MODEL REFERENCE ADAPTIVE CONTROL FOR MOBILE ROBOTS IN TRAJECTORY TRACKING USING RADIAL BASIS FUNCTION NEURAL NETWORKS

F. G. ROSSOMANDO, C. SORIA, D. PATIÑO and R. CARELLI

Instituto de Automática – Universidad Nacional de San Juan
Av. San Martín 1109 Oeste, 5400 San Juan, Argentina
{frossoma; csoria; dpatino; rcarelli}@inaut.unsj.edu.ar

***Abstrac t*— This paper propose an Model Reference Adaptive Control (MRAC) for mobile robots for which stability conditions and performance evaluation are given. The proposed control structure combines a feedback linearization model, based on a kinematics nominal model, and a direct neural network-based adaptive dynamics control.**

The architecture of the dynamic control is based on radial basis functions neural networks (RBF-NN) to construct the MRAC controller. The parameters of the adaptive dynamic controller are adjusted according to a law derived using Lyapunov stability theory and the centers of the RBF are adapted using the supervised algorithm.

The resulting MRAC controller is efficient and robust in the sense that it succeeds to achieve a good tracking performance with a small computational effort. Stability result for the adaptive neuro-control system is given. It is proved that control errors are ultimately bounded as a function of the approximation error of the RBF-NN. Experimental results showing the practical feasibility and performance of the proposed approach to mobile robotics are given.

***Keywords* — system identification, adaptive neural nets, mobile robot control.**

I. INTRODUCTION

Real-time trajectory control of a mobile robot is a very important issue in mobile robotics. Due to slippage, disturbances, noise, robot-base interaction and sensor errors, it is very difficulty to reduce the errors between the desired and real robot position. How to effectively control a mobile robot to precisely track a desired trajectory is still an open question in robotics.

Several studies have been published regarding the design of controllers to guide mobile robots during trajectory tracking. Most of the controllers designed so far are based only on the kinematics of the mobile robot, like the controllers presented in Carelli *et al.* (1999), in Wu *et al.* (1999) and in Künhe *et al.* (2005). To perform tasks requiring high speed movements and/or heavy load transportation, it is important to consider the robot dynamics, besides its kinematics. No matter the uncertainties or changes in its dynamics, the tasks must be performed with due precision. As an example, in the case of load transportation, the dynamic characteristics such as mass, center of mass and inertia, change when the robot is loaded. Then, to keep a good performance, the controller should be capable of adapting itself to this kind of changes. This adaptive capability is also important whenever it is difficult to model the system exactly, even without dynamic changes from task to task. Some works present the design of controllers that compensate for the robot dynamics. Fukao *et al.* (2000) propose the design of an adaptive trajectory tracking controller to generate torques based on a dynamic model whose parameters are unknown. In this work, only simulation results are shown. Other types of trajectory tracking controllers assuming uncertainty in the robot dynamics are developed by Liu *et al.* (2004), Dong and Guo (2005) and Dong and Huo (1999), with the performance shown just through simulations. Das and Kar (2006) show an adaptive fuzzy logic-based controller where the system uncertainty, which includes mobile robot parameters variation and unknown nonlinearities, is estimated by a fuzzy logic system and its parameters are tuned on-line. Bugeja and Fabri (2007) present the use of a *RBF-NN* for mobile robot dynamics approximation, in which the centroids remain fixed and the weights are estimated stochastically in real-time. The authors show simulation results.

Kim *et al*. (2000) have proposed a robust adaptive controller for a mobile robot divided in two parts. The first one is based on robot kinematics and is responsible of generating references for the second one, which compensates for the modeled dynamics. However, the adapted parameters are not real parameters of the robot, and no experimental results are presented. Additionally, the control actions are given in terms of torques, while usual commercial robots accept velocity commands. In (De la Cruz and Carelli, 2006) it is presented a linear parameterization of a unicycle-like mobile robot and the design of a trajectory tracking controller based on its complete known model. One advantage of their controller is that its parameters are directly related to the robot parameters. However, if the parameters are not correctly identified or change with time due, for example, to load variation, the performance of the controller will be severely affected.

In this paper, the design of an adaptive trajectory tracking controller based on a nominal robot dynamics and neural controller is developed. The whole control system is designed in two parts: one including a kinematics controller and another one with a dynamics controller, similar to the control structure in Kim *et al.* (2000). As a realistic assumption, it is supposed that model uncertainties may appear in the robot dynamics alone. Therefore, the neuro-controller is designed based on *RBF-NN* with supervised and self organized selection of centers (fast k-means). An analysis of trajectory control errors for the MIMO robot system is presented as a function of the approximation *NN* errors.

