CEAI, Vol. 4, No. 3, pp. 25-32, 2002 Printed in Romania

AN ADAPTIVE TRACKING CONTROLLER FOR A MOBILE ROBOT USING NEURAL NETWORKS

Emil PETRE

University of Craiova, Department of Automatic Control

E-mail: epetre@automation.ucv.ro

***Abstract**: In this paper a design procedure for an adaptive tracking controller for a mobile robot subject to kinematic constraints is presented. The dynamics of the mobile robot is assumed to be completely unknown, and is on-line identified using neural network based estimators. Both the form of the controller and the adaptation laws of neural network weights are derived from a Lyapunov analysis of stability. Under certain conditions, the tracking stability of the closed loop system, and the convergence of the neural network weight updating process are guaranteed. No preliminary learning stage of neural network weights is required. Computer simulations conducted in the case of a mobile robot with two independently actuated wheels are included to demonstrate the performances of this neural network controller.*

***Keywords:** Non-linear systems, Mobile robot, Adaptive tracking control, Neural networks.*

1. INTRODUCTION

In the recent years, the artificial neural networks, with their strong learning capability, have proven to be suitable tool for controlling complex non-linear dynamic systems [1], [6], [7], [8]. The basic idea behind the neural network (NN) based control is to use a neural network estimator to identify the unknown non-linear dynamics and compensate for it. Also, the neural network based approach can deal with the control of non-linear systems that may not be linearly parameterizable, as required in many adaptive approaches.

With regard to neural networks it must be noted, they have been widely adopted in the modelling and control of robotic manipulators [5].

In this paper we present a design procedure for the motion control of a mobile robot subject to kinematic constraints. The dynamics of the mobile robot is assumed to be completely unknown, and is on-line identified using neural network based estimators. Both the form of the controller and the adaptation law of neural network weights are derived from a Lyapunov analysis of stability. Under certain conditions, the tracking stability of the closed loop system, and the convergence of the neural network weight updating process are guaranteed. No preliminary learning stage of neural network weights is required.

Computer simulations conducted in the case of a mobile robot with two independently actuated wheels are included to demonstrate the performances of this neural network controller.

The rest of this paper is organized as follows.

Section 2 is devoted to kinematics and dynamics modelling of a mobile robot. A design procedure of a neural network adaptive tracking controller for a mobile robot is proposed in Section 3. The performances of the proposed control algorithm are presented in Section 4 and, finally, Section 5 concludes the paper.

2. KINEMATICS AND DYNAMICS OF A MOBILE ROBOT

The dynamics of a mobile robot subject to kinematic constrains has the form [2], [5]:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G(q,\dot{q}) + A^T(q)\lambda + \delta = B(q)\tau \qquad (1)$$

where $q \in \Re^n$ are generalized coordinates, $\tau \in \Re^p$ is the torque input vector, $\lambda \in \Re^m$ is the vector of constraint forces, $M(q) \in \Re^{n\times n}$ is a symmetric and positive definite inertia matrix, $V(q,\dot{q}) \in \Re^{n\times n}$ is the centripetal and Coriolis matrix, $G(q,\dot{q}) \in \Re^n$ is the friction and gravitational vector, $A(q) \in \Re^{m\times n}$ is the matrix associated with constrains, $\delta \in \Re^n$ denotes bounded unknown disturbances including unstructured dynamics, and $B(q) \in \Re^{n\times p}$ is the input transformation matrix.

The m kinematic constrains are described by:

$$A(q)\dot{q} = 0 \qquad (2)$$

Note that, in the following, the $p = n - m$ case is considered.

With respect to the dynamics of a mobile robot (1), the following properties hold [5].

Property 1: $M(q)$ is a bounded symmetric and positive definite matrix.

Property 2: The parameter matrix is bound, that is:

$$M_{\min} I_n \le M(q) \le M_{\max} I_n , \qquad (3)$$

$$V(q,\dot{q}) \le V_b(q)\|\dot{q}\| , \qquad (4)$$

where:

- $M_{\min}, M_{\max}$ are positive scalar constants depending on the mass properties and constraint matrix,
- I_n is the $n \times n$ identity matrix,
- $V_b(q)$ is a positive definite function of q.

