

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Design of Neural Network Mobile Robot Motion Controller

Jasmin Velagic, Nedim Osmic and Bakir Lacevic
*University of Sarajevo, Faculty of Electrical Engineering
 Bosnia and Herzegovina*

1. Introduction

Neural networks are recommended for intelligent control as a part of well known structures with adaptive critic (Barto, 1983; Lacaze, 1997; Hendzel, 2007). Recently, much research has been done on applications of neural networks for control of nonlinear dynamic processes (Narendra & Pathasarathy, 1990; Nguyen & Widrow, 1990). These works are supported by two of the most important capabilities of neural networks; their ability to learn and their good performance for the approximation of nonlinear functions (Hornik et al., 1990). At present, most of the works on system control using neural networks are based on multilayer feedforward neural networks with backpropagation learning or more efficient variations of this algorithm (Narendra & Pathasarathy, 1990). It has been shown (Hornik, 1990) that a neural network with one hidden layer with an arbitrarily large number of neurons in the hidden layer can be approximate any continuous functions over a compact subnet of \mathcal{R}^n .

The neural network-based control of mobile robots has recently been the subject of intense research (Corradini et al., 2003). It is usual to work with kinematic models of mobile robot to obtain stable motion control laws for trajectory following or goal reaching (Jiang, 2001; Ramírez & Zeghloul, 2000). The most authors solved the problem of mobile robot motion control and stability using nonlinear backstepping algorithm (Tanner & Kyriakopoulos, 2003; Velagic et al., 2006). Fierro and Lewis (1998) developed a neural network based model by combining the backstepping tracking technique and a torque controller, using a multi-layer feedforward neural network, where the neural network can learn the dynamics of the mobile robot by its on-line learning. But the control algorithm and the neural network learning algorithm are very complicated and it is computationally expensive. In (Yang & Meng, 2001) a single-layer neural network based controller for robot is proposed. This approach does not include any nonholonomic kinematics therefore it can not be used for robot with kinematic constraints. However, the problem of control of mobile robots has attracted the interest of researchers in view of its theoretical challenges. In fact, these systems are a typical example of nonholonomic mechanisms due to the perfect rolling constraints (no longitudinal or lateral slipping of the wheels).

In this paper we used a recurrent neural network for controlling the mobile robot with nonholonomic constraints. This network is trained on-line using the backpropagation optimization algorithm with an adaptive learning rate (Velagic & Hebibovic, 2004; Velagic

et al., 2005). The optimization algorithm is performed at each sample time to compute the optimal control inputs. This is a simple control system and computationally very effective for real-time requirements.

2. Control System of Mobile Robot

The proposed neural network control system is shown in Fig. 1. The control system consists of the neural network controller, the kinematic model of mobile robot, a reference trajectory generator and an encoder which provides odometric information. In this section the kinematic model of mobile robot with differential drive and convergence conditions of overall control structures are described. Design of neural network controller will be presented in the next subsection.

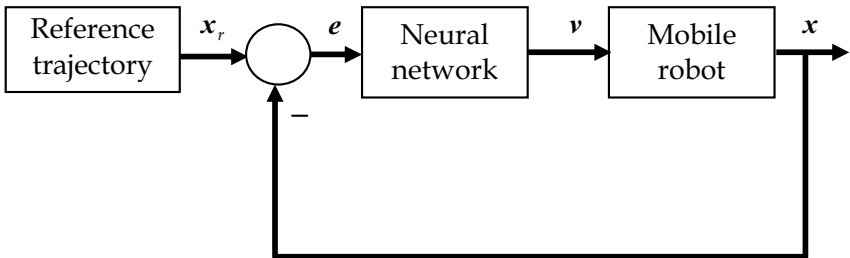


Fig. 1. Mobile robot motion control system

2.1. Kinematic model of mobile robot

In this paper the mobile robot with differential drive is used (Fig. 2).

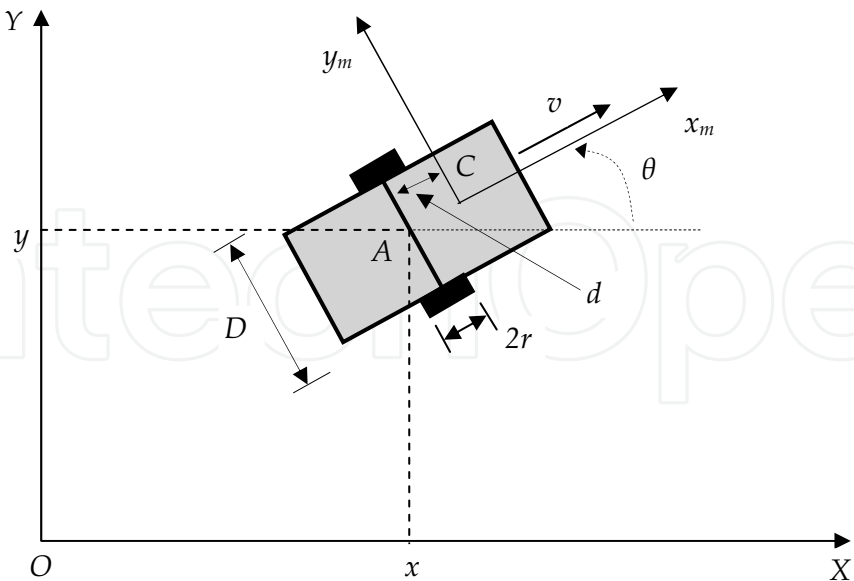


Fig. 2. The representation of a nonholonomic mobile robot.