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

II. ROBOT MODEL

A. System overview

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

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

Kinematics model

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

Dynamic model

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\vartheta_3}{\vartheta_1}\omega^2 - \frac{\vartheta_4}{\vartheta_1}v \\ -\frac{\vartheta_5}{\vartheta_2}v\omega - \frac{\vartheta_6}{\vartheta_2}\omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\vartheta_1} & 0 \\ 0 & \frac{1}{\vartheta_2} \end{bmatrix} \begin{bmatrix} v^d_{ref} \\ \omega^d_{ref} \end{bmatrix} + \begin{bmatrix} \delta_v \\ \delta_\omega \end{bmatrix} \tag{2}$$

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

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

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

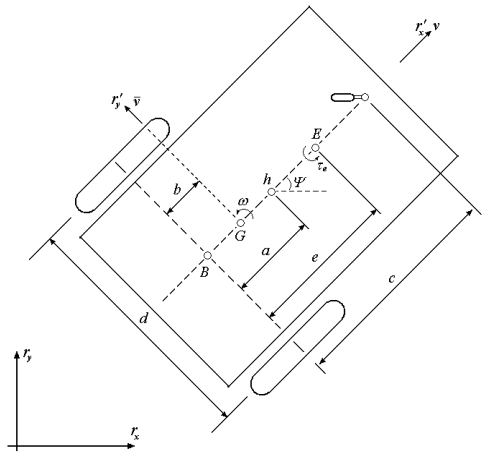

Figure 1. Parameters of the unicycle-like mobile robot

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

III. KINEMATICS CONTROLLER

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

$$\begin{bmatrix} v^c_{ref} \\ \omega^c_{ref} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\frac{1}{a}\sin\psi & \frac{1}{a}\cos\psi \end{bmatrix} \begin{bmatrix} \dot{r}_{xd} + l_x \tanh\left(\frac{k_x}{l_x}\tilde{r}_x\right) \\ \dot{r}_{yd} + l_y \tanh\left(\frac{k_y}{l_y}\tilde{r}_y\right) \end{bmatrix}, \tag{4}$$

where $\begin{bmatrix} \dot{r}_{xd} & \dot{r}_{yd} \end{bmatrix}^T$ are the desire velocities , $\tilde{r}_x = r_{xd} - r_x$ and $\tilde{r}_y = r_{yd} - r_y$ are the trajectory errors.

By replacing Eq. (4) in the upper part of Eq. (1) under the assumption of perfect velocity tracking ($v = v^c_{ref}$, $\omega = \omega^c_{ref}$), the closed-loop equation is,

$$\begin{bmatrix} \dot{\tilde{r}}_x \\ \dot{\tilde{r}}_y \end{bmatrix} + \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_x}{l_x}\tilde{r}_x\right) \\ \tanh\left(\frac{k_y}{l_y}\tilde{r}_y\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5}$$

Defining the output error vector $\tilde{\mathbf{h}} = \begin{bmatrix} \tilde{r}_x & \tilde{r}_y \end{bmatrix}^T$, Equation (5) can be written as

$$\dot{\tilde{\mathbf{h}}} = -\begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x}\tilde{r}_x\right) & l_y \tanh\left(\frac{k_y}{l_y}\tilde{r}_y\right) \end{bmatrix}^T. \tag{6}$$

Which implies that $h \to 0$ when $t \to \infty$. The perfect velocity tracking assumption will be relaxed when analysing the stability of the whole control system.

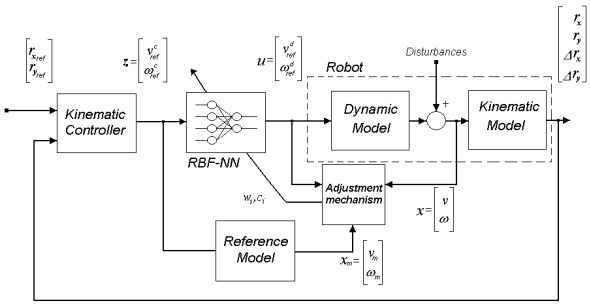

Figure 2. Control structure

IV. DYNAMIC CONTROLLER

A. Problem formulation

The dynamic controller, implemented as a NN, receives the references of linear and angular velocities which are generated by the kinematics controller $\begin{bmatrix} v_{ref}^c & \omega_{ref}^c \end{bmatrix}^T$, and produces another pair of linear and angular velocities commands to be sent to the robot servos $\begin{bmatrix} v_{ref}^d & \omega_{ref}^d \end{bmatrix}^T$, as shown in Figure 2