Property 3: The matrix $\dot{M} - 2V$ is skew symmetric [2], [5], that is

$$\dot{M} - 2V = -\left(\dot{M} - 2V\right)^T \text{ with } \dot{M} = V + V^T \qquad (5)$$

or equivalently,

$$x^T\left(\dot{M} - 2V\right)x = 0, \quad \forall x \in \Re^n \qquad (6)$$

Assume that the robot is fully actuated. Let $S(q) \in \Re^{n\times(n-m)}$ denote a full rank matrix formed by $(n-m)$ columns that span the null space of $A(q)$ defined in (2), as:

$$S^T(q)A^T(q) = 0 \qquad (7)$$

From (7), one cans find an auxiliary vector $\omega(t) \in \Re^{n-m}$ so that for all t,

$$\dot{q} = S(q)\omega(t) \qquad (8)$$

This is called the *steering system* where $\omega(t)$ can be regarded as a velocity input vector.

Equations (1) and (8) describe the *dynamics equations* of a mobile robot subject to kinematic constrains. Multiplying both sides of (1) by S^T and using (7) to eliminate the constraint force we obtain

$$S^T M(q)\ddot{q} + S^T V(q,\dot{q})\dot{q} + S^T G(q,\dot{q}) + S^T\delta = S^T B(q)\tau \qquad (9)$$

Substituting (8) and its derivative, that is $\ddot{q} = S(q)\dot{\omega} + \dot{S}(q)\omega$ into (9) one obtains:

$$S^T M(q)\dot{S}\omega + S^T M(q)S\dot{\omega} + S^T V(q,\dot{q})S\omega + S^T G(q,\dot{q}) + S^T\delta = S^T B(q)\tau \qquad (10)$$

Equation (10) can by written in a compact form as:

$$\overline{M}(q)\dot{\omega} + \overline{V}(q,\dot{q})\omega + \overline{G}(q) + \overline{\delta} = \overline{B}(q)\tau \qquad (11)$$

where:

- $\overline{M} = S^T MS$,
- $\overline{V} = S^T(M\dot{S} + VS)$,
- $\overline{G} = S^T G$,
- $\overline{\delta} = S^T\delta$,
- $\overline{B} = S^T B$.

Property 4: The matrix $\dot{\overline{M}} - 2\overline{V}$ in (11) is skew symmetric.

Proof: $\dot{\overline{M}} - 2\overline{V} = 2S^T M\dot{S} + S^T \dot{M} S -$

$$- 2S^T (M\dot{S} + VS) = S^T (\dot{M} - 2V)S \,.$$

Since $\dot{M} - 2V$ is skew symmetric, therefore, $\dot{\overline{M}} - 2\overline{V}$ is also skew symmetric.

3. NEURAL NETWORK CONTROLLER DESIGN FOR MOBILE ROBOT

3.1. *Problem statement*

In an application, a mobile robot is required to perform some tasks defined in its task-space. In order to achieve this objective firstly, some reference trajectories $q_d(t)$ are derived. Torque commands τ are then generated by the controller to make the mobile robot track the reference trajectories.

3.2. *Neural network controller design procedure*

In this section, a neural network based control design procedure for a stable tracking of a reference trajectory for the mobile robot described by (8) and (11) is derived. The procedure steps are as follows:

(i) the robot dynamics is redefined as an error dynamics based on a set of appropriate chosen Lyapunov functions;

(ii) a NN-based estimator is constructed and a NN learning law is proposed;

(iii) a new control law is derived and

(iv) a proof on the tracking stability of the overall closed-loop system and the boundedness on NN weight estimation errors is derived.

In this paper, it is assumed that the reference trajectories are available, i.e. they have already been derived based on the desired task-trajectories. The main concern is to provide proper torque inputs that guarantee a stable tracking of reference trajectories in the presence of parameter uncertainty and unknown disturbances.

From previous section it can be seen that for a mobile robot a tracking error may be defined as

$$\tilde{q} = q_d - q \tag{12}$$

where q_d is a desired trajectory.

Assume that there exist a Lyapunov function $V_1(\tilde{q}, t)$, a positive continuous function $W_1(t) > 0$ and a reference smooth feedback velocity $\omega_d(t)$, such that [5]:

$$\frac{\partial V_1}{\partial t} + \frac{\partial V_1}{\partial \tilde{q}} \dot{\tilde{q}} \Big|_{\dot{q}=S(q)\omega_d} \le -W_1(t) \text{ when } \tilde{q} \ne 0 \,. \tag{13}$$

Now, the objective is to derive proper torque input τ in (11), such that the angular velocity trajectory $\omega(t)$ defined in (8) tracks the reference velocity $\omega_d(t)$.

