Neural Network Controller for Mobile Robot Motion Control

Jasmin Velagic, Nedim Osmic, and Bakir Lacevic

***Abstract*—**In this paper the neural network-based controller is designed for motion control of a mobile robot. This paper treats the problems of trajectory following and posture stabilization of the mobile robot with nonholonomic constraints. For this purpose the recurrent neural network with one hidden layer is used. It learns relationship between linear velocities and error positions of the mobile robot. This neural network is trained on-line using the backpropagation optimization algorithm with an adaptive learning rate. The optimization algorithm is performed at each sample time to compute the optimal control inputs. The performance of the proposed system is investigated using a kinematic model of the mobile robot.

***Keywords*—**Mobile robot, kinematic model, neural network, motion control, adaptive learning rate.

I. Introduction

NEURAL NETWORKS are recommended for intelligent control as a part of well known structures with adaptive critic [1], [2], [3]. Recently, much research has been done on applications of neural networks for control of nonlinear dynamic processes [4], [5]. 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 [6]. At present, most of the works on system control using neural networks are based on multilayer feedforward neural networks with backpropogation learning or more efficient variations of this algorithm [4]. It has been shown [6] 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 $\Re^n$.

The neural network-based control of mobile robots has recently been the subject of intense research [7]. It is usual to work with kinematic models of mobile robot to obtain stable motion control laws for trajectory following or goal reaching [8], [9]. The most authors solved the problem of mobile robot motion control and stability using nonlinear backstepping algorithm [10], [11]. Fierro and Lewis [12] 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 [13] 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 [14], [15]. 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.

II. Control System of Mobile Robot

The proposed neural network control system is shown in Fig. 1.

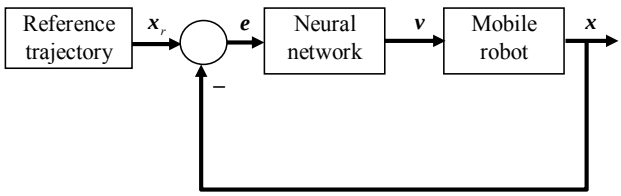

Fig. 1 Mobile robot motion control system

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

Manuscript received April 23, 2008.

J. Velagic is with the Faculty of Electrical Engineering, Department of Automatic Control and Electronics, Sarajevo, Bosnia and Herzegovina (phone: +387 33 25 07 65; fax: +387 33 25 07 25; e-mail: jasmin.velagic@etf.unsa.ba).

N. Osmic is with the Faculty of Electrical Engineering, Department of Automatic Control and Electronics, Sarajevo, Bosnia and Herzegovina (e-mail: nosmic2000@yahoo.com).

B. Lacevic is with the Faculty of Electrical Engineering, Department of Automatic Control and Electronics, Sarajevo, Bosnia and Herzegovina (phone: +387 33 250 747; fax: +387 33 250 725; e-mail: bakir.lacevic@ etf.unsa.ba).

overall control structures are described. Design of neural network controller will be presented in the next section.

A. Kinematic Model of Mobile Robot

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

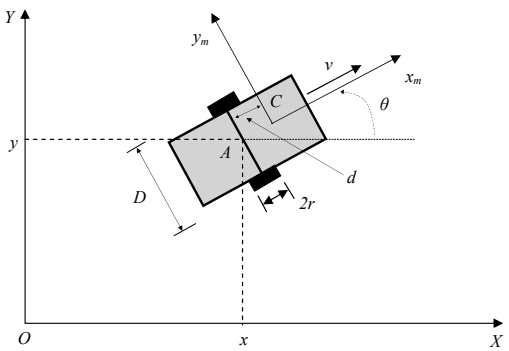

Fig. 2 The representation of a nonholonomic mobile robot

Kinematic equations of the two-wheeled mobile robot are:

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

and

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

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:

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

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:

$$\dot{x}\sin\theta + \dot{y}\cos\theta - d\dot{\theta} = 0\,. \tag{4}$$

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

B. Convergence Condition of Control System

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

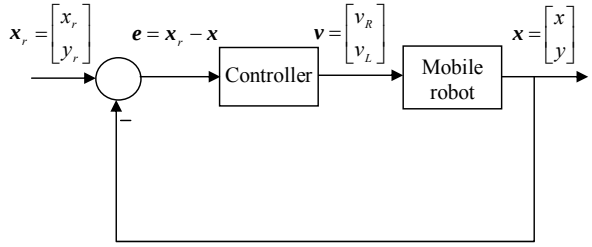

Fig. 3 Mobile robot control system

The time derivation of mobile robot output yields:

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

or

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

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

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \boldsymbol{v} = \boldsymbol{J} \begin{bmatrix} v_R \\ v_L \end{bmatrix}. \tag{7}$$

where $\boldsymbol{J}$ is Jacobian matrix.