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}\mathbf{u}\,, \tag{7}$$

where

$$\mathbf{x} = \begin{bmatrix} v & \omega \end{bmatrix}^T \qquad \mathbf{u} = \begin{bmatrix} v_{ref}^d & \omega_{ref}^d \end{bmatrix}^T$$

are the state variables and the control action, respectively, and the reference input from kinematics controller is

$$\mathbf{z} = \begin{bmatrix} v_{ref}^c & \omega_{ref}^c \end{bmatrix}^T$$

The robot dynamics system (7) can be expressed as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\vartheta_3}{\vartheta_1}\omega^2 - \frac{\vartheta_4}{\vartheta_1}v \\ -\frac{\vartheta_5}{\vartheta_2}v\omega - \frac{\vartheta_6}{\vartheta_2}\omega \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} \frac{1}{\vartheta_1} & 0 \\ 0 & \frac{1}{\vartheta_2} \end{bmatrix}$$

It is assume that the state vector $\mathbf{x} \in \Re^n$ is available for measurement.

The error is defined by

$$\mathbf{e} = \mathbf{x}_m - \mathbf{x} \tag{8}$$

where $\boldsymbol{x}$ is the state vector and $\mathbf{i}_m$ is the desired state vector from a reference model.

B. The RBF adaptive controller

The control objective is to determine a state feedback control $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{w},\boldsymbol{\xi})$ based on Radial Basis Function (RBF) network, and an adaptive law for adjusting the parameter vector $\boldsymbol{w}$ of the network such that the tracking error is small as possible. In order to achieve these objectives, a method for a direct adaptive controller based on RBF network is developed. As in the main trend of neural network adaptive control, an ideal control law $\boldsymbol{u}(t)$ based on feedback linearization and the functions $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}$ is proposed.

An RBF-NN function is defined by

$$F(\mathbf{x}) = \sum_{i=1}^{M} \xi_i . w_i = \mathbf{w}^T . \xi(\mathbf{x}) \qquad \xi_i = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) \tag{9}$$

where $\boldsymbol{x}$ is the input vector, φ is a non linear function called radial basis function, w_i are connections weights (parameters) between the hidden layer and the output layer, c_i are centres of basis functions, M is the number of basis functions. An RBF network with only one output is considered here. The most used basis function is the Gaussian function.

$$\varphi(r) = \exp\left(\frac{r^2}{2.\sigma^2}\right), \tag{10}$$

with $r = \|\mathbf{x} - \mathbf{c}_i\|$, c_i is the centre of $\varphi(r)$, σ is an associated constant to the function $\varphi(r)$ and represents the width (spread) of the Gaussian function (in this case $\sigma=1$).

The *RBF* network can be considered as a two-layer network in which the hidden layer performs a fixed nonlinear transformation to map the input space into an intermediate space, then the output layer combines the outputs of the intermediate layer linearly as the outputs of the whole network. High order differential equations can be represented by *RBF-NN*. The Eq. (7) is represented by:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{w}_f^{*T}\xi(\mathbf{x}) + \mathbf{w}_g^{*T}\mathbf{u} + \\ &+ \left[(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}) - \left(\mathbf{F}(\mathbf{x}/\mathbf{w}_f^*) + \mathbf{G}(\mathbf{w}_g^*)\mathbf{u}\right)\right] \end{aligned} \tag{11}$$

where $\mathbf{w}^*$ and $\boldsymbol{\xi}^*$ are the optimal parameter vector corresponding to the optimal approximator control signal $\mathbf{u}(\mathbf{x},\mathbf{w}^*,\boldsymbol{\xi})$ of the ideal control signal $(\mathbf{u}^*)$.

C. Stability and weights adaptation

In the following the adaptation law for the connections weights and centers of the basis functions using Lyapunov synthesis approach is derived. Defining the approximation error

$$\Delta = \left(\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}\right) - \left(\mathbf{F}(\mathbf{x}/\mathbf{w}_f^*) + \mathbf{G}(\mathbf{w}_g^*)\mathbf{u}\right), \tag{12}$$

equation (11) can be written as,

$$\dot{\mathbf{x}} = \mathbf{w}_f^{*T}\xi(\mathbf{x}) + \mathbf{w}_g^{*T}\mathbf{u} + \Delta\,. \tag{13}$$

Now, defining the errors $\tilde{\mathbf{w}}_f = \mathbf{w}_f^* - \mathbf{w}_f$ and $\tilde{\mathbf{w}}_g = \mathbf{w}_g^* - \mathbf{w}_g$, $\xi(\mathbf{x})$ is a bounded funtion, (13) can be expressed as:

$$\dot{\mathbf{x}} = \mathbf{w}_f^T\xi(\mathbf{x}) + \tilde{\mathbf{w}}_f^T\xi(\mathbf{x}) + \tilde{\mathbf{w}}_g^T\mathbf{u} + \mathbf{w}_g^T\mathbf{u} + \Delta\,. \tag{14}$$

The reference model is:

$$\dot{\mathbf{x}}_m = \mathbf{A}_m.\mathbf{x}_m + \mathbf{B}_m.\mathbf{u}_m\,, \tag{15}$$

where $\mathbf{A}_m$ is a $\Re^{2x2}$ Hurwitz matrix, $\mathbf{B}_m$ is a known matrix, and $\mathbf{u}_m$ is a two-dimensional input vector with bounded elements.