The robot has two driving wheels mounted on the same axis and a free front wheel. The two driving wheels are independently driven by two actuators to achieve both the transition and orientation. The position of the mobile robot in the global frame $\{X, O, Y\}$ can be defined by the position of the mass center of the mobile robot system, denoted by C , or alternatively by position A , which is the center of mobile robot gear, and the angle between robot local frame $\{x_m, C, y_m\}$ and global frame.

Kinematic equations of the two-wheeled mobile robot are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

and

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} r & r \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix}. \quad (2)$$

where x and y are coordinates of the center of mobile robot gear, θ is the angle that represents the orientation of the vehicle, v and ω are linear and angular velocities of the vehicle, v_R and v_L are velocities of right and left wheels, r is a wheel diameter and D is the mobile robot base length.

Combining equations (1) and (2) yields:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r \cos(\theta) & r \cos(\theta) \\ r \sin(\theta) & r \sin(\theta) \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix}. \quad (3)$$

Inputs of kinematic model of mobile robot are velocities of right and left wheels v_R and v_L . The constraint that the wheel cannot slip in the lateral direction is:

$$\dot{x} \sin \theta + \dot{y} \cos \theta - d \dot{\theta} = 0. \quad (4)$$

The stability conditions of mobile robot system with PI controller will be investigated in the next subsection.

2.2. Convergence condition of control system

The feedback control system of mobile robot is shown in Fig. 3.

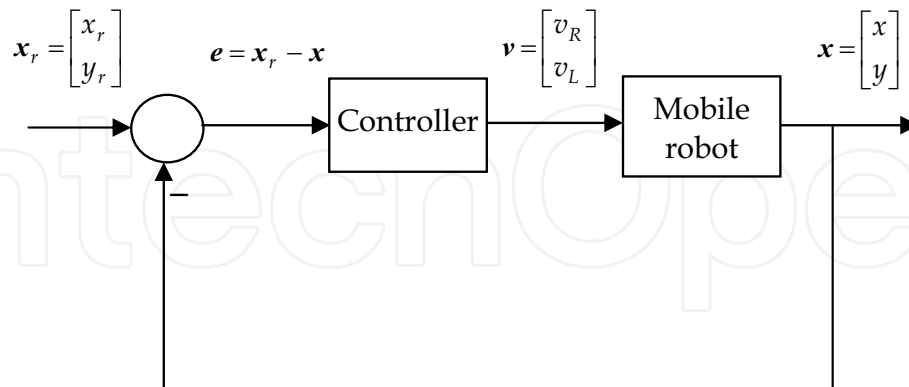


Fig. 3. Mobile robot control system

The time derivation of mobile robot output yields:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \quad (5)$$

or

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r \cos(\theta) & r \cos(\theta) \\ r \sin(\theta) & r \sin(\theta) \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix}. \quad (6)$$

The equation (6) can be rewritten in compact form as:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \mathbf{v} = \mathbf{J} \begin{bmatrix} v_R \\ v_L \end{bmatrix}. \quad (7)$$

where \mathbf{J} is Jacobian matrix.

It is required that the position error vector go exponentially to a zero as a function of time:

$$\frac{d\mathbf{e}}{dt} + \mathbf{K}\mathbf{e} = 0, \quad (8)$$

where $e_i = e_{i,0} \exp(-kt)$.

This requirement is satisfied with the following control law:

$$\mathbf{v} = \mathbf{K}(\mathbf{x}_r - \mathbf{x}) + \dot{\mathbf{x}}_r, \quad (9)$$

where matrix \mathbf{K} need to be positive definitive. On the other words the following relation hold:

$$k_1, k_2 > 0, \quad (10)$$

where k_1 and k_2 are elements of the matrix \mathbf{K} , respectively:

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}. \quad (11)$$

Using equation (8) velocities of the right and left wheels:

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \mathbf{J}^{-1} \mathbf{v} = \mathbf{K}(\mathbf{x}_r - \mathbf{x}) + \dot{\mathbf{x}}_r. \quad (12)$$

The Jacobian matrix \mathbf{J} is a non-singular one so as to assure that a control law based on the inversion of \mathbf{J} is possible.

Various controller parameters were tested by using lamniscate desired motion trajectories. The controller parameters, which ensure the good tracking performance is found to be:

$$k_1 = 102.9822, k_2 = 1.3536. \quad (13)$$

On the basis of this velocity controller, the neural network controller will be designed in the next section.

3. Neural Network Control Design

In design of velocity mobile robot controller the following simplifications are assumed:

- Kinematic model is captured by first order differential equation.
- Assume that an arbitrary commanded speed can be achieved (instantaneously).

Under these simplifications, results are expected:

- Exponential convergence of trajectory to desired trajectory.
- Speed of convergence can be tuned.

3.1. Design of neural network controller

A neural network (NN) performs the system model identification that will be used to design the appropriate intelligent mobile robot controller. The usage of NN for controlling a mobile robot is justified from the following reasons: the operational conditions considered raises

complex nonholonomic mobile robot kinematics and NN has universal approximation and supervised learning capabilities.

The neural network controller in Fig. 4, based on the recurrent network architecture, has a time-variant feature: once a trajectory is learned, the following learning takes a shorter time. The dynamic neural network is composed of two layered static neural network with feedbacks (one hidden and one output layers) (Fig. 5). The hidden layer contains ten tansigmoidal neurons and the output layer has one neuron with a linear activation function. It is important to note that rather than learning explicit trajectories, the neural network controller learns the relationship between linear velocities and position errors of the mobile robot. This network is trained using the backpropagation algorithm through time with an adaptive learning rate (Velagic & Hebibovic, 2004; Velagic et al., 2005). In the training phase the network is presented with a series of input-answer pairs.