Define the *robot velocity tracking error* $\tilde{\omega}$ as

$$\tilde{\omega} = \omega - \omega_d \,. \tag{14}$$

Differentiating (14), multiplying both side by $\overline{M}$ and substituting (7) into it yields

$$\overline{M}(q)\dot{\tilde{\omega}} + \overline{V}(q,\dot{q})\tilde{\omega} + \overline{G}(q) + \overline{\delta} +$$

$$+ \overline{M}(q)\dot{\omega}_d + \overline{V}(q,\dot{q})\omega_d = \overline{B}(q)\tau \tag{15}$$

Equation (15) represents the *mobile robot dynamics* in term of tracking errors.

Let us choose a Lyapunov function V_2 as

$$V_2 = \frac{1}{2}\tilde{\omega}^T \overline{M}\tilde{\omega} \,. \tag{16}$$

Differentiating (16) along the system trajectories and using (15) and *Property 4* yields

$$\dot{V}_2 = \tilde{\omega}^T \overline{M}\dot{\tilde{\omega}} + \frac{1}{2}\tilde{\omega}^T \dot{\overline{M}}\tilde{\omega} =$$

$$= \tilde{\omega}^T (\overline{B}\tau - \overline{G} - \overline{\delta} - \overline{M}\dot{\omega}_d - \overline{V}\omega_d) \,. \tag{17}$$

To design the robot torque input, we choose a Lyapunov function as

$$V_3 = V_1 + \frac{1}{2}(S\tilde{\omega})^T M (S\tilde{\omega}) = V_1 + V_2 \,. \tag{18}$$

Differentiating (18) yields

$$\dot{V}_3 \le -W_1(t) + \tilde{\omega}^T (\overline{B}\tau - \overline{G} - \overline{\delta} - \overline{M}\dot{\omega}_d - \overline{V}\omega_d)$$

$$= -W_1(t) + \tilde{\omega}^T (\overline{B}\tau - \overline{G} - \overline{M}\dot{\omega}_d - \overline{V}\omega_d) - \tilde{\omega}^T \overline{\delta}$$

$$= -W_1(t) + \tilde{\omega}^T (\overline{B}\tau - \psi) - \tilde{\omega}^T \overline{\delta} \,, \tag{19}$$

with the unknown nonlinear term

$$\psi = \overline{M}\dot{\omega}_d + \overline{V}\omega_d + \overline{G} \,. \tag{20}$$

The nonlinear term ψ in (20) will be identified on-line by using a radial basis function (RBF) NN estimator.

It is known that RBF networks have capacity to approximate any smooth function on a compact set $S_x \subset \Re^n$ [6], [7], [8], [9]. If $f(\cdot): S_x \to \Re^n$ is a smooth function and $\{\varphi(x)\}$ is a RBFs basis set, then for each continuous function $f(\cdot)$, there exists a weight matrix W such that

$$f(x) = W^T \varphi(x) + \varepsilon, \tag{21}$$

where the approximation error is bounded by

$$\|\varepsilon\| < \varepsilon_N \text{ with } \varepsilon_N > 0. \tag{22}$$

Then, the unknown function ψ in (20) may by identified using a RBF neural network with sufficiently high number n_n of nodes such that

$$\psi = W^T h(x) + \varepsilon, \tag{23}$$

where x is the input pattern to the neural network defined as

$$x = [q_d^T \ \omega_d^T \ \dot{\omega}_d^T \ \tilde{\omega}^T]^T. \tag{24}$$

$W \in \Re^{n_n \times (4n-3m)}$ in (23) is the ideal and unknown weight matrix, which is assumed to be constant and bounded by

$$\|W\|_F = \sqrt{tr(W^T W)} \le W_B, \tag{25}$$

with W_B a known positive constant and $\|W\|_F$ the Frobenius norm. The basis functions in vector $h(x)$ can be chosen as Gaussian functions defined as

$$h_i(x) = \exp\left(-\|x - c_i\|^2 / \sigma_i^2\right), \quad i = 1, 2, \ldots, n_n, \tag{26}$$

where:

- c_i are centers and
- σ_i are widths,

which are chosen a priori and kept fixed throughout for simplicity. Therefore, during the learning process, only the weight matrix W must to be adjusted. The estimates of ψ are given by

$$\hat{\psi} = \hat{W}^T h(x). \tag{27}$$

In the development of the NN on-line estimators, radial basis function (RBF) network with fixed centers and widths is employed.