The control error from (8) is defined by:

$$\mathbf{e} = \mathbf{x}_m - \mathbf{x} \rightarrow \dot{\mathbf{e}} = \dot{\mathbf{x}}_m - \dot{\mathbf{x}}\,. \tag{16}$$

Replacing (14) and (15) in (16):

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{A}_m\mathbf{e} + \mathbf{A}_m\mathbf{x} + \mathbf{B}_m\mathbf{u}_m - \mathbf{w}_f^T\xi(\mathbf{x}) - \\ &- \tilde{\mathbf{w}}_f^T\xi(\mathbf{x}) - \tilde{\mathbf{w}}_g^T\mathbf{u} - \mathbf{w}_g^T\mathbf{u} - \Delta \end{aligned} \tag{17}$$

The control action is selected as:

$$\mathbf{u} = \left(\mathbf{w}_g^T\right)^{-1}\left[\mathbf{A}_m\mathbf{x} + \mathbf{B}_m\mathbf{u}_m - \mathbf{w}_f^T\boldsymbol{\xi}(\mathbf{x})\right]. \tag{18}$$

The stability of the system is studied in order to develop an adaptive law to adjust the parameter vectors $\boldsymbol{w}$ and $\boldsymbol{\xi}$ of the RBF controller. Replacing $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{w},\boldsymbol{\xi})$ of the RBF system (18), in the penultimate term of (17), the error equation can be rewritten as

$$\dot{\mathbf{e}} = \mathbf{A}_m\mathbf{e} - \mathbf{w}_f^T\xi(\mathbf{x}) - \tilde{\mathbf{w}}_g^T\mathbf{u} - \Delta\,. \tag{19}$$

The term $\mathbf{w}_f^T\xi(\mathbf{x})$ involving a product of two errors, is now considered included in Δ. Taking into account that

$$\mathbf{e} = \begin{bmatrix} e_v & e_\omega \end{bmatrix}^T \quad \mathbf{A}_m = \begin{bmatrix} a_v & 0 \\ 0 & a_\omega \end{bmatrix}$$

$$\mathbf{w}_g = \begin{bmatrix} \mathbf{w}_{gv} & \mathbf{w}_{g\omega} \end{bmatrix} \quad \mathbf{w}_f = \begin{bmatrix} \mathbf{w}_{fv} & \mathbf{w}_{f\omega} \end{bmatrix} \quad \Delta = \begin{bmatrix} \Delta_v & \Delta_\omega \end{bmatrix}^T$$

the error equation can be rewritten in two components

$$\begin{cases} \dot{e}_v = a_v e_v - \tilde{\mathbf{w}}_{fv}^T\xi(x) - \tilde{\mathbf{w}}_{gv}^T\mathbf{u} - \Delta_v \\ \dot{e}_\omega = a_\omega e_\omega - \tilde{\mathbf{w}}_{f\omega}^T\xi(x) - \tilde{\mathbf{w}}_{g\omega}^T\mathbf{u} - \Delta_\omega \end{cases} \tag{20}$$

Defining the Lyapunov function candidate:

$$V = \frac{1}{2}p_v e_v^2 + \frac{1}{2}p_\omega e_\omega^2 + \frac{1}{2.\gamma_1}\tilde{\mathbf{w}}_{fv}^T\tilde{\mathbf{w}}_{fv} + \frac{1}{2.\gamma_2}\tilde{\mathbf{w}}_{f\omega}^T\tilde{\mathbf{w}}_{f\omega} + \\ + \frac{1}{2.\gamma_3}\tilde{\mathbf{w}}_{gv}^T\tilde{\mathbf{w}}_{gv} + \frac{1}{2.\gamma_4}\tilde{\mathbf{w}}_{g\omega}^T\tilde{\mathbf{w}}_{g\omega} \tag{21}$$