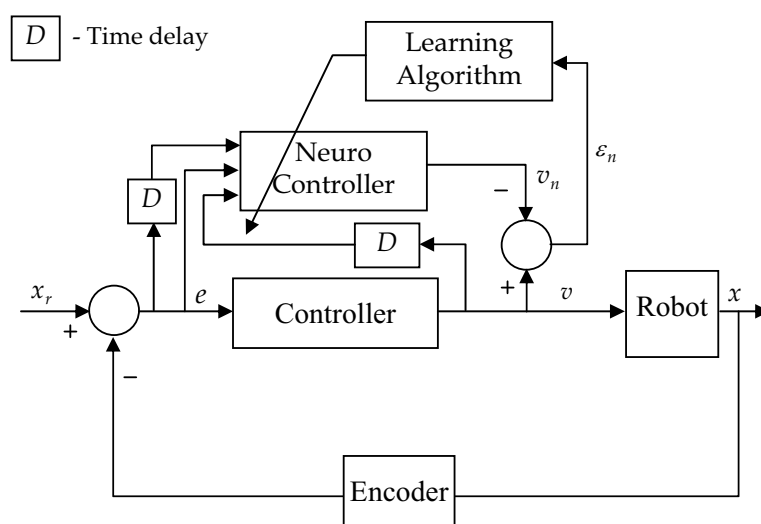


Fig. 4. Neural network learning architecture

The network's current output v_k is compared with the desired input e_k , and the errors are used to correct the weights in order to reduce the network's error on this input:

$$\varepsilon_n(x) = \frac{1}{2} \sum_k (e_k - v_k)^2. \quad (14)$$

Synaptic weights are updated as

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_j v_i, \quad (15)$$

$$w_{kj}(t+1) = w_{kj}(t) + \eta \delta_k v_j, \quad (16)$$

where η is the learning rate, δ_j is the error gradient at unit j , δ_k is the error at unit k and v_i and v_j are the outputs of unit i and j , respectively.

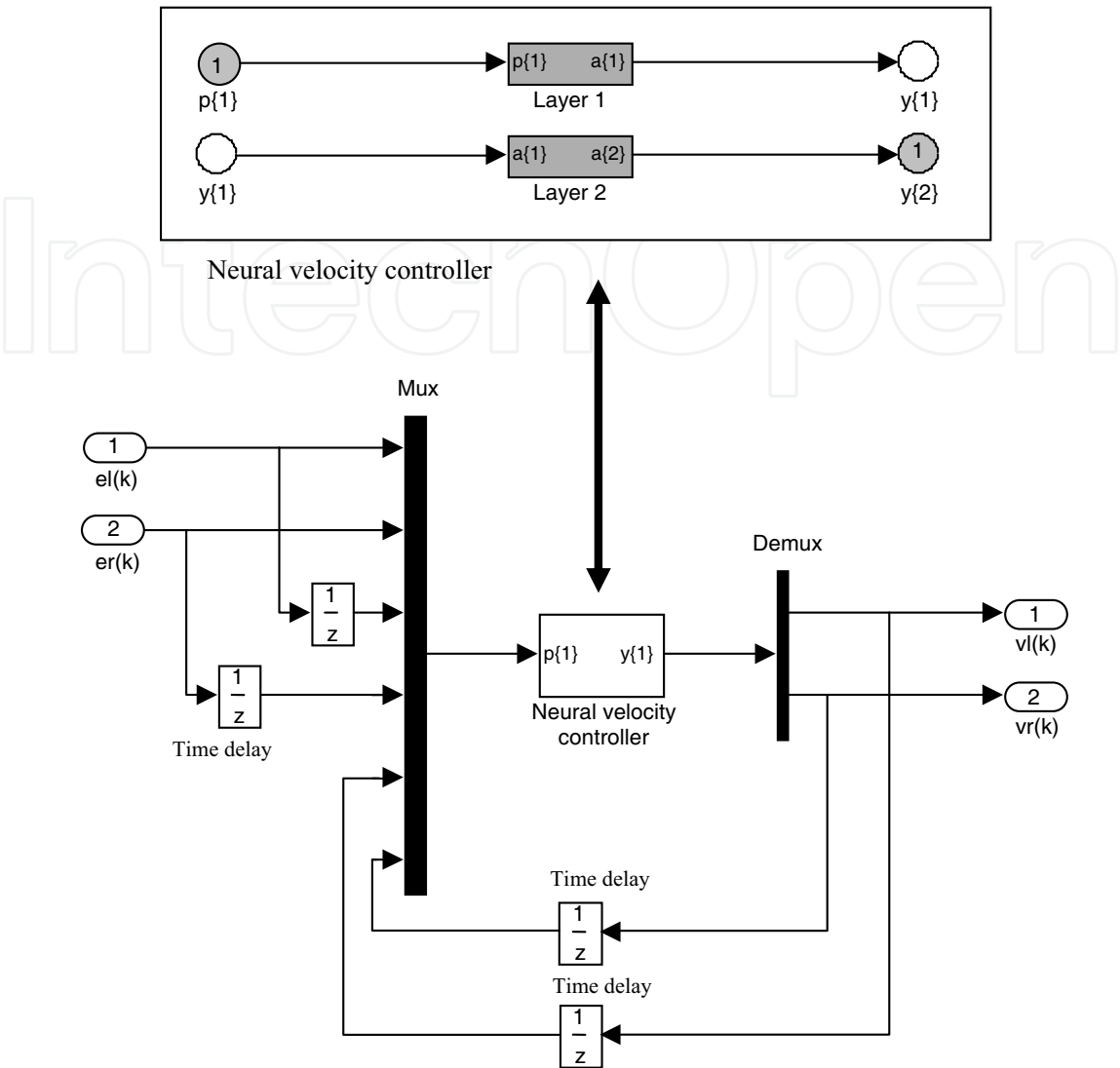


Fig. 5. Dynamical neural network.