Thus, the main objective is to design a proper control law and properly NN learning laws, such that the unknown robot dynamics (20) can be compensated for by the NN estimator (27), and the stability of the robot error dynamics (15) and the boundedness on the estimation weights can be guaranteed. In this way we formulate the following theorem.

Theorem. If for the system (11) the control law is chosen as

$$\tau = \bar{B}^{-1}(-k\tilde{\omega} + \hat{\psi}), \tag{28}$$

with $\tilde{\omega}$ given by (10), and the weight updating law for the neural network as

$$\dot{\hat{W}} = -\beta(h\tilde{\omega}^T + \mu\|\omega\|\hat{W}) \tag{29}$$

where:

- $k > 0$ is the control gain,
- $\beta > 0$ is the learning rate and
- $\mu > 0$ is a design parameter.

Then, by properly choosing of k and μ, the tracking errors of error dynamics described by (8) and (15) and the NN estimation weights $\hat{W}$ are all guaranteed to be uniformly ultimately bounded (UUB).

Definition. Consider the dynamic system $\dot{x} = f(x,t)$ with $x \in \Re^n$. The equilibrium point x_e is said to be uniformly ultimately bounded (UBB) if there exist a compact set $S_x \subset \Re^n$ so that for all $x_0 \in S_x$ there exist a bound B and a time $T(B, x_0)$, such that $\|x(t) - x_e\| \le B$, for all $t \ge t_0 + T$, where t_0 is the initial time and $x_0 = x(t_0)$ the initial condition.

Proof of the Theorem. Assume that the approximation (23) holds, for all x in a compact set S_x. Substituting (28) into (19) yields

$$\begin{aligned} \dot{V}_3 &\le -W_1(t) + \tilde{\omega}^T\left(\bar{B}\bar{B}^{-1}(-k\tilde{\omega} + \hat{\psi}) - \psi\right) - \tilde{\omega}^T\bar{\delta} \\ &\le -W_1(t) - k\tilde{\omega}^T\tilde{\omega} - \tilde{\omega}^T\tilde{W}^T h - \tilde{\omega}^T\varepsilon - \tilde{\omega}^T\bar{\delta}, \end{aligned} \tag{30}$$

where $\tilde{W} = W - \hat{W}$. Since $W_1(t) > 0$, from (30) one obtains

$$\dot{V}_3 \le -k\tilde{\omega}^T\tilde{\omega} - \tilde{\omega}^T\tilde{W}^T h - \tilde{\omega}^T(\varepsilon + \bar{\delta}). \tag{31}$$

If denote by $k^* = \min k$, based on the boundedness ε defined in (22), from (31) it follows that

$$\dot{V}_3 \le -k^*\|\tilde{\omega}\|^2 - \tilde{\omega}^T\tilde{W}^T h + \|\tilde{\omega}\|(\varepsilon_N + \delta_N)\,, \qquad (32)$$

where $\delta_N = \|\bar{\delta}\|$ with $\delta_N > 0$.

Le us chose a Lyapunov function as

$$V = V_3 + \frac{1}{2\beta} tr\{\tilde{W}^T\tilde{W}\}. \qquad (33)$$

Differentiating (33) and substituting (32) into it yields

$$\dot{V} = \dot{V}_3 + \frac{1}{\beta} tr\{\tilde{W}^T\dot{\tilde{W}}\} \le -k^*\|\tilde{\omega}\|^2 - \tilde{\omega}^T(\tilde{W}^T h) +$$
$$+\|\tilde{\omega}\|(\varepsilon_N + \delta_N) - \frac{1}{\beta} tr\left\{\tilde{W}^T\dot{\hat{W}}\right\} \le$$
$$\le -k^*\|\tilde{\omega}\|^2 + \|\tilde{\omega}\|(\varepsilon_N + \delta_N) -$$
$$-\frac{1}{\beta} tr\left\{\tilde{W}^T(\dot{\hat{W}} + \beta h\tilde{\omega}^T)\right\}. \qquad (34)$$