$\gamma_{1,2,3,4}$ are positive constants and $\mathbf{P}$ is a solution of the Lyapunov equation :

$$\mathbf{A}_m^T.\mathbf{P} + \mathbf{P}.\mathbf{A}_m = -\mathbf{Q} \qquad \mathbf{Q} > 0\,, \tag{22}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are

$$\mathbf{P} = \begin{bmatrix} p_v & 0 \\ 0 & p_\omega \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} q_v & 0 \\ 0 & q_\omega \end{bmatrix}.$$

Differentiate V with respect to time:

$$\dot{V} = p_v e_v \dot{e}_v + p_\omega e_\omega \dot{e}_\omega + \\ + \frac{1}{2.\gamma_1}\dot{\tilde{\mathbf{w}}}_{fv}^T\tilde{\mathbf{w}}_{fv} + \frac{1}{2.\gamma_1}\tilde{\mathbf{w}}_{fv}^T\dot{\tilde{\mathbf{w}}}_{fv} + \frac{1}{2.\gamma_2}\dot{\tilde{\mathbf{w}}}_{f\omega}^T\tilde{\mathbf{w}}_{f\omega} + \frac{1}{2.\gamma_2}\tilde{\mathbf{w}}_{f\omega}^T\dot{\tilde{\mathbf{w}}}_{f\omega} + \\ + \frac{1}{2.\gamma_3}\dot{\tilde{\mathbf{w}}}_{gv}^T\tilde{\mathbf{w}}_{gv} + \frac{1}{2.\gamma_3}\tilde{\mathbf{w}}_{gv}^T\dot{\tilde{\mathbf{w}}}_{gv} + \frac{1}{2.\gamma_4}\dot{\tilde{\mathbf{w}}}_{g\omega}^T\tilde{\mathbf{w}}_{g\omega} + \frac{1}{2.\gamma_4}\tilde{\mathbf{w}}_{g\omega}^T\dot{\tilde{\mathbf{w}}}_{g\omega}. \tag{23}$$

Substituing (20) in (23),

$$\dot{V} = -\frac{q_v e_v^2}{2} - e_v^T p_v \tilde{\mathbf{w}}_{fv}^T\xi(\mathbf{x}) - e_v p_v \mathbf{w}_{fv}^T\tilde{\xi}(\mathbf{x}) - e_v p_v \tilde{\mathbf{w}}_{gv}^T\mathbf{u} - \\ - \frac{q_\omega e_\omega^2}{2} - e_\omega^T p_\omega \tilde{\mathbf{w}}_{f\omega}^T\xi(\mathbf{x}) - e_\omega p_\omega \mathbf{w}_{f\omega}^T\tilde{\xi}(\mathbf{x}) - e_\omega^T p_\omega \tilde{\mathbf{w}}_{g\omega}^T\mathbf{u} + \\ + \frac{\tilde{\mathbf{w}}_{fv}^T\dot{\tilde{\mathbf{w}}}_{fv}}{\gamma_1} + \frac{\tilde{\mathbf{w}}_{f\omega}^T\dot{\tilde{\mathbf{w}}}_{f\omega}}{\gamma_2} + \frac{\tilde{\mathbf{w}}_{gv}^T\dot{\tilde{\mathbf{w}}}_{gv}}{\gamma_3} + \frac{\tilde{\mathbf{w}}_{g\omega}^T\dot{\tilde{\mathbf{w}}}_{g\omega}}{\gamma_4} + \frac{\tilde{\xi}^T(\mathbf{x})\tilde{\xi}(\mathbf{x})}{\gamma_5} \tag{24}$$

where $\rho = -\left(e_v^T.p_v.\Delta_v + e_\omega^T.p_\omega.\Delta_\omega\right)$. Equation (24) can be rewritten as,

$$\dot{V} = -\tilde{\mathbf{w}}_{fv}^T\left(e_v^T p_v\xi(\mathbf{x}) - \frac{\dot{\tilde{\mathbf{w}}}_{fv}}{\gamma_1}\right) - \tilde{\mathbf{w}}_{f\omega}^T\left(e_\omega^T p_\omega\xi(\mathbf{x}) - \frac{\dot{\tilde{\mathbf{w}}}_{f\omega}}{\gamma_2}\right) - \\ - \tilde{\mathbf{w}}_{gv}^T\left(e_v^T p_v\mathbf{u} - \frac{\dot{\tilde{\mathbf{w}}}_{gv}}{\gamma_3}\right) - \tilde{\mathbf{w}}_{g\omega}^T\left(e_\omega^T p_\omega\mathbf{u} - \frac{\dot{\tilde{\mathbf{w}}}_{g\omega}}{\gamma_4}\right) - \\ - \frac{e_v^T q_v e_v}{2} - \frac{e_\omega^T q_\omega e_\omega}{2} + \rho \tag{25}$$