A unifying framework for neural networks that encompasses process identification concept is to view neural network training as a nonlinear optimization problem:

$$\min_w J(\boldsymbol{w})$$

(17)

That is, we need to find values for neural network parameters \boldsymbol{w} (weight vector) for which some cost function $J(\boldsymbol{w})$ is minimized.

Let us assume that the controller is described by the following time difference equation:

$$v(t) = f(v(t-1), \dots, v(t-n); e(t), \dots, e(t-m)),$$

(18)

where $v(t)$ is the process output at time t depends on the past n output values and on the past m values of the input e . For identification plant model the neural network is used in the

following form:

$$v_n(\mathbf{w}, t) = f_w(v(t-1), \dots, v(t-n); e(t), \dots, e(t-m)), \quad (19)$$

where $n \geq m$ for physically realizable systems.

Here $f_w(\cdot)$ represents the nonlinear input-output map of the neural network which approximates the controller mapping $f(\cdot)$. The input to the neural network includes the past values of the controller output.

The training process for neural network modeling can be expressed as the minimization of an error measure. If the sampled process data are collected over a period $[0, T]$, the cost function $J(w)$ is defined as:

$$J(\mathbf{w}, t) = \sum_{t=0}^T [v(t) - v_n(\mathbf{w}, t)]^2. \quad (20)$$

In our algorithm (Velagic & Hebibovic, 2004; Velagic et al., 2005) we define relationship between $\Delta J(w, t)$ and $J(w, t)$ by the relative factor $\chi(t)$:

$$\chi(t) = \frac{\Delta J(\mathbf{w}, t)}{J(\mathbf{w}, t)} = \frac{J(\mathbf{w}, t) - J(\mathbf{w}, t-1)}{J(\mathbf{w}, t)}. \quad (21)$$

Then, we determine how to adjust learning rate term according to the relative factor χ . The adjustment of the learning rate is given as follows:

$$\eta(t+1) = \eta(t)[1 - \text{sgn}(\chi(t))v \cdot e^{-\chi(t)}], \quad v \in (0, 1). \quad (22)$$

The learning rate is adjusted at each iteration according to equations (21) and (22). The algorithm proceeds as follows. First, we select the number of neurons in hidden and output layers, initial value of learning rate and the parameter v . Then, the training process in the closed control loop is performed for various values of parameter v , $v \in [0, 1]$. We adopt the value of v for which a satisfactory identification performance is achieved. Our neural network has 10 sigmoid neurons in hidden layer and 2 linear neurons in its output layer. The proposed algorithm starts with the same initial learning rate $\eta=0.02$ for both layers. The good results were obtained with $v=0.71$. The trajectory tracking performance obtained by adopted neural controller will be shown in the following section.

4. Simulation Results

The effectiveness of the neural network controller is demonstrated in the case of tracking of a lamniscate curve. The trajectory tracking problem for a mobile robot is based on a virtual reference robot (Egerstedt et al., 2001) (Fig. 6) that has to be tracked.

The tracking position error between the reference robot and the actual robot can be expressed in the robot frame as:

$$\mathbf{e}_p = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \mathbf{T}_p \mathbf{e}_q = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}, \quad (23)$$

where $\mathbf{e}_q = [e_x \ e_y \ e_\theta]^T$.

The position error dynamics can be obtained from the time derivative of the (23) as:

$$\dot{e}_1 = \omega e_2 + u_1, \quad \dot{e}_2 = -\omega e_1 + v_r \sin e_3, \quad \dot{e}_3 = u_2, \quad (24)$$

where $v = v_r \cos e_3 - u_1$ and $\omega = \omega_r - u_2$.

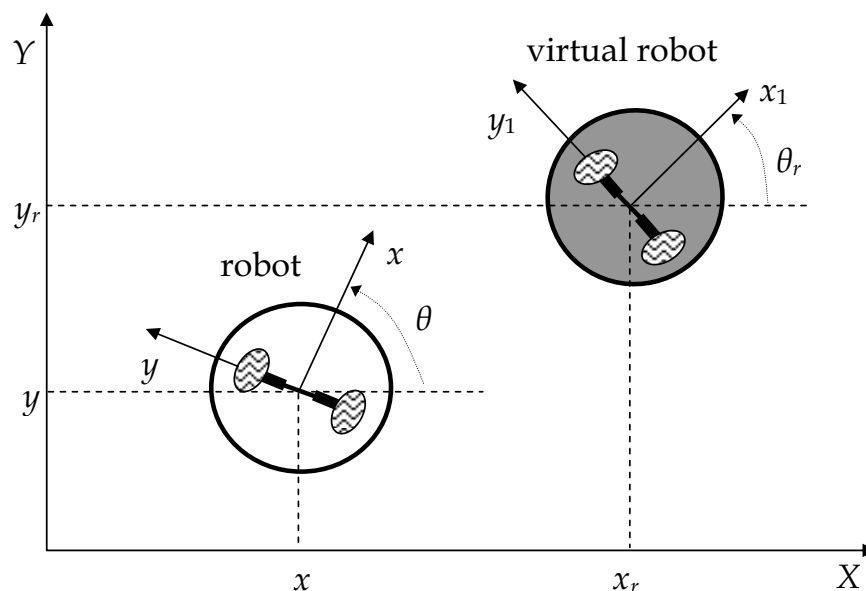


Fig. 6. The concept of tracking of a virtual reference robot.