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

$$\frac{d\boldsymbol{e}}{dt} + \boldsymbol{K}\boldsymbol{e} = 0\,, \tag{8}$$

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

This requirement is satisfied with the following control law:

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

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

$$k_1, k_2 > 0 \,, \tag{10}$$

where k_1 and k_2 are elements of the matrix $\boldsymbol{K}$, respectivelly:

$$\boldsymbol{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}. \tag{11}$$

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

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

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

$$k_1 = 102.9822,\ k_2 = 1.3536. \tag{13}$$

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

III. 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.

B. 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 then 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 [14] and [15]. In the training phase the network is presented with a series of input-answer pairs.

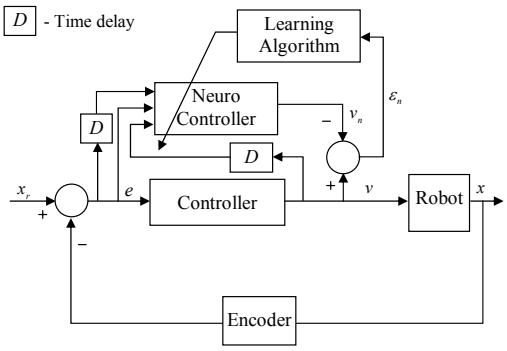

Fig. 4 Neural network learning architecture

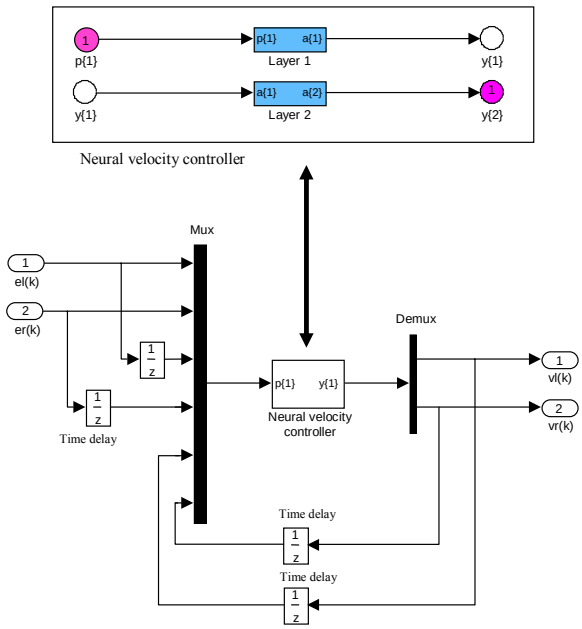

Fig. 5 Dynamical neural network

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:

$$\varepsilon_n(x) = \frac{1}{2}\sum_k (e_k - v_k)^2 \ . \tag{15}$$

Synaptic weights are updated as

$$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j v_i \ , \tag{16}$$

$$w_{kj}(t+1) = w_{kj}(t) + \eta\delta_k v_j \ , \tag{17}$$

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.

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

$$\min_{\mathbf{w}} J(\boldsymbol{w}) \cdot \tag{18}$$

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

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

$$v(t) = f(v(t-1),...,v(t-n);e(t),...,e(t-m)) \ , \tag{19}$$

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

$$v_n(\mathbf{w},t) = f_w(v(t-1),...,v(t-n);e(t),...,e(t-m)) \ , \tag{20}$$

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 $J(\boldsymbol{w})$ is defined as:

$$J(\mathbf{w},t) = \sum_{t=0}^{T} [v(t) - v_n(\mathbf{w},t)]^2 \ . \tag{21}$$

In our algorithm [14], [15] we define relationship between $\Delta J(\boldsymbol{w},t)$ and $J(\boldsymbol{w},t)$ by the relative factor $\chi(t)$:

$$\chi(t) = \frac{\Delta J(\mathbf{w},t)}{J(\mathbf{w},t)} = \frac{J(\mathbf{w},t)\text{-}J(\mathbf{w},t-1)}{J(\mathbf{w},t)} \ . \tag{22}$$

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:

$$\eta(t+1) = \eta(t)[1 - \mathrm{sgn}(\chi(t))\upsilon \cdot e^{-\chi(t)}], \quad \upsilon \in (0,1) \tag{23}$$

The learning rate is adjusted at each iteration according to Eqs. 6 and 7. 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 υ. Then, the training process in the closed control loop is performed for various values of parameter υ, $\upsilon \in [0,1]$. We adopt the value of υ 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 η=0.02 for both layers. The good results were obtained with υ=0.71.

The trajectory tracking performance obtained by adopted neural controller will be shown in the following section.

IV. Simulation Results

The effectiveness of the neural network controllers 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 [16] that has to be tracked. 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. 6-13. Results achieved in Figs. 6-10 demonstrate the good position tracking performance. Also, a desired orientation during robot motion is following in satisfactory manner, which is depicted in Fig. 11. The proposed neural controller ensures small values of the control input velocities (linear) for obtaining the reference position trajectories (Figs. 12 and 13). This mean that smaller power of DC motors is requested.

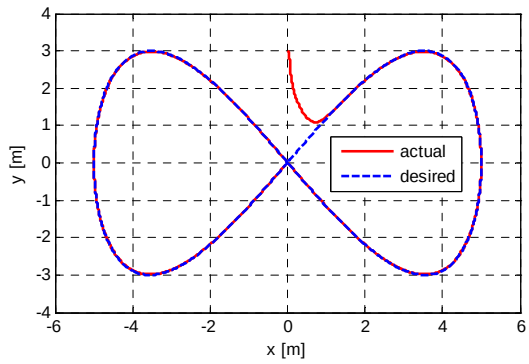

Fig. 6 Tracking the lamniscate trajectory

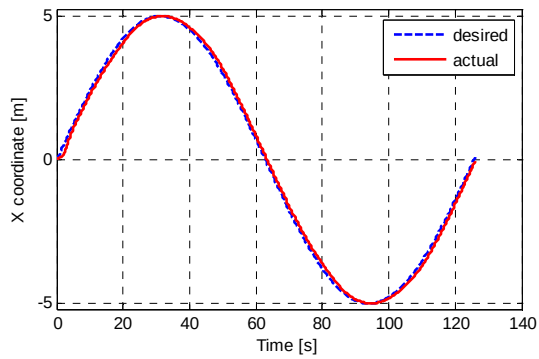

Fig. 7 Time history of *x* coordinate

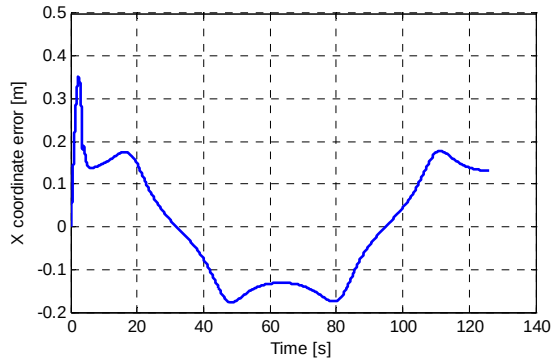

Fig. 8 *X* coordinate error

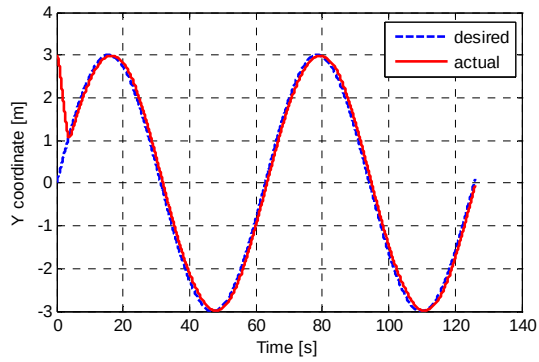

Fig. 9 Time history of *y* coordinate

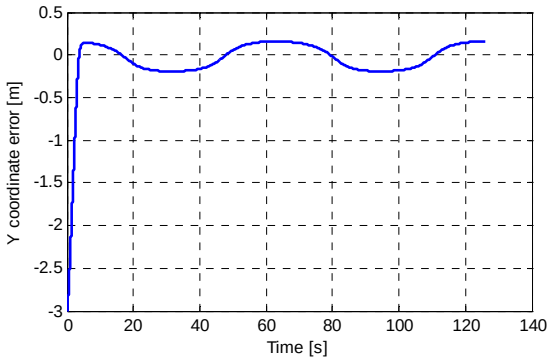

Fig. 10 *Y* coordinate error

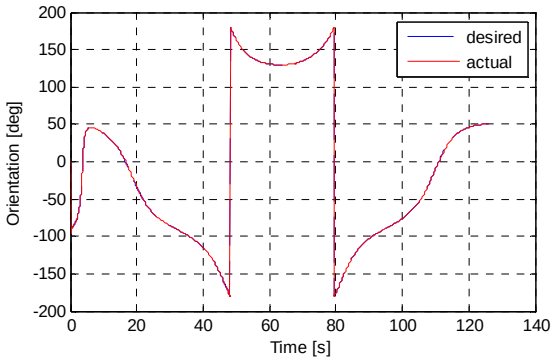

Fig. 11 Time response of robot orientation

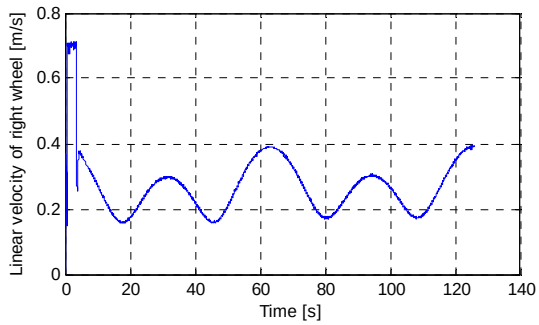

Fig. 12 Linear velocity of right wheel

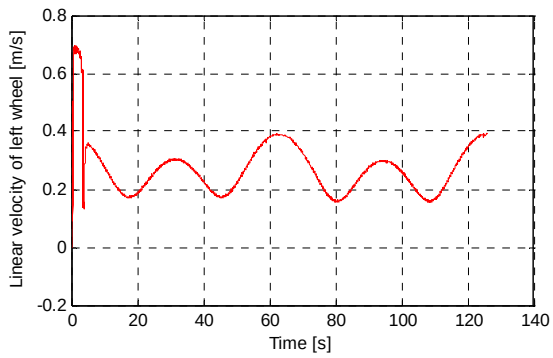

Fig. 13 Linear velocity of left wheel

V. Conclusion

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.

REFERENCES

[1] A. Lacaze, "A neural network planner that prunes its own tree," Term paper, Knowledge Engineering Department, University of Meryland, 1997.

[2] A. G. Barto, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on System, Man and Cybernetics*, vol. 13, pp. 834-846, 1983.

[3] Z. Hendzel, "Adaptive Critic Neural Networks for Motion Control of Wheeled Mobile Robot," *Nonlinear Dynamics*, vol. 50, no. 4, pp. 849-855, 2007.

[4] K.S. Narenda, and K. Pathasarathy, "Identification and control of dynamic systems using neural network," *IEEE Transaction on Neural Networks*, vol. 1, no. 1, pp. 4-27, 1990.

[5] D. Nguyen, and B. Widrow, "Neural Networks for Self- Learning Control Systems*," IEEE Control System Magazine*, vol. 10, no. 1, pp. 18-23, Feb. 1990.

[6] K. Hornik, M. Stinchombe, and H. White, "Universal Approximation of an Unknown mapping and its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, vol. 3, 1990.

[7] M. Corradini, G. Ippoliti, and S. Longhi, "Neural Networks Based Control of Mobile Robots: Development and Experimental Validation", *Journal of Robotic Systems*, vol. 20, no. 10, pp. 587-600, 2003.

[8] Z. P. Jiang, E. Lefeber, and H. Nijmeijer, "Saturated stabilization and tracking of a nonholonomic mobile robot," *Systems & Control Letters*, vol. 42, pp. 327-332, 2001.

[9] G. Ramírez,. and S. Zeghloul, "A New Local Path Planner for Nonholonmic Mobile Robot Navigation in Cluttered Environments, " in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000,p. 2058–2063..

[10] H.G. Tanner, and K.J. Kyriakopoulos, "Backstepping for nonsmooth systems, " *Automatica*, vol. 39, 2003, pp. 1259-1265.

[11] J. Velagic, B. Lacevic, and B. Perunicic, "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, Dec. 2006.

[12] R. Fierro, and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks", *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 389-400, 1998.

[13] S. X. Yang, and M. Meng, "Real-time fine motion control of robot manipulators with unknown dynamics," in: *Dynamics in Continuous, Discrete and Impulse Systems*, Series B, 2001.

[14] J. Velagic, and M. Hebibovic, "Neuro-Fuzzy Architecture for Identification and Tracking Control of a Robot," in *Proc. 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), 2004.

[15] J. Velagic, B. Lacevic, and M. Hebibovic, "On-Line Identification of a Robot Manipulator Using Neural Network with an Adaptive Learning Rate," in *Proc. 16th IFAC World Congress*, 03-08 June, Prague, Czech Republic, 2005, paper no. 2684(1-6), 2005.

[16] M. Egerstedt, X. Hu, A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control* , vol. 46, no. 11, pp. 1777-1882, 2001.