Substituting now (29) into (34) we obtain

$$\dot{V} \le -k\|\tilde{\omega}\|^2 + \|\tilde{\omega}\|(\varepsilon_N + \tau_N)$$
$$-\frac{1}{\beta} tr\left\{-\tilde{W}^T\beta h\tilde{\omega}^T - \mu\beta\tilde{W}^T\|\tilde{\omega}\|\hat{W} + \tilde{W}^T\beta h\tilde{\omega}^T\right\}$$
$$= -k^*\|\tilde{\omega}\|^2 + \|\tilde{\omega}\|(\varepsilon_N + \delta_N) + \mu\|\tilde{\omega}\| tr\left\{\tilde{W}^T\hat{W}\right\}. \qquad (35)$$

Using [5],

$$tr\left\{\tilde{W}^T\hat{W}\right\} = tr\left\{\tilde{W}^T(W - \tilde{W})\right\} =$$
$$= \left\langle \tilde{W}, W \right\rangle_F - \left\|\tilde{W}\right\|_F^2 \le \left\|\tilde{W}\right\|_F \|W\|_F - \left\|\tilde{W}\right\|_F^2, \qquad (36)$$

relation (35) can be written as:

$$\dot{V} \le -k^*\|\tilde{\omega}\|^2 + \mu\|\tilde{\omega}\|\left\|\tilde{W}\right\|_F \|W\|_F -$$
$$-\mu\|\tilde{\omega}\|\left\|\tilde{W}\right\|_F^2 + \|\tilde{\omega}\|(\varepsilon_N + \delta_N)$$
$$\le -k^*\|\tilde{\omega}\|^2 + \mu\|\tilde{\omega}\|\left\|\tilde{W}\right\|_F W_B - \mu\|\tilde{\omega}\|\left\|\tilde{W}\right\|_F^2 +$$
$$+\|\tilde{\omega}\|(\varepsilon_N + \delta_N)$$
$$= -\|\tilde{\omega}\|\left\{k^*\|\tilde{\omega}\| - \mu\left\|\tilde{W}\right\|_F W_B + \mu\left\|\tilde{W}\right\|_F^2 - (\varepsilon_N + \delta_N)\right\}$$
$$= -\|\tilde{\omega}\|\left\{k^*\|\tilde{\omega}\| + \mu\left(\left\|\tilde{W}\right\|_F - \frac{W_B}{2}\right)^2 - \right.$$
$$\left. -\left(\frac{\mu W_B^2}{4} + \varepsilon_N + \delta_N\right)\right\} \qquad (37)$$

It can be seen that if the parameters μ and k^* are chosen so that

$$\left\|k^*\right\| > \frac{\mu W_B^2 + 4(\varepsilon_N + \delta_N)}{4\|\tilde{\omega}\|}. \qquad (38)$$

then $\dot{V} \le 0$.

According to Lyapunov theory and to LaSalle principle, this demonstrates the UUB [4] of the tracking error $\tilde{\omega}$ and the NN weight errors $\tilde{W}$ and, subsequently, the weight estimates $\hat{W}$. Therefore, the control torque (28) is also bounded.

4. CASE STUDY

In this section, an adaptive NN-based tracking controller is designed for the kinematic and the dynamic model corresponding to a mobile robot with two actuated wheels, shown in Fig. 1.

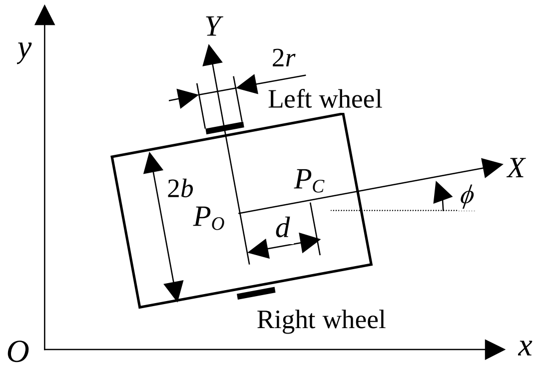

Fig. 1. A mobile robot with two actuated wheels.

The performance of this controller is compared to the performance of a feedback controller designed for the kinematic model like in Fukao, *et al.*, 2000 [2].

4.1. *Mathematical model*

Consider the mobile robot with two actuated wheels, shown in Fig. 1, where $2b$ is the width of the mobile robot and r is the radius of the wheel, Oxy is the world coordinate system and P_0XY is the coordinate system fixed to the mobile robot. P_0 is the middle between the right and the left driving wheels. P_c denotes the center of mass of the mobile robot, which is on the X - axis, at the distance d from the origin P_0.

Let denote by m_c and m_w the mass of the robot body and a wheel with a motor respectively, and I_c, I_w and I_m the moment of inertia of the body about the vertical axis through P_c, the wheel with a motor about the wheel axis, and the wheel with a motor about the wheel diameter, respectively.

The configuration of the mobile robot can be described by five generalized coordinates:

$$q = [x\ y\ \phi\ \theta_r\ \theta_l]^T, \tag{39}$$

where (x, y) are the coordinates of the origin P_0, ϕ is the heading angle of the mobile robot, and θ_r and θ_l are the angles of the rights and the left driving wheels.

Assuming the wheels roll and do not sleep, then, there exist three constraints: the velocity of P_0 must be in the direction of the axis of symmetry and the wheels must do not slip:

$$\begin{aligned} &\dot{y}\cos\phi - \dot{x}\sin\phi = 0, \\ &\dot{x}\cos\phi + \dot{y}\sin\phi + b\dot{\phi} = r\dot{\theta}_r, \\ &\dot{x}\cos\phi + \dot{y}\sin\phi - b\dot{\phi} = r\dot{\theta}_l. \end{aligned} \tag{40}$$

The three kinematic constraints can be rewritten in the form (2), i.e.

$$A(q)\dot{q} = 0, \tag{41}$$

where

$$A(q) = \begin{bmatrix} \sin\phi & -\cos\phi & 0 & 0 & 0 \\ \cos\phi & \sin\phi & b & -r & 0 \\ \cos\phi & \sin\phi & -b & 0 & -r \end{bmatrix}. \tag{42}$$

The kinematic model has the form (5), i.e.

$$\dot{q} = S(q)\omega(t), \tag{43}$$

with

$$S(q) = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ \frac{r}{2b} & \frac{r}{2b} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \omega = \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}, \tag{44}$$

where ω_r and ω_l represent the angular velocities of right and left wheels.

If we denote by v_{mr} and ω_{mr} the linear and angular velocities of the mobile robot at the point P_0, the relationship between v_{mr} and ω_{mr} and ω_r, ω_l is the following:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{45}$$

Then, the simplest kinematic form of a mobile robot with two actuated wheels is:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \phi \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{mr} \\ \omega_{mr} \end{bmatrix} \tag{46}$$

The dynamic model has the form (7), i.e.

$$\overline{M}(q)\dot{\omega} + \overline{V}(q,\dot{q})\omega + \overline{G}(q) + \overline{\delta} = \overline{B}(q)\tau \tag{47}$$

where $\overline{\tau}_d = 0$ and $\overline{M}, \overline{V}, \overline{B}$ are expressed as [5]:

$$\overline{M} = \begin{bmatrix} \frac{r^2}{4b^2}(mb^2 + I) + I_w & \frac{r^2}{4b^2}(mb^2 - I) \\ \frac{r^2}{4b^2}(mb^2 - I) & \frac{r^2}{4b^2}(mb^2 + I) + I_w \end{bmatrix},$$

$$\overline{V} = \begin{bmatrix} 0 & \frac{r^2}{2b} m_c d\dot{\phi} \\ -\frac{r^2}{2b} m_c d\dot{\phi} & 0 \end{bmatrix},$$

$$\overline{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{48}$$

with $m = m_c + 2m_w$, $I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$.

$\tau = [\tau_r, \tau_l]^T$ consists of motors' torques τ_r and τ_l, which act on the right and left wheels, respectively.

4.2. *Reference trajectory*

Let the reference trajectory of the robot be prescribed as:

$$\frac{d}{dt}\begin{bmatrix} x_{ref} \\ y_{ref} \\ \phi_{ref} \end{bmatrix} = \begin{bmatrix} \cos\phi_{ref} & 0 \\ \sin\phi_{ref} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix} \tag{49}$$

where x_{ref}, y_{ref} and ϕ_{ref} are the configure of the reference robot, and v_{ref} and ω_{ref} are its reference inputs.