The overall system is designed and implemented within Matlab/Simulink environment. We consider the following profiles: position, orientation, linear and angular velocities.

The simulation results obtained by neural network controller are shown in Figs. 7-14. Results achieved in Figs. 7-11 demonstrate the good position tracking performance. Also, a desired orientation during robot motion is following in satisfactory manner, which is depicted in Fig. 12. The proposed neural controller ensures small values of the control input velocities (linear) for obtaining the reference position trajectories (Figs. 13 and 14). This means that smaller power of DC motors is requested.

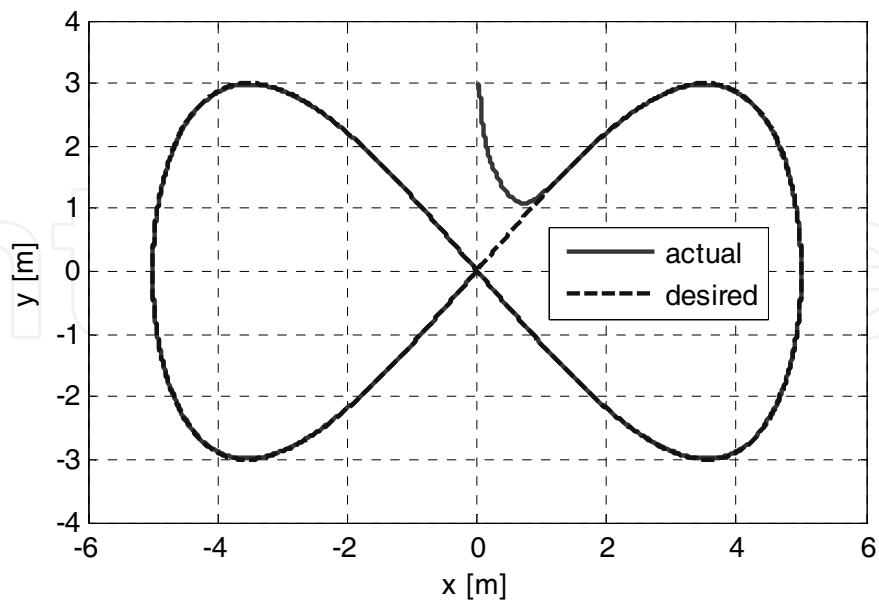


Fig. 7. Tracking the lamniscate trajectory

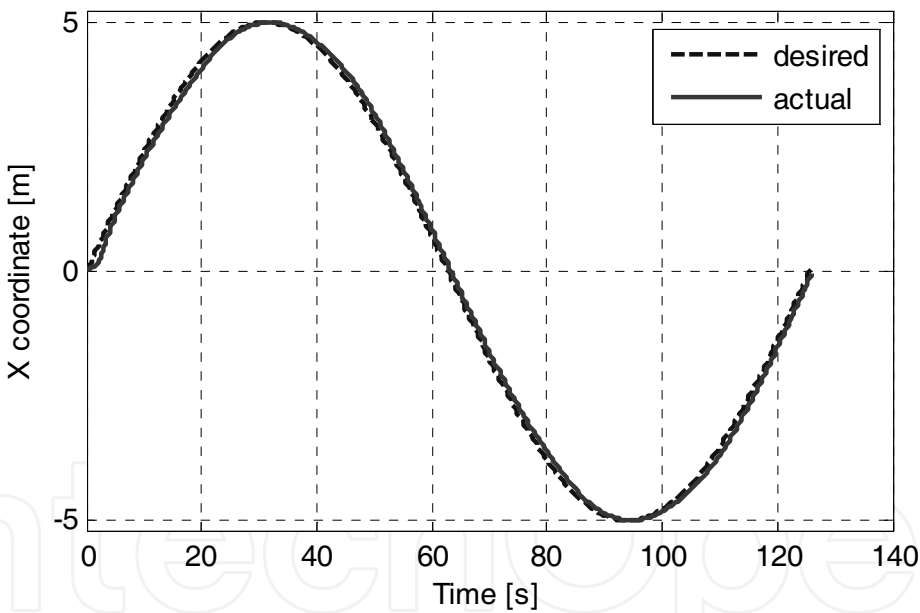


Fig. 8. Time history of x coordinate

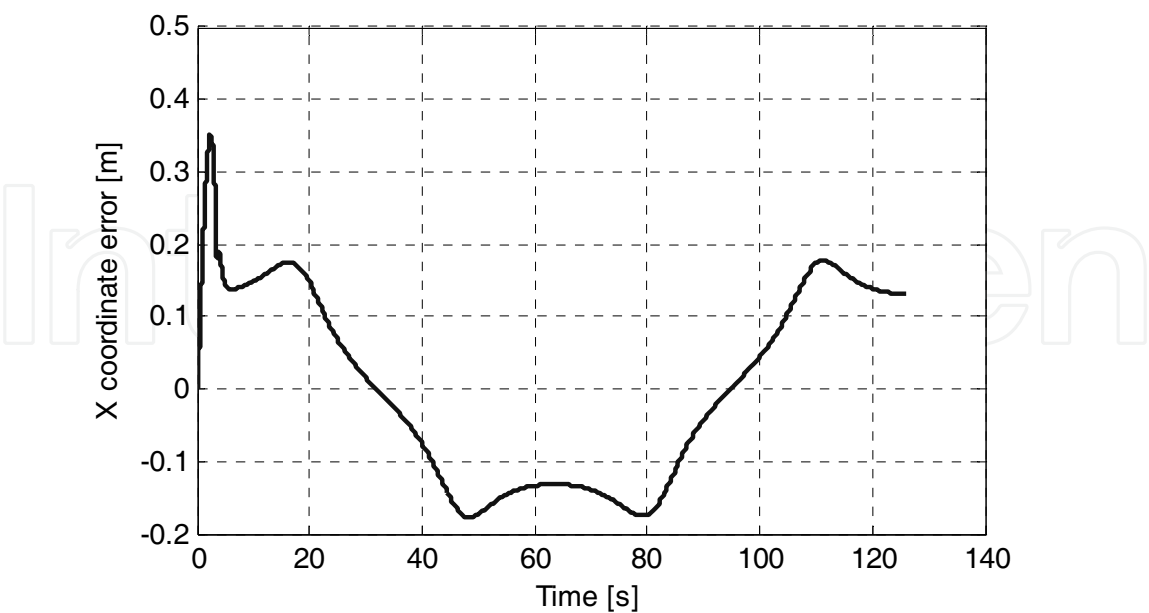


Fig. 9. X coordinate error

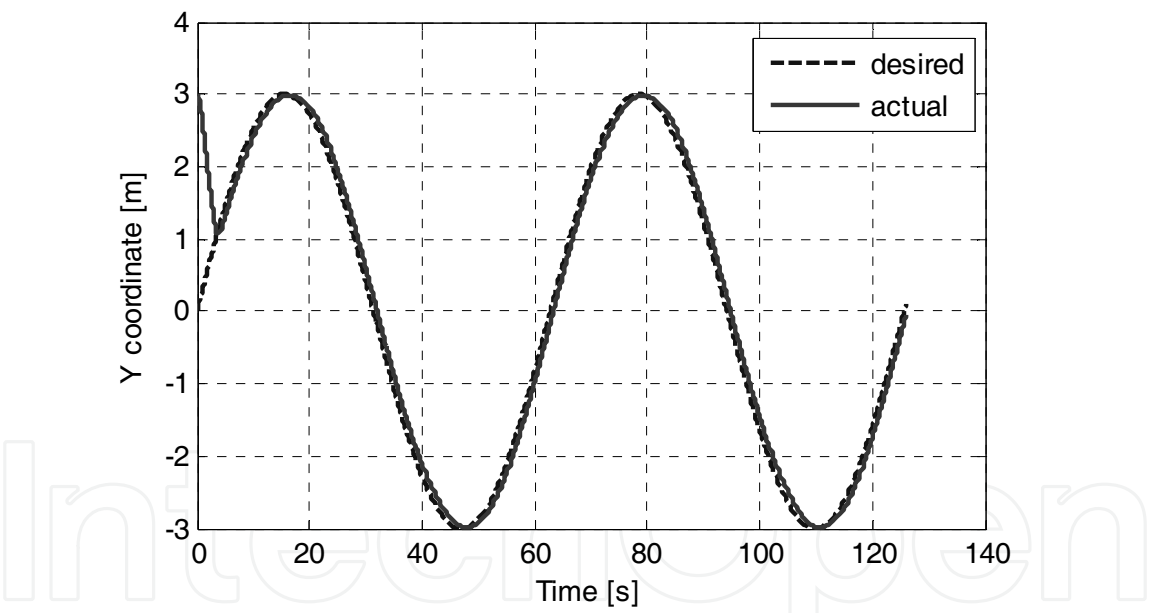


Fig. 10. Time history of y coordinate

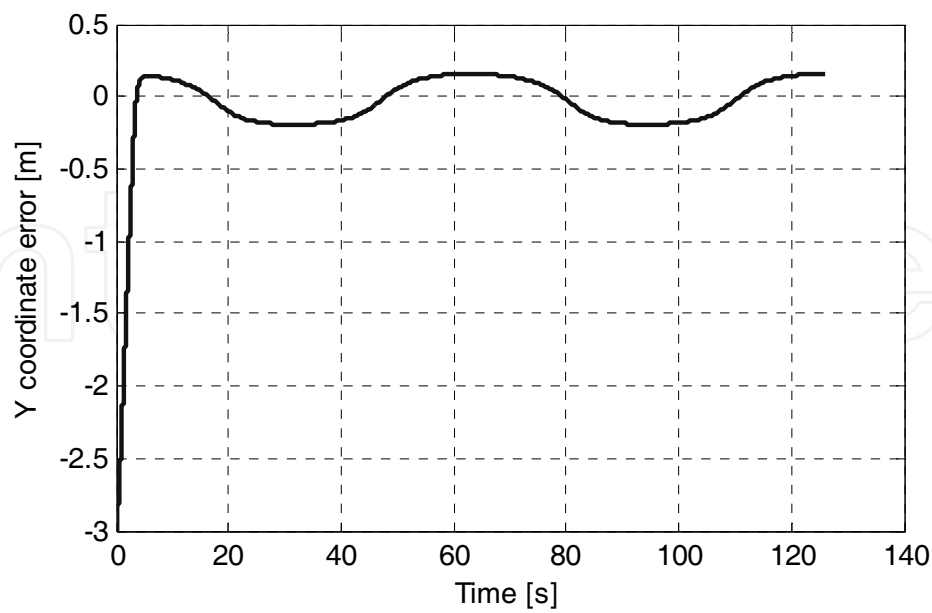


Fig. 11. Y coordinate error

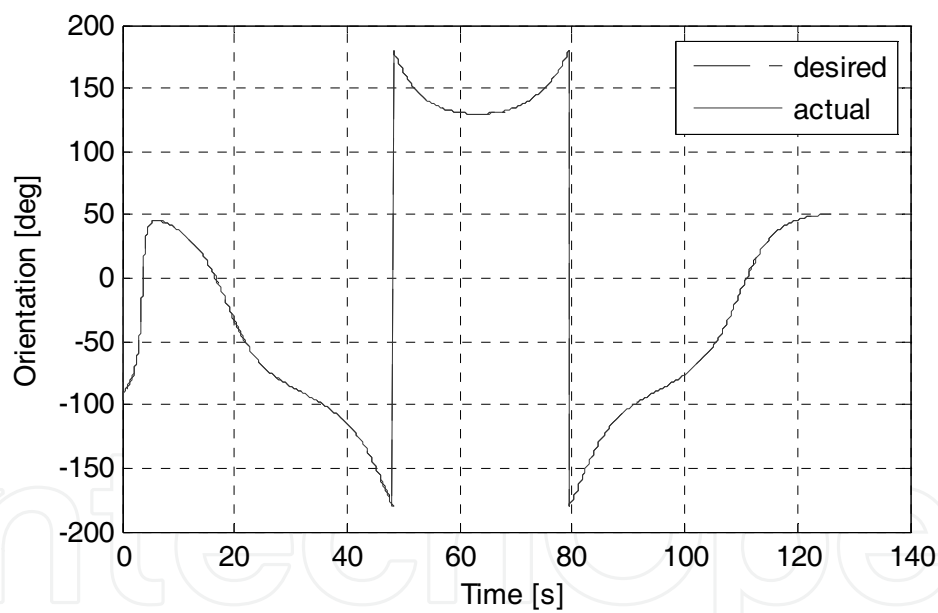


Fig. 12. Time response of robot orientation

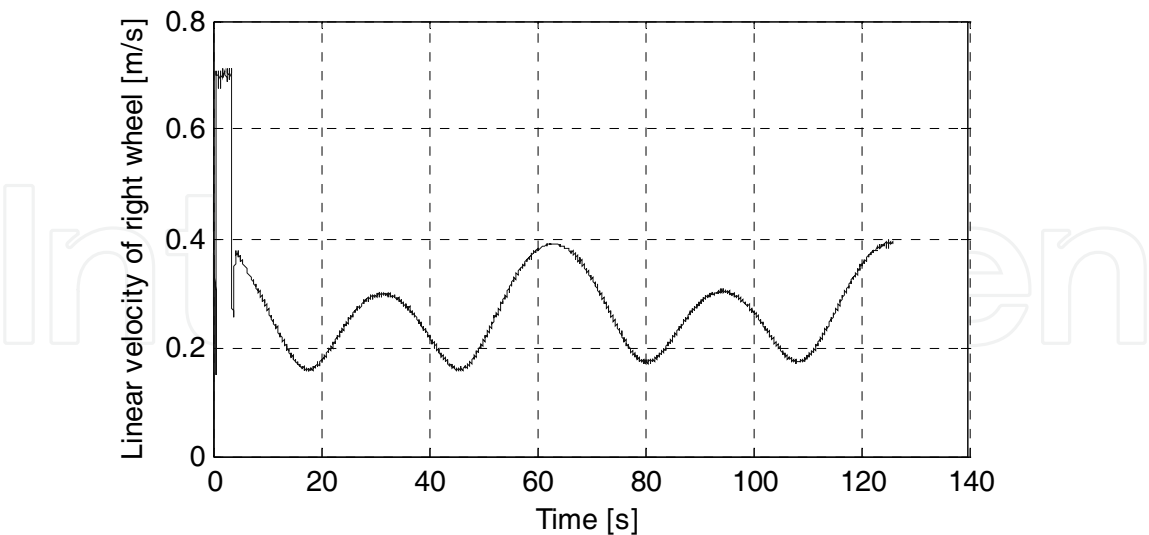


Fig. 13. Linear velocity of right wheel

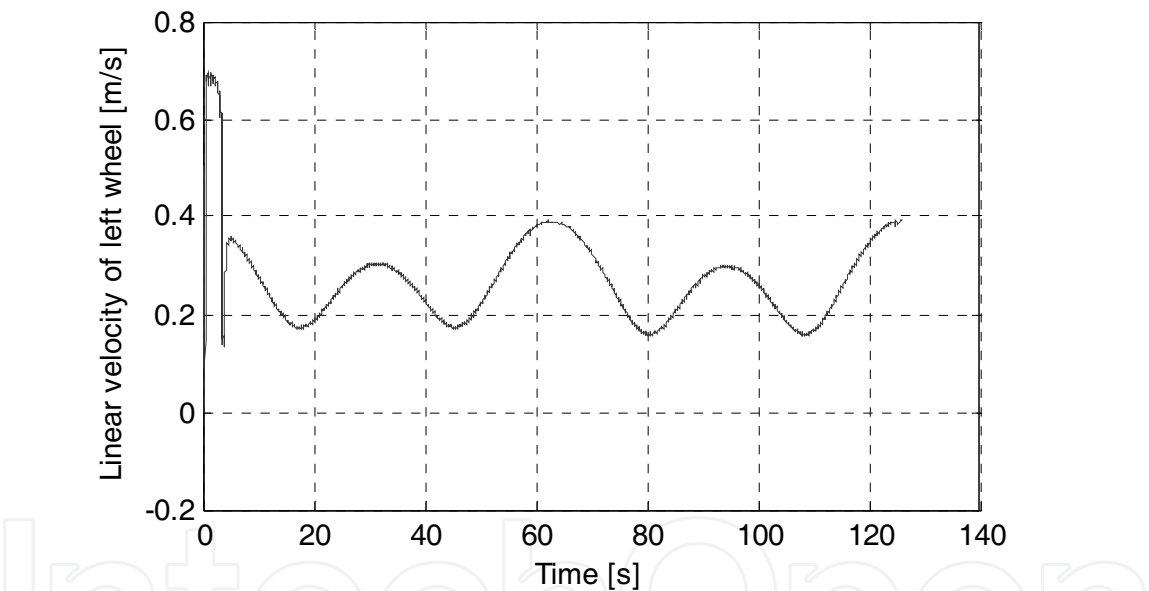


Fig. 14. Linear velocity of left wheel.

5. Conclusions