In order to make $\dot{V} < 0$, it is setting the nonnegative terms of $\dot{V}$ equal to zero, and recalling that $\dot{\tilde{\mathbf{w}}}_{f,g} = \dot{\mathbf{w}}_{f,g} - \dot{\mathbf{w}}_{f,g}^* = \dot{\mathbf{w}}_{f,g}$, the adaptation laws are obtained:

$$\begin{aligned} \dot{\mathbf{w}}_{fv} &= \gamma_1 e_v^T p_v\xi(\mathbf{x}) \\ \dot{\mathbf{w}}_{f\omega} &= \gamma_2 e_\omega^T p_\omega\xi(\mathbf{x}) \\ \dot{\mathbf{w}}_{g\omega} &= \gamma_3 e_\omega^T p_\omega\mathbf{u} \\ \dot{\mathbf{w}}_{gv} &= \gamma_4 e_v^T p_v\mathbf{u} \end{aligned} \tag{26}$$

Now, $\dot{V} = -\frac{e_v^2 q_v}{2} - \frac{e_\omega^2 q_\omega}{2} + \rho$. In order to make $\dot{V} < 0$, it should be verified that

$$\frac{e_v^2 q_v}{2} + \frac{e_\omega^2 q_\omega}{2} \geq \rho \tag{27}$$

A sufficient condition is

$$\left\|q_v e_v^2\right\| + \left\|q_\omega e_\omega^2\right\| \geq 2\left\|\rho\right\| \tag{28}$$

which is verified if

$$\lambda_{\min}\left(q_v\right)\left\|e_v\right\|^2 + \lambda_{\min}\left(q_\omega\right)\left\|e_\omega\right\|^2 \geq 2\left\|\rho\right\| \tag{29}$$

Then, a sufficient condition for $\dot{V} < 0$ is expressed as,

$$\min\left(\left\|e_v\right\|^2, \left\|e_\omega\right\|^2\right) \geq \frac{2\left\|\rho\right\|}{\min\left[\lambda_{\min}\left(q_v\right), \lambda_{\min}\left(q_\omega\right)\right]} \tag{30}$$

This condition implies that the control errors norms are ultimately bounded by the value

$$B_{\delta_e} = \frac{2\left\|\rho\right\|}{\min\left[\lambda_{\min}\left(q_v\right), \lambda_{\min}\left(q_\omega\right)\right]}$$

This is a practical result allowing to state that the control error is bounded in terms of the *NN* approximation error.

D. Training and center placement in an RBF network

Usually the training procedure for *RBF* networks is divided in two stages: the training for the centers adjustment of basis functions in the hidden layer, followed by the training for the connections weights adjustment between the output layer and the hidden layer. However, in control applications, online training is concerned with only the connections weights between the hidden and output layer, and the centers are set off line. In this work, we consider to adjust online the centers of the basis functions (self organized or supervised) and the connections weights, according to (26) (Haykin, 1994).

The fast k-means algorithm (self organized training method) and gradient descendent (seupervised method) are often used respectively for centers adjustment. The best computational cost for applications in mobile robots is obtained.

Position of centers (hidden layer) can be expressed by:

$$\mathbf{c}_i(k+1) = \mathbf{c}_i(k) + \alpha\frac{\partial J(k)}{\partial \mathbf{c}_i(k)}, \\ J(k) = \frac{1}{2}\left(\varepsilon^T(k)\varepsilon(k)\right) = \frac{1}{2}\left[\left(\mathbf{z}(k) - \mathbf{x}(k)\right)^T\left(\mathbf{z}(k) - \mathbf{x}(k)\right)\right], \tag{31}$$

where $J(k)$ is the cost function, and now expressing the Eq. (13) in discrete time and integrating, it is obtain:

$$\mathbf{x}(k) = \sum_0^N\left(\mathbf{w}_f^{*T}\xi(\mathbf{x}(k)) + \mathbf{w}_g^{*T}\mathbf{u}(t)\right)T_0\,, \tag{32}$$

where T_0=0.1 sec. is the sample time and k is the step time.