The tracking errors that is the difference of position and direction of the real robot from the reference robot denoted by e_1, e_2, e_3 are defined as [2]:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \\ \phi_{ref} - \phi \end{bmatrix}. \quad (50)$$

It is easily to show that these errors satisfy the equation

$$\frac{d}{dt}\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = v\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + \omega\begin{bmatrix} e_2 \\ -e_1 \\ -1 \end{bmatrix} + \begin{bmatrix} v_r \cos e_3 \\ v_r \sin e_3 \\ \omega_r \end{bmatrix}. \quad (51)$$

The control inputs denoted by v_c and ω_c which make e_1, e_2, e_3 converge asymptotically to zero, are given by [2], [5]:

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_{ref} \cos e_3 + k_1 e_1 \\ \omega_{ref} + v_{ref} k_2 e_2 + k_3 \sin e_3 \end{bmatrix}, \quad (52)$$

where the positive constants k_1, k_2, k_3 are control gains.

The stability of this tracking system can be proven by choosing the following Lyapunov function [2]:

$$V_0 = \frac{1}{2}(e_1^2 + e_2^2) + \frac{1 - \cos e_3}{k_2}. \quad (53)$$

It is easily to show that the derivative of V_0 along the system trajectories is negative definite.

4.3. *Simulation results*

The values of physical and design parameter are [2]: a = 2, b = 0.75, d = 0.3, r = 0.15, m_c = 30, m_w =1, I_c =15.625, I_w =0.005, I_m = 0.0025, k_1 = k_2 = k_3 = 5.

The reference inputs are chosen as follow [2] and are represented in Fig. 2:

$t \in [0, 5)$: $v_{ref} = 0.25(1 - \cos(\pi t / 5))$,
$\omega_{ref} = 0$;
$t \in [5, 20)$: $v_{ref} = 0.5,\ \omega_{ref} = 0$;
$t \in [20, 25)$: $v_{ref} = 0.25(1 + \cos(\pi t / 5))$,
$\omega_{ref} = 0$;
$t \in [25, 30)$: $v_{ref} = 0.15\pi(1 - \cos(2\pi t / 5))$,
$\omega_{ref} = -v_{ref} / 1.5$;
$t \in [30, 35)$: $v_{ref} = 0.15\pi(1 - \cos(2\pi t / 5))$,
$\omega_{ref} = v_{ref} / 1.5$;
$t \in [35, 40)$: $v_{ref} = 0.25(1 + \cos(\pi t / 5))$,
$\omega_{ref} = 0$;
$t \in [40, 50)$: $v_{ref} = 0.5,\ \omega_{ref} = 0$.

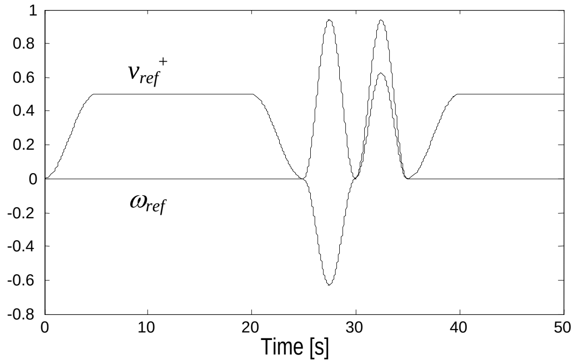

Fig. 2. Reference inputs v_{ref}, ω_{ref}.

For these reference inputs, the appropriate angular velocities ω_r and ω_l of the two robot wheels are represented in Fig. 3.

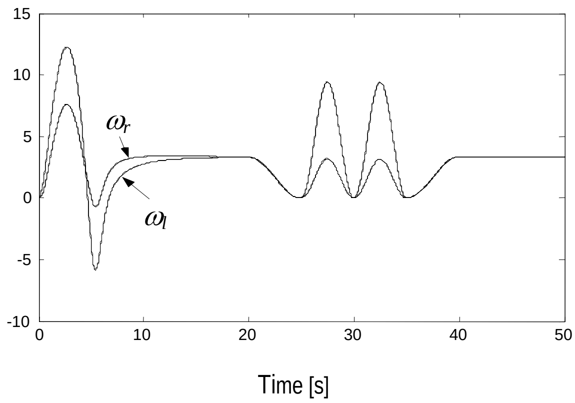

Fig. 3. Angular velocities ω_r, ω_l

The torque commands τ_r and τ_l that realize the two angular velocities designed according the procedure presented in Section 3.2 are shown in

Fig. 4.

Fig. 5 shows the tracking performance of the neural adaptive controller. It must be noted that this controller contains a RBF neural network, which identifies the completely unknown dynamics of the mobile robot included in the unknown nonlinear term ψ from (20).

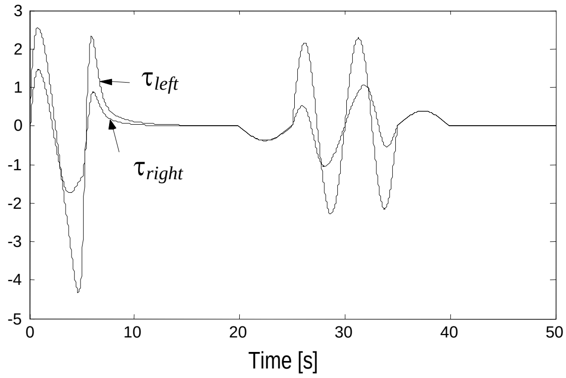

Fig. 4. Torque commands τ_r, τ_l .

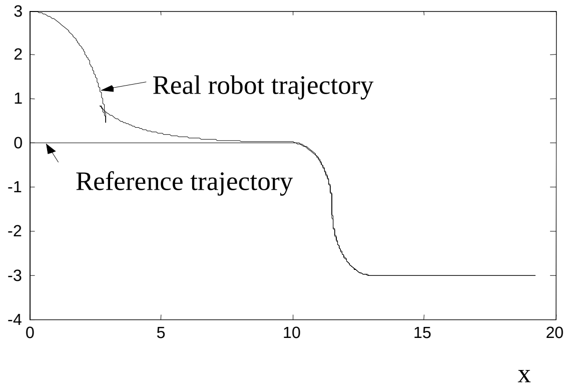

Fig. 5. Mobile robot trajectory.

The network weights were initialized with zero, and the widths σ_i of Gaussian functions have all been chosen at the value 0.025. The values of design parameters used in simulations are: β = 0.025, μ = 0.005, k = 0.02. No preliminary learning stage of neural network weights is required.

From Fig. 5 one can observe a good behavior of the proposed neural adaptive control system. Although the initial position of the mobile robot is distanced from the reference trajectory, the NN adaptive controller makes the mobile robot quite quickly approach the reference trajectory and afterwards get very close to it and follow its shape.

5. CONCLUSIONS

In this paper a design procedure of a neural network adaptive tracking controller for a mobile robot subject to kinematic constraints was presented.

The unknown dynamics of the mobile robot was on-line identified using NN-based estimators.

The form of the controller and the adaptation laws of NN weights were derived from a Lyapunov analysis of stability.

The simulation results conducted in the case of a mobile robot with two actuated wheels demonstrate a good behaviour of this NN adaptive controller.

6. REFERENCES

[1] Chen, F.C. and Liu, C.C. – “Adaptively controlling nonlinear continuous-time systems using multilayer neural networks”, *IEEE Trans. Automat. Contr.*, vol.39, no.6, pp. **1306-1310**, 1994.

[2] Fukao, T., Nakagawa, H. and Adachi, N. – “Adaptive tracking control of a nonholonomic mobile robot”, *IEEE Trans. Robot. Automat.*, vol. 16, no. 5, pp. **609-615**, 2000.

[3] Haykin, S. – “Neural Networks”, Macmillan Publishing Company, Englewood Cliffs, New York, 1994.

[4] Ioannou, P. and Sun, J., “Robust adaptive Control”, Prentice Hall PTR, 1996.

[5] Lin, S. and Goldenberg, A.A. – “Neural network control of mobile manipulators”, *IEEE Trans. Neural Networks*, vol. 12, no. 5, pp. **1121-1133**, 2001.

[6] Mayosky, M.A. and Cancelo, G.I.E. – “Direct adaptive control of wind energy conversion systems using Gaussian networks”, *IEEE Trans. Neural Networks*, vol. 10, no. 4, pp. **898-906**, 1999.

[7] McLain, R.B., Henson, M.A. and Pottmann M. – “Direct adaptive control of partially known nonlinear systems”. *IEEE Trans. Neural Networks*, vol. 10, no. 3, pp. **714-721**, 1999.

[8] Petre, E. – “An adaptive controller for a class of nonlinear dynamic systems”, *The 10th International Symposium on Systems Theory - SINTES 10*, Craiova-Romania, May 25-26, pp. A-**135-138**, 2000.

[9] Spooner, J.T. and Passino, K.M. – “Decentralized adaptive control of nonlinear systems using radial basis neural networks”. *IEEE Trans. Aut. Contr.*, vol.44, no.11, pp. **2050-2057**, 1999.