algorithms (i. e., individual partitioned neural networks can be separately tuned). Another advantage of this partitioned neural networks is that if some terms in the mobile robots dynamics are well-known (e.g., inertia matrix $\bar{H}(q)$), then their neural networks can be replaced by deterministic equations. These neural networks can be used to reconstruct only the unknown terms or those too complicated to be computed, which will probably include the friction $\bar{F}(v)$ and the Coriolis/centripetal terms $\bar{C}(q, \dot{q})$.

In addition, the values of weights (W , and X) of the RBFNNs are adjusted on-line and all initialised to zero. Also, the C1 is obtained of the modeling by RBFNNs of the centripetal and Coriolis matrix $C(q, \dot{q})$ through of the inertia matrix $H(q)$ of the mobile robot dynamics. Thus, C1 is constituted of static RBFNNs only, what makes possible the reduction of the size of the RBFNNs, of the computational load and the implementation in real time.

The RBFNNs used in the PNTCs neither require an off-line learning nor the priori information of the mobile robot dynamics. Stability and convergence of the robot control system, and the learning algorithm for weights are proved by using Lyapunov theory, considering the presence of bounded unstructured and unmodeled dynamics. The simulation results show the efficiency of the PNTCs, where it is possible to note different dynamic behaviours, since these controllers have

different structures.

As future works, it is validation of the PNTCs in real-time applications of a nonholonomic mobile robot, as well as to realize the integration of the PNTCs with others kinematic controllers of the literature.

6 ACKNOWLEDGEMENTS

The authors would like to acknowledge the CAPES and CNPq by the financial support.

7 REFERENCES

- Fierro, R. and Lewis, F. L. (1998). Control of a Nonholonomic Mobile Robot using Neural Networks. *IEEE Transactions on Neural Networks*, Vol. 9, N° 4, pp. 589-600.
- Fukao, T., Nakagawa, H. and Adachi, N. (2000). Adaptive Tracking Control of a Nonholonomic Mobile Robot. *IEEE Transactions on Robotics and Automation*, Vol. 16, N° 5, pp. 609-615.
- Ge, S. S. (1996). Robust Adaptive NN Feedback Linearization Control of Nonlinear Systems. *International Journal Systems Science*, Vol. 27, N° 12, pp. 1327-1338.
- Hu, T. and Yang, S.X. (2001). An Efficient Neural Controller for a Nonholonomic Mobile Robot. *Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 461-466.
- Kanayama, Y., Kimura, Y., Miyazaki, F. and Noguchi, T. (1990). A Stable Tracking Control Method for an Autonomous Mobile Robot. *Proceedings of the IEEE International Conference on Robotics Automation*, pp. 384-389.
- Narendra, K. S. and Annaswamy, A. M. (1987). A New Adaptive Law for Robust Adaptation without Persistent Excitation. *IEEE Transactions on Automatic Control*, Vol. 32, N° 2, pp. 134-145.
- Oh, C., Kim, M.-S., Lee, J.-Y. and Lee, J.-J. (2003). Control of Mobile Robots using RBF Network. *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 4, N° 3, pp. 3528-3533.
- Poggio, T. and Girosi, F. (1990). Networks for Approximation and Learning. *Proceedings IEEE*, Vol. 78, N° 9, pp. 1481-1497.
- Rice, J. C. (1964). *The Approximation of Functions*. Addison-Wesley, MA - USA.
- Rumelhart, D. E. and J. L. McClelland (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA - USA.
- Sarker, N., Yun, X. and Kumar, V. (1994). Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robot. *International Journal Robotics Research*, Vol. 13, N° 1, pp. 55-69.
- Seshagiri, S. and Khalil, H.K. (2000). Output Feedback Control of Nonlinear Systems using RBF Neural Networks. *IEEE Transactions on Neural Networks*, Vol. 11, N° 1, pp. 69-79.
- Yang, J. M. and Kim, J. H. (1999). Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots. *IEEE Transactions on Robotics and Automation*, Vol. 15, N° 3, pp. 578-587.
- Zhihong, M., Wu, H.R. and Palaniswami, M. (1998). An Adaptive Tracking Controller using Neural Networks for a Class of Nonlinear Systems. *IEEE Transactions on Neural Networks*, Vol. 9, N° 5, pp. 947-955.