Applying the chain rule in $J(k)$ it is have:

$$\frac{\partial J(k)}{\partial \mathbf{c}_i(k)} = \frac{\partial J(k)}{\partial \mathbf{x}(k)} \cdot \frac{\partial \mathbf{x}(k)}{\partial \mathbf{u}(k)} \cdot \frac{\partial \mathbf{u}(k)}{\partial \mathbf{c}_i(k)}, \tag{33}$$

where

$$\frac{\partial \mathbf{x}(k)}{\partial \mathbf{c}_i(k)} = \left(\sum_0^N \left(\mathbf{w}_f^{*T} \xi(\mathbf{x}(k)) \right) \right) . \left(\mathbf{x}(k) - \mathbf{c}_i(k) \right) \tag{34}$$

Hence to effectively use these gradients, we need to know ∂x/∂u, which is difficult to calculate when system model is unknown. However, we may use an approximation:

$$\frac{\partial \mathbf{x}(k)}{\partial \mathbf{u}(k)} = sign\left(\frac{\Delta \mathbf{x}(k)}{\Delta \mathbf{u}(k)} \right) \tag{35}$$

Substituing (33), (34) and (35) in (31)

$$\mathbf{c}_i(k+1) = \mathbf{c}_i(k) - \alpha \varepsilon(k) . \left(sign\left(\frac{\Delta \mathbf{x}(k)}{\Delta \mathbf{u}(k)} \right) \right) \sum_0^N \left(\mathbf{w}_f^{*T} \xi(\mathbf{x}(k)) \right) . \left(\mathbf{x}(k) - \mathbf{c}_i(k) \right) \tag{36}$$

The constant T is considered into α.

This is a practical result allowing adjusting the centers of a *RBF-NN*.

E. Design of the MRAC-RBF controller

The design of the MRAC-RBF controller can be summarized in the following steps.

Step 1. Off-line computations

-Define the number of basis functions.

-Specify the parameters $a_{v,\omega}$ such that all eigenvalues of matrix $\boldsymbol{A_m}$, are in the open left-half plane.

-Specify a positive definite *nxn* matrix $\boldsymbol{Q}$.

-Solve the Lyapunov Eq. (22) to obtain a symmetric $\boldsymbol{P}>0$.

Step 2. On-line adaptation

- Apply the feedback control (18) to the plant (7).
- Use the adaptive law (26) to adjust the connections weights and the supervised algorithm to adjust the centers of the radial basis functions.

V. EXPERIMENTAL RESULTS

To show the performance of the proposed controller several experiments and simulations were executed and some of the results are presented in this section. The proposed controller was implemented on a Pioneer 2DX mobile robot (Fig. 3), which admits linear and angular velocities as input reference signals. The Pioneer2DX has an 800MHz Pentium III with 512Mb ram onboard computer in which was programmed the controller. To sensing the robot position, it is used the odometric sensors.

$$\begin{cases} r_x = 0.75 \sin(0.03\pi t) \\ r_y = 0.75 \cos(0.03\pi t) \end{cases} \tag{37}$$

The *RBF-NN* controller was initialized with random dynamic parameters. In the experiment, the robot starts at r_x =0.0 m and r_y =0.0 m, and must follow a circular trajectory reference (Eq. 37). The center of the reference circle is at r_x =0.0 m and r_y =0.0 m. The reference trajectory starts at r_x = 0.75 m and r_y = 0.75 m, and follows a circle with a radius of 0.75 m. After 75 seconds, the reference trajectory suddenly changes to a circle of radius 0.325 m.

Figure 3. The mobile robot Pioneer 2DX.

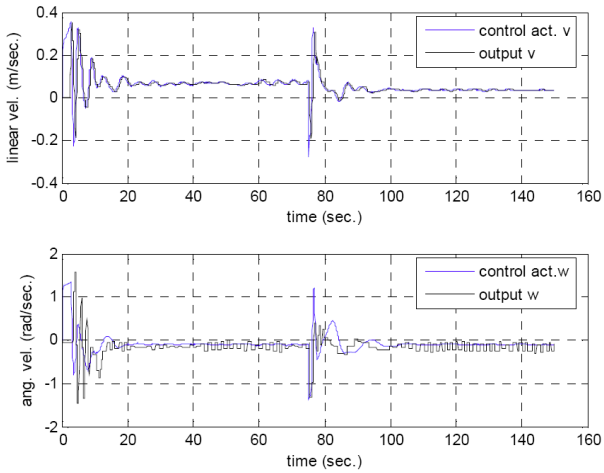

Figure 4. Angular and linear velocities output and control actions of RBF adaptive controller.

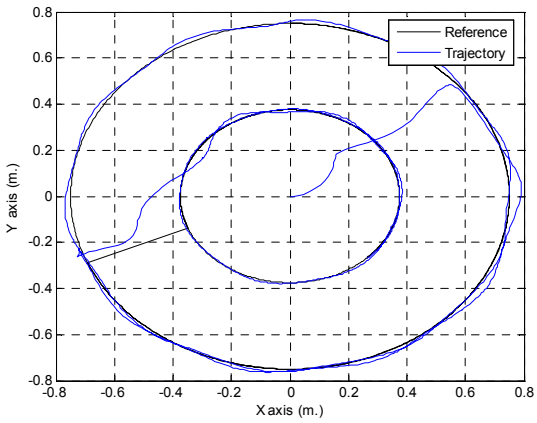

Figure 5. Reference and actual trajectory with the adaptive neural controller.

The Fig. 4 depicts the speeds and control actions of RBF adaptive controller and outputs velocities. The trajectory followed by the mobile robot with neural adaptive controller is shown Fig. 5. The Fig. 6 shows the distance errors for experiments using the proposed *RBF* controller to follow the desired reference trajectory. The distance error is defined as the instantaneous distance between the reference and the robot position. Notice the high initial error, which is due to the fact that the reference trajectory starts on a point that is far from the initial robot position.

There are different training strategies (supervised and self organized) to obtain the *RBF* centre's, the objective is obtain the best computational cost for robotics applications.

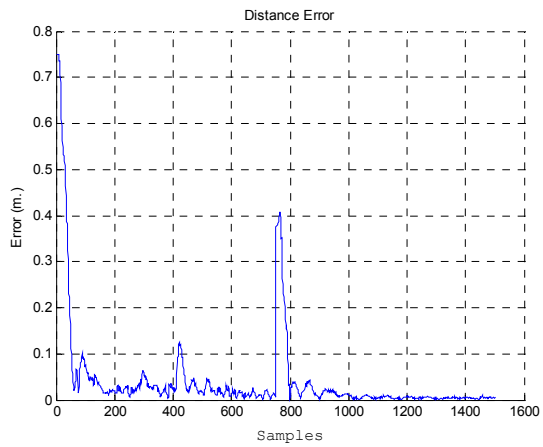

Figure 6. Evolution of the trajectory error.

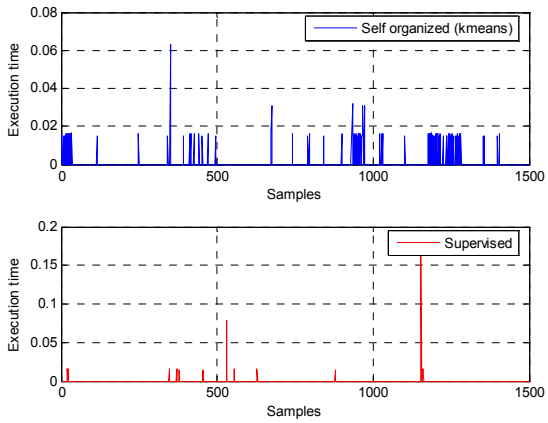

Figure 7: Computational cost between a) self organized and b) supervised selection of centers.

The evolution of computational cost estimates along time (see Fig. 7a and 7b) shows that supervised algorithm smaller than self organized algorithm.

VI. CONCLUSIONS

This paper has presented an MRAC control system for trajectory tracking in mobile robots, using RBF neural networks. The stability conditions and performance evaluation has been included.

The connections weights adjustment laws are derived from the Lyapunov-based stability analysis. The proposed neuro-controller with supervised selection of centers is a good solution in applications requiring fast and precise action, such as tracking and trajectory control. This particular scheme, based on *RBF-NNs*, is computationally more efficient than the case of using the learning capabilities of the neural network to be adapted, as that used in feedback structures that need to propagate back control errors through the model (or network model) to adjust the neurocontroller. To show the practical feasibility and performance of the *NN*-based adaptive control algorithms as well as stability properties obtained in the present work, an experimentation studies was carried out for a Pioneer 2DX mobile robot. The results show the practical feasibility and good performance of the proposed approach to mobile robotics.

REFERENCES

Bugeja, M.K. and S.G. Fabri, "Dual Adaptive Control for Trajectory Tracking of Mobile Robots," *IEEE ICRA,* Roma, Italy, 10-14 April (2007).

Carelli, R., H. Secchi and V. Mut, "Algorithms for Stable Control of Mobile Robots with Obstacle Avoidance," *Latin American Applied Research*, **29**, 191-196 (1999).

Das, T. and I.N. Kar, "Design and Implementation of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots," *IEEE Transactions on Control Systems Technology*, **14**, 501-510 (2006).

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

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

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

Fukao, T., H. Nakagawa and N. Adachi, "Adaptive Tracking Control of a Mobile Robot," *IEEE Transaction on Robotics and Automation*, **16**, 609-615 (2000).

Haykin, S., *Neural Networks, A Comprehensive Foundation,* Prentice Hall, (1994).

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

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

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

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

Received: October 2, 2008.
Accepted: April 14, 2010.
Recommended by Subject Editor José Guivant.