This paper presented a design of the neural network-based velocity controller for the kinematic model of mobile robot with differential drive. The proposed neural controller has two inputs (position errors of wheels) and two outputs (velocities of wheels). It is trained on-line through controlling of the mobile robot by PD controller. For this purpose the backpropagation algorithm with adaptive learning rate is applied. Simulation results demonstrate the effectiveness of the proposed neural network-based control system.

6. References

- Lacaze, A. (1997). A neural network planner that prunes its own tree. Term paper, Knowledge Engineering Department, University of Maryland.
- Barto, A.G. (1983). Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Transactions on System, Man and Cybernetics*, Vol. 13, pp. 834-846.
- Hendzel, Z. (2007). Adaptive critic neural networks for motion control of wheeled mobile robot. *Nonlinear Dynamics*, Vol. 50, No. 4, pp. 849-855.
- Narendra, K.S. & Pathasarathy, K. (1990). Identification and control of dynamic systems using neural network. *IEEE Transaction on Neural Networks*, Vol. 1, No. 1, pp. 4-27.
- Nguyen, D. & Widrow, B. (1990). Neural networks for self- learning control systems. *IEEE Control System Magazine*, Vol. 10, No. 1, pp. 18-23.
- Hornik, K., Stinchcombe, M. & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, Vol. 3.
- Corradini, M., Ippoliti, G. & Longhi, S. (2003). Neural networks based control of mobile robots: development and experimental validation, *Journal of Robotic Systems*, Vol. 20, No. 10, pp. 587-600.
- Jiang, Z.P., Lefeber, E. & Nijmeijer, H. (2001). Saturated stabilization and tracking of a nonholonomic mobile robot. *Systems & Control Letters*, Vol. 42, pp. 327-332.
- Ramírez, G. & Zeghloul, S. (2000). A new local path planner for nonholonomic mobile robot navigation in cluttered environments, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2058-2063.
- Tanner, H.G. & Kyriakopoulos, K.J. (2003). Backstepping for nonsmooth systems. *Automatica*, Vol. 39, pp. 1259-1265.
- Velagic, J., Lacevic, B. & Perunicic, B. (2006). A 3-Level autonomous mobile robot navigation system designed by using reasoning/search approaches. *Robotics and Autonomous Systems*, Vol. 54, No. 12, pp. 989-1004.
- Fierro, R. & Lewis, F. L. (1998). Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, Vol. 9, No. 4, pp. 389-400.
- Yang, S.X. & Meng, M. (2001). Real-time fine motion control of robot manipulators with unknown dynamics," in: *Dynamics in Continuous, Discrete and Impulse Systems*, Series B.
- Velagic, J. & Hebibovic, M. (2004). Neuro-fuzzy architecture for identification and tracking control of a robot, *Proceedings of the World Automation Congress - 5th International Symposium on Soft Computing for Industry ISSCI2004*, June 28 - July 1, Sevilla, Spain, paper no. ISSCI-032 (1:9).
- Velagic, J., Lacevic, B. & Hebibovic, M. (2005). On-Line identification of a robot manipulator using neural network with an adaptive learning rate, *Proceedings of 16th IFAC World Congress*, 03-08 June, Prague, Czech Republic, 2005, paper no. 2684(1-6).
- Egerstedt, M., Hu, X. & Stotsky, A. (2001). Control of mobile platforms using a virtual vehicle approach. *IEEE Transactions on Automatic Control*, Vol. 46, No. 11, pp. 1777-1882.



New Trends in Technologies

Edited by Blandna ramov

ISBN 978-953-7619-62-6

Hard cover, 242 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

This book provides an overview of subjects in various fields of life. Authors solve current topics that present high methodical level. This book consists of 13 chapters and collects original and innovative research studies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jasmin Velagic, Nedim Osmic and Bakir Lacevic (2010). Design of Neural Network Mobile Robot Motion Controller, New Trends in Technologies, Blandna ramov (Ed.), ISBN: 978-953-7619-62-6, InTech, Available from: <http://www.intechopen.com/books/new-trends-in-technologies/design-of-neural-network-mobile-robot-motion-controller>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen