Stable Path Tracking Control of a Mobile Robot Using a Wavelet Based Fuzzy Neural Network

Joon Seop Oh, Jin Bae Park*, and Yoon Ho Choi

Abstract: In this paper, we propose a wavelet based fuzzy neural network (WFNN) based direct adaptive control scheme for the solution of the tracking problem of mobile robots. To design a controller, we present a WFNN structure that merges the advantages of the neural network, fuzzy model and wavelet transform. The basic idea of our WFNN structure is to realize the process of fuzzy reasoning of the wavelet fuzzy system by the structure of a neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of a neural network. In our control system, the control signals are directly obtained to minimize the difference between the reference track and the pose of a mobile robot via the gradient descent (GD) method. In addition, an approach that uses adaptive learning rates for training of the WFNN controller is driven via a Lyapunov stability analysis to guarantee fast convergence, that is, learning rates are adaptively determined to rapidly minimize the state errors of a mobile robot. Finally, to evaluate the performance of the proposed direct adaptive control system using the WFNN controller, we compare the control results of the WFNN controller with those of the FNN, the WNN and the WFM controllers.

Keywords: Fuzzy neural network, gradient descent method, Lyapunov stability, mobile robot, path tracking control, wavelet fuzzy model, wavelet neural network.

1. INTRODUCTION

The localization and path tracking problems for mobile robots have been given great attention by automatic control researchers in recently published literature. The motion control of mobile robots is a typical nonlinear tracking control issue and has been discussed with different control schemes such as PID, GPC, sliding mode, predictive control, etc., [1-6]. Intelligent control techniques, based on neural networks and fuzzy logic, have also been developed for path tracking control of mobile robots [7,8]. Even though these intelligent control strategies have shown their effectiveness, especially for nonlinear systems, they have certain drawbacks due to their own characteristics. While conventional neural networks have good ability for self-learning, they also have some limitations such as slow convergence, the difficulty in reaching the global minima in the parameter space, and sometimes even instability as well. In the case of fuzzy logic, it is a human-imitating logic, but lacks the ability for self-learning and self-tuning. Therefore, in the research area of intelligent control, fuzzy neural networks (FNNs) are devised to overcome these limitations and to combine the advantages of both neural networks and fuzzy logic [9-11]. This provides a strong motivation for using FNNs in the modeling and control of nonlinear systems. The wavelet fuzzy model (WFM) has the advantage of wavelet transform by constituting the fuzzy basis function (FBF) and the conclusion part to equalize the linear combination of FBF with the linear combination of wavelet functions [12-15]. The conventional fuzzy model cannot provide a satisfactory result for the transient signal. On the contrary, in the case of the WFM, the accurate fuzzy model can be obtained because the energy compaction by the unconditional basis and the description of a transient signal by wavelet basis functions are distinguished [16,17]. Therefore, we have designed a FNN structure based on wavelet, which merges the advantages of neural network, fuzzy model and wavelet. The basic idea of the wavelet based fuzzy neural network (WFNN) is to realize the process of fuzzy reasoning of the WFM by the structure of a neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of a

Manuscript received February 28, 2005; revised August 22, 2005; accepted September 5, 2005. Recommended by Editor Jae-Bok Song under the direction of Editor-in-Chief Myung Jin Chung.

Joon Seop Oh and Jin Bae Park are with the Department of Electrical and Electronic Engineering, Yonsei University, Sinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea (e-mails: {jsoh, jbpark}@control.yonsei.ac.kr).

Yoon Ho Choi is with the School of Electronic Engineering, Kyonggi University, Kyonggi-Do, Suwon-Si, 443-760, Korea (e-mail: yhchoi@kyonggi.ac.kr).

* Corresponding author.

neural network. An approach that uses adaptive learning rates is driven via a Lyapunov stability analysis to guarantee fast convergence. In this paper, we design the direct adaptive control system using the WFNN structure. The control inputs are directly obtained by minimizing the difference between the reference track and the pose of a mobile robot that is controlled through a WFNN controller. The control process is a dynamic on-line one that uses the WFNN trained by the gradient descent (GD) method. Through computer simulations, we demonstrate the effectiveness and feasibility of the proposed control method and compare the control performance of the WFNN with those of the FNN, the WFM and the wavelet neural network (WNN). The remainder of this paper is composed as follows. Section 2 illustrates the network structure and learning algorithm of the WFNN. Section 3 then develops the direct adaptive control system and adaptive learning rates for the stable network. Sections 4 and 5 present the simulation results and conclusions, respectively.

2. STRUCTURE OF WAVELET BASED FUZZY NEURAL NETWORK

While the WFM has the advantage of the wavelet transform, neural networks utilize their learning capability for automatic identification and tuning, but they have the following problems among others: (i) they need accurate input-output data, and (ii) their learning process is time-consuming. Therefore, we have designed a FNN structure based on wavelet that merges the advantages of the neural network, fuzzy modeling and wavelet. The basic idea of the WFNN is to realize the process of fuzzy reasoning of the wavelet fuzzy model by the structure of a neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of a neural network. WFNNs can automatically identify

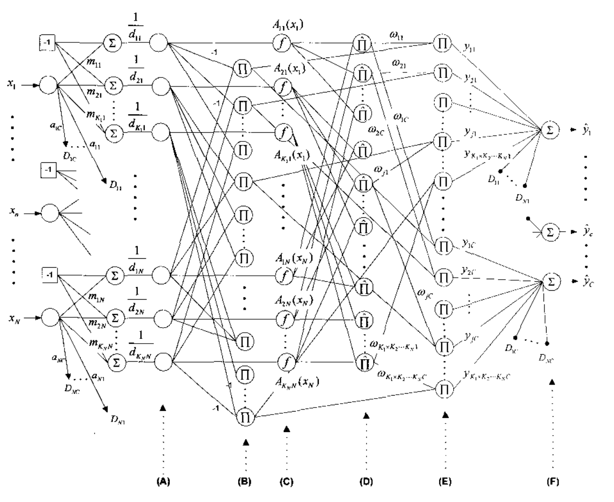


Fig. 1. WFNN structure.

the fuzzy rules by modifying the connection weights of the networks using the GD scheme. Among various fuzzy inference methods, WFNNs use the sum-product composition. The functions that are implemented by the networks must be differentiable in order to apply the GD scheme to their learning.

Fig. 1 shows the configuration of the WFNN, which has N inputs $(x_1, x_2, \cdots, x_N)$, C outputs $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_C)$, and K_n membership functions in each input x_n.

The circles and the squares in the figure represent the units of the network. The denotations a, ω, m, d and the numbers (1,–1) between the units denote the connection weights of the network. WFNN can be divided into two parts according to the fuzzy reasoning process: the premise part and the consequence part. The premise part consists of nodes (A), (C) and (D), and the consequence part consists of nodes (D) through (F). The grades of the membership functions in the premise are calculated in nodes (A) and (C). Nodes (B) and (E) are used to equalize the linear combination of FBF with the linear combination of wavelet functions for the advantage of wavelet transform. Therefore, the output node (F) is equivalent to wavelet transform. Consequently, in our WFNN structure, the output $\hat{y}_c$ is calculated as follows:

$$\hat{y}_c = \sum_{n=1}^{N} a_{nc}x_n + \sum_{j=1}^{R} B_{jc}\Phi_j , \tag{1}$$

where

$$\Phi_j = \prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n}) = \prod_{n=1}^{N} -\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right):$$

wavelet function, k_n : k-th fuzzy variable of input n, k_n : the number of fuzzy variables for the n-th input, N: the number of inputs, R : the number of fuzzy rules (the number of wavelet function), $\phi_{k_n n}(z_{k_n n})$: mother wavelet function.

The detailed descriptions of input and output nodes are as follows. Here, input and output nodes are denoted by I and O, respectively and the subscript denotes each node.

Node A:

$$O_A = \frac{x_n - m_{k_n n}}{d_{k_n n}}. \tag{2}$$

Node B:

$$O_B = \prod_{n=1}^{N} O_{A_{k_n n}} = \prod_{n=1}^{N}\left(-\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)\right). \tag{3}$$

Node C:

$$O_C = A_{k_n n}(x_n) = \exp\left(-\frac{1}{2}O_A^2\right) = \exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right). \quad (4)$$

Node D:

$$I_D = \mu_j = \prod_{n=1}^{N} O_{C_{k_n n}} = \prod_{n=1}^{N} A_{k_n n}(x_n) = \prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right). \quad (5)$$

$$O_D = \hat{\mu}_j = \frac{\mu_j}{\sum_{j=1}^{R}\mu_j} = \frac{\prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)}{\sum_{j=1}^{R}\left(\prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)\right)_j}, \quad (6)$$

where $R = \prod_{k=1}^{N} K_k$.

Node E:

$$\begin{aligned} O_E &= y_{jc} = \omega_{jc} O_D O_B \\ &= \omega_{jc}\frac{\prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)}{\sum_{j=1}^{R}\left(\prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)\right)_j}\prod_{n=1}^{N}\left(-\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)\right) \\ &= \omega_{jc}\frac{\prod_{n=1}^{N}\left(-\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)\right)\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)}{\sum_{j=1}^{R}\left(\prod_{n=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{x_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)\right)_j} \\ &= B_{jc}\Phi_j, \end{aligned} \quad (7)$$

where $B_{jc} = \frac{\omega_{jc}}{\sum_{j=1}^{R} I_{D_j}}$.

Node F:

$$O_F = \hat{y}_c = \sum_{n=1}^{N} a_{nc}x_n + \sum_{j=1}^{R} y_{jc} = \sum_{n=1}^{N} a_{nc}x_n + \sum_{j=1}^{R} B_{jc}\Phi_j. \quad (8)$$

The input space is divided into R fuzzy subspaces. The truth value of the fuzzy rule in each subspace is given by the product of the grades of the membership functions for the units in node (D). Here, μ_j is the truth value of the j-th fuzzy rule and $\hat{\mu}_j$ is the normalized value of μ_j. The fuzzy system realizes the center of gravity defuzzification formula using $\hat{\mu}_j$ in (6).

The consequence part consists of nodes (D) through (F) and the fuzzy reasoning is realized as follows:

R^j :
If x_1 *is* $A_{k_1 1}$, $\cdots$, x_n *is* $A_{k_n n}$, $\cdots$ *and* x_N *is* $A_{k_N N}$
Then $y_{jc} = \omega_{jc}$ $(j = 1, 2, \cdots, R$ *and* $c = 1, 2, \cdots, C)$,

where R^j is the j-th fuzzy rule, $A_{k_n n}$ is a fuzzy variable in the premise, and ω_{jc} is a constant. Consequently, the output value of node (F) includes the inferred values.

In our network structure, the network weight set, $\boldsymbol{\gamma} = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, is tuned to minimize the model errors via the GD method. In order to apply the GD method, the squared error function is defined as follows:

$$J = \frac{1}{2}((y_{r1} - \hat{y}_1)^2 + (y_{r2} - \hat{y}_2)^2 + \cdots + (y_{rC} - \hat{y}_C)^2, \quad (9)$$

where $\hat{\mathbf{Y}} = [\hat{y}_1\ \hat{y}_2 \cdots \hat{y}_C]$ are the output values of a WFNN and $\mathbf{Y}_r = [y_{r1}\ y_{r2} \cdots y_{rC}]$ are the desired values.

Using the GD method, the weight set, $\boldsymbol{\gamma} = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, can be tuned as follows:

$$\begin{aligned} \boldsymbol{\gamma}_p(k+1) &= \boldsymbol{\gamma}_p(k) + \Delta\boldsymbol{\gamma}_p(k) \\ &= \boldsymbol{\gamma}_p(k) - \eta\frac{\partial J}{\partial\boldsymbol{\gamma}_p(k)} \\ &= \boldsymbol{\gamma}_p(k) - \eta\frac{\partial J}{\partial\hat{\mathbf{Y}}}\frac{\partial\hat{\mathbf{Y}}}{\partial\boldsymbol{\gamma}_p(k)} \\ &= \boldsymbol{\gamma}_p(k) + \eta\cdot\mathbf{E}\cdot\hat{\boldsymbol{\upsilon}}_p, \end{aligned} \quad (10)$$

where $\mathbf{E} = [(y_{r1} - \hat{y}_1)\ (y_{r2} - \hat{y}_2) \cdots (y_{rC} - \hat{y}_C)]$, subscript p denotes each network weight and η is called the learning rate.

The gradient set of WFNN output $\hat{\mathbf{Y}}$ with respect to weight set is calculated as in (11), and each gradient of WFNN output $\hat{y}$ with respect to each weight is presented as in (12) to (14):

$$\begin{aligned} \hat{\boldsymbol{\upsilon}}_p &= \frac{\partial\hat{\mathbf{Y}}}{\partial\boldsymbol{\gamma}_p(k)} = [\hat{\boldsymbol{\upsilon}}_a\ \hat{\boldsymbol{\upsilon}}_\omega\ \hat{\boldsymbol{\upsilon}}_m\ \hat{\boldsymbol{\upsilon}}_d] \\ &= \left[\frac{\partial\hat{\mathbf{Y}}}{\partial\mathbf{a}(k)}\ \frac{\partial\hat{\mathbf{Y}}}{\partial\boldsymbol{\omega}(k)}\ \frac{\partial\hat{\mathbf{Y}}}{\partial\mathbf{m}(k)}\ \frac{\partial\hat{\mathbf{Y}}}{\partial\mathbf{d}(k)}\right], \end{aligned} \quad (11)$$

$$\hat{\upsilon}_{a_{nc}} = \frac{\partial \hat{y}_c}{\partial a_{nc}(k)} = x_n, \tag{12}$$

$$\hat{\upsilon}_{\omega_{jc}} = \frac{\partial \hat{y}_c}{\partial \omega_{jc}(k)} = \frac{\partial \sum_{j=1}^{R} y_{jc}}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}}, \tag{13}$$

$$\hat{\upsilon}_{m_{k_n n}, d_{k_n n}} = \frac{\partial \hat{y}_c}{\partial m_{k_n n}, d_{k_n n}(k)} = \frac{\partial\left(\sum_{j=1}^{H} B_{jc}\Phi_j\right)}{\partial m_{k_n n}, d_{k_n n}(k)} \tag{14}$$

$$= \sum_{h=1}^{H}\left(\omega_{jc}\left(\frac{N\dot{U}M(m_{k_n n}, d_{k_n n})}{DEN} - \frac{DE\dot{N}(m_{k_n n}, d_{k_n n})NUM}{DEN^2}\right)\right)_h,$$

where $H = \frac{\prod_{k=1}^{N} K_k}{K_N}$, $NUM = \Phi_j$, $DEN = \sum_{j=1}^{R} I_{D_j}$,

$$N\dot{U}M(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}}$$

$$= -\frac{1}{d_{k_n n}}\left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})}\left((O_{A_{k_n n}}^2 - 1)\exp\left(-\frac{1}{2}O_{A_{k_n n}}^2\right)\right)\right),$$

$$DE\dot{N}(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial DEN}{\partial z_{k_n n}}$$

$$= -\frac{1}{d_{k_n n}}\sum_{h=1}^{H}\left(\frac{\prod_{n=1}^{N} O_{C_{k_n n}}}{O_{C_{k_n n}}}\left(-O_{A_{k_n n}}\exp\left(-\frac{1}{2}O_{A_{k_n n}}^2\right)\right)\right)_h,$$

$$N\dot{U}M(d_{k_n n}) = \frac{\partial z_{k_n n}}{\partial d_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}}$$

$$= -\frac{O_{A_{k_n n}}^2}{d_{k_n n}}\left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})}\left((O_{A_{k_n n}}^2 - 1)\exp\left(-\frac{1}{2}O_{A_{k_n n}}^2\right)\right)\right),$$

$$DE\dot{N}(d_{k_n n}) = \frac{\partial z_{k_n n}}{\partial d_{k_n n}} \frac{\partial DEN}{\partial z_{k_n n}}$$

$$= -\frac{O_{A_{k_n n}}^2}{d_{k_n n}}\sum_{h=1}^{H}\left(\frac{\prod_{n=1}^{N} O_{C_{k_n n}}}{O_{C_{k_n n}}}\left(-O_{A_{k_n n}}\exp\left(-\frac{1}{2}O_{A_{k_n n}}^2\right)\right)\right)_h.$$

3. PATH TRACKING CONTROL FOR MOBILE ROBOT USING THE WFNN

3.1. Dynamic model of mobile robot

The mobile robot used in this paper is composed of two driving wheels and four casters. It is fully described by a three dimensional vector of generalized coordinates constituted by the coordinates of the midpoint between the two driving wheels, and by the orientation angle with respect to a fixed frame as shown in Fig. 2.

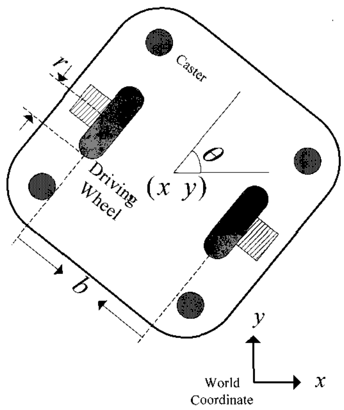


Fig. 2. Mobile robot model and world coordinate.

The equation for motion dynamics is as follows:

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \delta d_k cos(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta d_k sin(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta\theta_k \end{bmatrix}, \tag{15}$$

where $\delta d = \frac{d_r - d_l}{2}$ and $d\theta = \frac{d_r - d_l}{b}$ are linear velocity and angular velocity, respectively, and d_r, d_l and b are two incremental distances of two driving wheels and distance between these two wheels, respectively. In this model, the control input vector is represented by $\mathbf{U} = [u_d \; u_\theta]^T = [\delta d \; \delta\theta]^T$.

3.2. The direct adaptive control system using the WFNN

In our control system, the direct adaptive control system is designed using the WFNN structure. The purpose of our control system is to minimize the state error $\mathbf{E}(e_x, e_y, e_\theta)$ between the reference trajectory $\mathbf{Y}_r(x_r, y_r, \theta_r)$ and the controlled trajectory $\mathbf{Y}(x, y, \theta)$ of a mobile robot. For this purpose, the parameters of the WFNN are trained via the GD method. The overall control system is shown in Fig. 3. The WFNN controller calculates the control input $\mathbf{U} = [u_d \; u_\theta]^T$ by training the inverse dynamics of the plant iteratively. However, the updating of parameters of the WFNN through the variation rate $J(\boldsymbol{\gamma}, \mathbf{Y})$ in the GD method cannot be calculated directly. So, we train the parameters of a WFNN through the transformation of the output error of the plant. In our

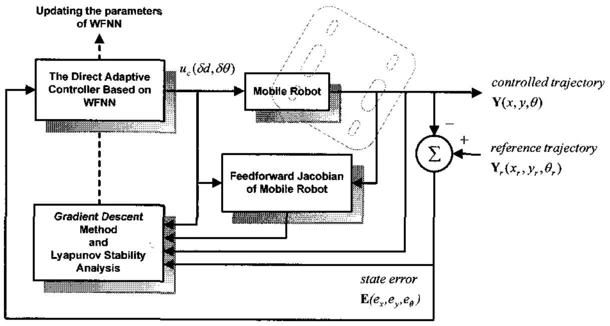


Fig. 3. Direct adaptive control system.

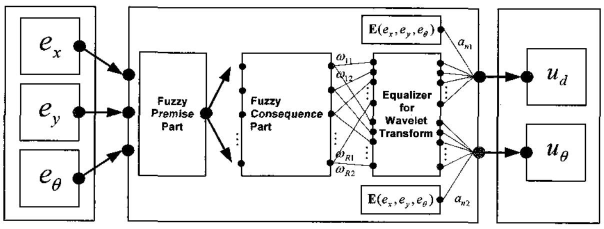


Fig. 4. WFNN structure for mobile robot.

WFNN structure, inputs, multidimensional wavelets, and two outputs are considered as shown in Fig. 4.

In this structure, inputs are composed of errors between the reference trajectory and the controlled trajectory, and outputs are control variables. Each control variable is as follows:

$$\begin{aligned} u_d &= \sum_{n=1}^{3} a_{nd} e_n + \sum_{j=1}^{R} y_{jd} = \sum_{n=1}^{3} a_{nd} e_n + \sum_{j=1}^{R} B_{jd}\Phi_j, \\ u_\theta &= \sum_{n=1}^{3} a_{n\theta} e_n + \sum_{j=1}^{R} y_{j\theta} = \sum_{n=1}^{3} a_{n\theta} e_n + \sum_{j=1}^{R} B_{j\theta}\Phi_j, \end{aligned} \tag{16}$$

where

$$B_{jc}\Phi_j = \omega_{jc} \frac{\prod_{n=1}^{3}\left(-\left(\frac{e_n - m_{k_n n}}{d_{k_n n}}\right)\right)\exp\left(-\frac{1}{2}\left(\frac{e_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)}{\sum_{j=1}^{R}\left(\prod_{n=1}^{3}\exp\left(-\frac{1}{2}\left(\frac{e_n - m_{k_n n}}{d_{k_n n}}\right)^2\right)\right)_j},$$

and $c = \{d, \theta\}$.

Training Procedure

The purpose of training the parameters of the WFNN is to minimize the state errors $\mathbf{E}(e_x, e_y, e_\theta)$. To do this, we present the following training procedure:

- Definition of the following cost function so as to train a WFNN controller based on direct adaptive control technique:

$$C = \frac{1}{2}((x_r - x)^2 + (y_r - y)^2 + (\theta_r - \theta)^2). \tag{17}$$

- Calculation of the partial derivative of the cost function with respect to the parameter set of a WFNN controller:

$$\begin{aligned} \frac{\partial C}{\partial \gamma_p} &= -e_x \frac{\partial x}{\partial \gamma_p} - e_y \frac{\partial y}{\partial \gamma_p} - e_\theta \frac{\partial \theta}{\partial \gamma_p} \\ &= -e_x \frac{\partial x}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \gamma_p} - e_y \frac{\partial y}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \gamma_p} - e_\theta \frac{\partial \theta}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial \gamma_p} \\ &= -\mathbf{E} J(u) \frac{\partial \mathbf{U}}{\partial \gamma_p}, \end{aligned} \tag{18}$$

where $e_x = x_r - x$, $e_y = y_r - y$, $e_\theta = \theta_r - \theta$, and $J(u) = \frac{\partial \mathbf{Y}}{\partial \mathbf{U}}$ is the feedforward Jacobian of a mobile robot and is as follows:

$$J(u) = \left.\begin{bmatrix} cos(\theta_k + \frac{\partial \theta_k}{2}) & -\frac{\delta d_k}{2} sin(\theta_k + \frac{\delta \theta_k}{2}) \\ sin(\theta_k + \frac{\delta \theta_k}{2}) & \frac{\delta d_k}{2} cos(\theta_k + \frac{\delta \theta_k}{2}) \\ 0 & 1 \end{bmatrix}\right|_{\theta_k = \theta_{k-1}}. \tag{19}$$

The partial derivative of the control input $\mathbf{U}$ with respect to the parameters of a WFNN controller can be calculated by using (20) and (21).

- Updating of the parameters of the WFNN via the following iterative GD method:

$$\begin{aligned} \gamma_p(k+1) &= \gamma_p(k) + \Delta\gamma_p(k) \\ &= \gamma_p(k) - \eta \frac{\partial C}{\partial \gamma_p} = \gamma_p(k) - \eta \mathbf{E} J(u) \frac{\partial \mathbf{U}}{\partial \gamma_p}, \end{aligned} \tag{20}$$

where η is the learning rate of a WFNN.

From (18) and (19), each gradient of the controller output u_c with respect to each weight is presented as follows:

$$\frac{\partial u_c}{\partial a_{nc}} = e_n, \tag{21}$$

$$\frac{\partial u_c}{\partial \omega_{jc}} = \frac{\partial \sum_{j=1}^{R} y_{jc}}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}}, \tag{22}$$

$$\begin{aligned} \frac{\partial u_c}{\partial m_{k_n n}, d_{k_n n}(k)} &= \frac{\partial\left(\sum_{j=1}^{H} B_{jc}\Phi_j\right)}{\partial m_{k_n n}, d_{k_n n}(k)} \\ &= \sum_{h=1}^{H}\left(\omega_{jc}\left(\frac{N\dot{U}M(m_{k_n n}, d_{k_n n})}{DEN} - \frac{D\dot{E}N(m_{k_n n}, d_{k_n n})NUM}{DEN^2}\right)\right)_h, \end{aligned} \tag{23}$$

and the detailed description is shown in (14).

3.3. Convergence and stability of the WFNN controller

In the update rule of (20), selection of the values for the learning rate η has a significant effect on the control performance. Generally, if η is too big, the system is unstable. For the small η, although the convergence is guaranteed, the control speed is very slow. Therefore, in order to train the WFNN effectively, adaptive learning rates, which guarantee both fast convergence and stability, must be derived. In this subsection, the specific learning rates for the type of network weights are derived based on the convergence analysis of a discrete type Lyapunov function.

Theorem 1: Let $\eta_{p,c}$ be the learning rate for the output u_c influenced by weight vector γ_p of the WFNN. $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ are defined as $\mathbf{G}_{p,c}(k)=\frac{\partial u_c(k)}{\partial \gamma_p(k)}$ and $\mathbf{G}_{p,c,\max}(k)\equiv \max_k \left\|\mathbf{G}_{p,c}(k)\right\|$, respectively, and $\|\cdot\|$ is the Euclidean norm in $\Re^n$. Here, subscripts p and c denote each weight and output, respectively. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as follows:

$$0<\eta_{p,c}<\frac{2}{\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}. \tag{24}$$

Proof: In this analysis, a discrete type Lyapunov function is selected as

$$V(k)=\frac{1}{2}\mathbf{E}^T\mathbf{E}(k), \tag{25}$$

where $\mathbf{E}(k)$ is the difference between the desired state $\mathbf{Y}_r(k)$ and the output state $\mathbf{Y}(k)$. Then, the change of Lyapunov function is obtained by

$$\begin{aligned}\Delta V(k)&=V(k+1)-V(k)\\&=\frac{1}{2}(e_x^2(k+1)-e_x^2(k)+e_y^2(k+1)-e_y^2(k)\\&\quad+e_\theta^2(k+1)-e_\theta^2(k)),\end{aligned} \tag{26}$$

where

$$\Delta e_x(k)=e_x(k+1)-e_x(k)\approx\left[\frac{\partial e_x(k)}{\partial \gamma_p(k)}\right]^T\Delta\gamma_p(k),$$

$$\Delta e_y(k)\approx\left[\frac{\partial e_y(k)}{\partial \gamma_p(k)}\right]^T\Delta\gamma_p(k),$$

$$\Delta e_\theta(k)\approx\left[\frac{\partial e_\theta(k)}{\partial \gamma_p(k)}\right]^T\Delta\gamma_p(k).$$

From (18), (19) and (20), $\Delta\gamma_p(k)$ is defined as

$$\begin{aligned}\Delta\gamma_p(k)&=-\eta_{p,c}\frac{\partial C}{\partial \gamma_p(k)}\\&=\eta_{p,c}\left(e_x(k)\frac{\partial x(k)}{\partial u_c(k)}+e_y(k)\frac{\partial y(k)}{\partial u_c(k)}+e_\theta(k)\frac{\partial \theta(k)}{\partial u_c(k)}\right)\\&\quad\cdot\left[\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right],\end{aligned} \tag{27}$$

and the error difference can be represented by

$$\begin{aligned}\Delta e_x(k)&\approx\left[\frac{\partial e_x(k)}{\partial \gamma_p(k)}\right]^T\Delta\gamma_p(k)\\&=-\left[\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right]^T\frac{\partial x(k)}{\partial u_c(k)}\eta_{p,c}\cdots\\&\quad\cdot\left(e_x(k)\frac{\partial x(k)}{\partial u_c(k)}+e_y(k)\frac{\partial y(k)}{\partial u_c(k)}+e_\theta(k)\frac{\partial \theta(k)}{\partial u_c(k)}\right)\left[\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right]\\&=-\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right\|^2\frac{\partial x(k)}{\partial u_c(k)}\cdots\\&\quad\cdot\left(e_x(k)\frac{\partial x(k)}{\partial u_c(k)}+e_y(k)\frac{\partial y(k)}{\partial u_c(k)}+e_\theta(k)\frac{\partial \theta(k)}{\partial u_c(k)}\right),\end{aligned} \tag{28}$$

where $\Delta e_y(k)$ and $\Delta e_\theta(k)$ have the identical description. Let $J_{s,c}$ be an element of the feedforward Jacobian for the state of a mobile robot with respect to the control input, where, subscripts s and c denote one state among three states of a mobile robot and the control input u_c, respectively. From (26) - (28), $\Delta V(k)$ can be represented as

$$\begin{aligned}\Delta V(k)&=V(k+1)-V(k)\\&=\frac{1}{2}\left[(e_x(k)+\Delta e_x(k))^2-e_x^2(k)+(e_y(k)+\Delta e_y(k))^2\cdots\right.\\&\quad\left.-e_y^2(k)+(e_\theta(k)+\Delta e_\theta(k))^2-e_\theta^2(k)\right]\\&=\Delta e_x(k)\left[e_x(k)+\frac{1}{2}\Delta e_x(k)\right]+\Delta e_y(k)\left[e_y(k)+\frac{1}{2}\Delta e_y(k)\right]\cdots\\&\quad+\Delta e_\theta(k)\left[e_\theta(k)+\frac{1}{2}\Delta e_\theta(k)\right]\\&=-\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right\|^2J_{x,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)\\&\quad\cdot\left[e_x(k)-\frac{1}{2}\left\|\frac{\partial u_c(k)}{\partial \gamma_p(k)}\right\|^2J_{x,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)\right]\end{aligned}$$

$$-\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2 J_{y,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)$$
$$\cdot\left[e_y(k)-\frac{1}{2}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2 J_{y,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)\right]$$
$$-\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2 J_{\theta,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)$$
$$\cdot\left[e_\theta(k)-\frac{1}{2}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2 J_{\theta,c}\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)\right]$$
$$=-\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)^2\left[\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2\cdots\right.$$
$$\left.\left(1-\frac{1}{2}\eta_{p,c}\left\|\frac{\partial u_c(k)}{\partial\gamma_p(k)}\right\|^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right)\right]$$
$$=-\left(e_x(k)J_{x,c}+e_y(k)J_{y,c}+e_\theta(k)J_{\theta,c}\right)^2\rho(k).$$

Let us define $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ as $\mathbf{G}_{p,c}(k)=\frac{\partial u_c(k)}{\partial\gamma_p(k)}$ and $\mathbf{G}_{p,c,\max}(k)\equiv\max_k\left\|\mathbf{G}_{p,c}(k)\right\|$, respectively.

Since

$$\rho(k)=\eta_{p,c}\left\|\mathbf{G}_{p,c}(k)\right\|^2\cdots$$
$$\cdot\left[1-\frac{1}{2}\eta_{p,c}\left\|\mathbf{G}_{p,c}(k)\right\|^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right]$$
$$=\eta_{p,c}\left\|\mathbf{G}_{p,c}(k)\right\|^2\cdots$$
$$\cdot\left[1-\frac{1}{2}\frac{\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\left\|\mathbf{G}_{p,c}(k)\right\|^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}{\mathbf{G}_{p,c,\max}^2(k)}\right] \quad (29)$$
$$\geq\eta_{p,c}\left\|\mathbf{G}_{p,c}(k)\right\|^2\cdots$$
$$\cdot\left[1-\frac{1}{2}\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right]>0,$$

we obtain

$$0<\eta_{p,c}<\frac{2}{\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}. \quad (30)$$

With this we complete the proof. □

Remark 1: The convergence is guaranteed as long as (29) is satisfied, *i.e.*:

$$\eta_{p,c}\left[1-\frac{1}{2}\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right]>0. \quad (31)$$

The maximum learning rate, which guarantees the fast convergence, can be obtained as $\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)=1$, *i.e.*:

$$\eta_{p,c,\max}=\frac{1}{\mathbf{G}_{p,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}, \quad (32)$$

which is half of the upper limit.

Theorem 2: Let $\eta_{p,c}=\{\eta_{a,c},\eta_{\omega,c},\eta_{m,c},\eta_{d,c}\}$ be the learning rate set for the weight set, $\boldsymbol{\gamma}=\{\mathbf{a},\boldsymbol{\omega},\mathbf{d},\mathbf{m}\}$, of WFNN, and $\mathbf{G}_{p,c}(k)$ is defined as the gradient set, $\left\{\frac{\partial u_c(k)}{\partial\mathbf{a}(k)},\frac{\partial u_c(k)}{\partial\boldsymbol{\omega}(k)},\frac{\partial u_c(k)}{\partial\mathbf{m}(k)},\frac{\partial u_c(k)}{\partial\boldsymbol{\omega}(k)},\frac{\partial u_c(k)}{\partial\mathbf{m}(k)},\frac{\partial u_c(k)}{\partial\mathbf{d}(k)}\right\}$, of WFNN output u_c with respect to the weight set. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as

$$(a)\ 0<\eta_{a,c}<\frac{2}{N\left|e_n\right|_{\max}^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)},$$
$$(b)\ 0<\eta_{\omega,c}<\frac{2}{R^2\left|O_{B_j}\right|_{\max}^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}, \quad (33)$$
$$(c)\ 0<\eta_{m,c}\cdots$$
$$<\frac{2}{\sqrt{C}\left|H\left|\omega_{jc}\right|_{\max}^2\left(\frac{\left|DEN\right|+\sqrt{H}}{\left|d_{k_nn}\right|_{\min}\left|DEN\right|^2}\right)^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right|_{\max}},$$
$$(d)\ 0<\eta_{d,c}\cdots$$
$$<\frac{2}{\sqrt{C}\left|H\left|\omega_{jc}\right|_{\max}^2\left(\frac{\left|O_{A,k_nn}^2\right|_{\max}\left(\left|DEN\right|+\sqrt{H}\right)}{\left|d_{k_nn}\right|_{\min}\left|DEN\right|^2}\right)^2\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)\right|_{\max}}.$$

Proof (a): Let us define $\mathbf{G}_{a,c,\max}(k)$ as $\mathbf{G}_{a,c,\max}(k)\equiv\max_k\ \left\|\mathbf{G}_{a,c}(k)\right\|$. Then from (30), we obtain $0<\eta_{a,c}<\frac{2}{\mathbf{G}_{a,c,\max}^2(k)\left(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\right)}$.

From the definition of Theorem 1, the maximum condition can be obtained as

$$\max_k\left\|\mathbf{G}_{a,c}(k)\right\|=\max_k\left\|\frac{\partial u_c(k)}{\partial a_{nc}(k)}\right\|=\max_k\left\|\mathbf{E}\right\|\leq\sqrt{N}\left|e_n\right|_{\max}.$$

Thus

$$\mathbf{G}_{a,c,\max}^2(k)=N\left|e_n\right|_{\max}^2,$$

where e_n is the n-th input value of WFNN and N is the number of inputs. The rest of the proof is shown in Appendix. □

Remark 2: The maximum learning rates of the WFNN, which guarantee fast convergence, are as follows:

$$\eta_{a,c,\max} = \frac{1}{N\left|e_n\right|^2_{\max}\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)},$$
$$\eta_{\omega,c,\max} = \frac{1}{R^2\left|O_{B_j}\right|^2_{\max}\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)},$$
$$\eta_{m,c,\max} = \cdots \frac{1}{\sqrt{C}\left|H\left|\omega_{jc}\right|^2_{\max}\left(\frac{|DEN|+\sqrt{H}}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)\right|_{\max}},$$
$$\eta_{d,c,\max} = \cdots \frac{1}{\sqrt{C}\left|H\left|\omega_{jc}\right|^2_{\max}\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN|+\sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)\right|_{\max}}. \quad (34)$$

4. SIMULATION RESULTS

In this Section, we present simulation results to validate the control performance of the proposed WFNN controller for the path tracking of mobile robots. Through computer simulations, we demonstrate the effectiveness and feasibility of the proposed control method and compare the control performance of the proposed WFNN controller with those of the FNN, the WFM and the WNN, respectively. The control process is a dynamic on-line process that uses the WFNN trained by the GD method. Generally, the characteristic of the network structure as a controller is very susceptible to several simulation environments such as the initial value of network weight, the sampling time, the learning rate, etc. In this computer simulation, the initial values of network weight are randomly determined and the sampling time of the control procedure is 0.01sec.

In the update rule of the GD method, selection of the values for the learning rate η has a significant effect on the control performance. So, in our control system, the learning rates are adaptively determined to rapidly minimize the state errors. The inputs of the controller are three state errors, $\mathbf{E}(e_x, e_y, e_\theta)$. The simulation environments are as shown in Table 1.

This simulation considers the tracking of a trajectory generated by the following displacements:

Linear velocity $\delta d = 20cm/\sec$,
Angular velocity $\delta\theta = 0°/\sec \quad (0 < t \le 5)$
Linear velocity $\delta d = 30cm/\sec$,
Angular velocity $\delta\theta = 59.3°/\sec \quad (5 < t \le 10)$
Linear velocity $\delta d = 30cm/\sec$,
Angular velocity $\delta\theta = -59.3°/\sec \quad (10 < t \le 15)$
Linear velocity $\delta d = 20cm/\sec$,
Angular velocity $\delta\theta = 0°/\sec \quad (15 < t \le 20)$

Fig. 5 shows the reference path and controlled path of a mobile robot using a WFNN controller. Figs. 6 and 7 present the control errors for path tracking of a

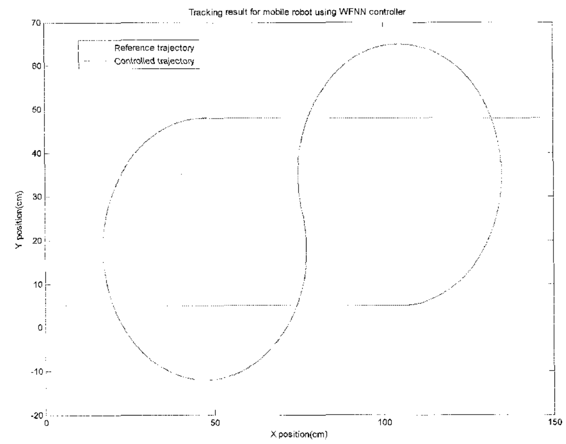


Fig. 5. Controlled path using a WFNN controller.

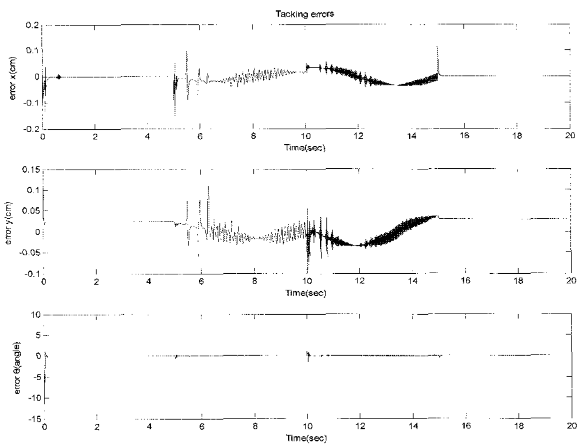


Fig. 6. Path tracking errors.

Table 1. The simulation environments.

	MF number of each input	Wavelet (Rule number)	Parameter	Learning rate
Our WFNN	3	27	78	**Adaptively** (initial value: 0.1)
Our WFNN	3	27	78	Experimentally fixed: 0.08
WFM[16]	16	16	80	Experimentally fixed: 0.011
FNN[11]	4	128	152	Experimentally fixed: 0.044
WNN[12]	*	11	94	Experimentally fixed: 0.214

Table 2. The simulation results.

	MSE		
	state $x\,[cm]$	state $y\,[cm]$	state $\theta\,[\circ]$
Our WFNN	**0.0002695**	**0.0003747**	**0.000053**
Our WFNN	0.003814	0.004329	0.002589
WFM[16]	0.05734	0.07925	0.3254
FNN[11]	0.4186	0.9527	1.08903
WNN[12]	0.009312	0.007823	0.05426

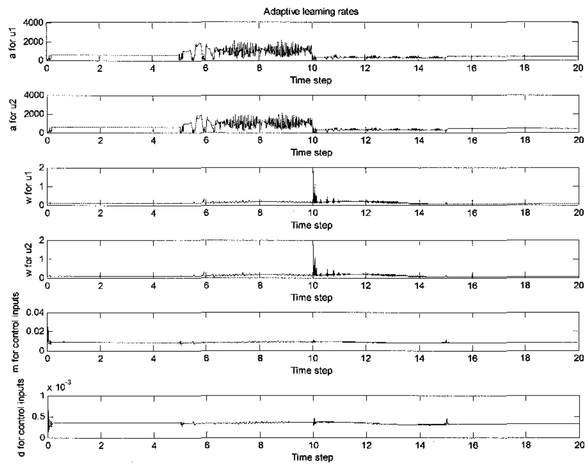


Fig. 7. Adaptive learning rates for the WFNN weights.

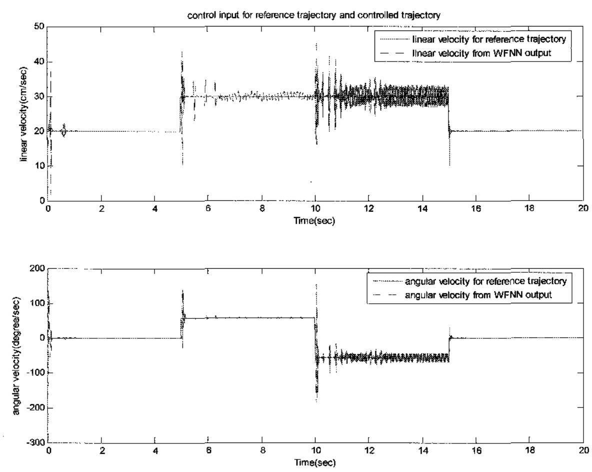


Fig. 8. Reference input and control input.

mobile robot and the adaptive learning rates for fast convergence and stability, respectively. As a result, if the control errors are changed, then the learning rates are also changed in the case of fast convergence and accuracy.

In our experiments, we use the mean squared error (MSE) as the tracking performance for comparison of performance with the FNN, the WFM and the WNN controllers. The simulation results are as indicated in Table 2. From these figures and from Table 2, we confirm that the WFNN controller works better than other controllers that use the FNN, the WFM and the

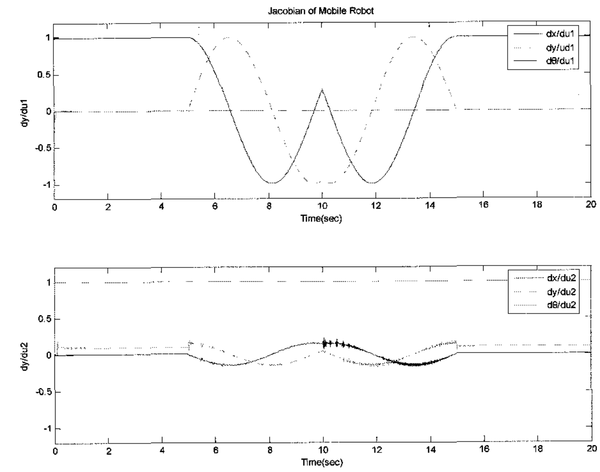


Fig. 9. Feedforward Jacobian of each control input.

WNN respectively, although the tracking errors are occurred in case that the direction is changed. In this comparison, the network structure such as the number of membership functions, the number of rules and the learning rate, is experimentally determined via numerous simulations.

Figs. 8 and 9 show the control inputs of a WFNN controller and the feedforward Jacobian of a mobile robot system, respectively.

5. CONCLUSIONS

In this paper, we have proposed a WFNN based direct adaptive control scheme for the solution of the tracking problem of mobile robots. In our control system, we have designed a FNN structure based on wavelet that merges the advantages of the neural network, fuzzy model and wavelet transform as a controller. The control signals have been directly obtained to minimize the difference between the reference track and the pose of a mobile robot via the GD method. In addition, an approach that has used adaptive learning rates for the training of the WFNN controller was driven via a Lyapunov stability analysis to guarantee fast convergence, that is, learning rates were adaptively determined to rapidly minimize the state errors of a mobile robot. Finally, to evaluate the performance of the proposed direct adaptive control system using the WFNN, we have compared the control results of the WFNN controller with those of the FNN, the WNN and the WFM controllers. As a result, we have confirmed that our WFNN controller works better than the FNN, the WNN and the WFM controllers, although tracking errors are occurred when the direction is changed.

APPENDIX A

Proof (b) of (32):

Let us define $\mathbf{G}_{\omega,c,\max}(k)$ as $\mathbf{G}_{\omega,c,\max}(k) \equiv \max_k \left\|\mathbf{G}_{\omega,c}(k)\right\|$. Then from (22) and the definition of Theorem 1, the gradient of WFNN output u_c with respect to weight ω_{jc} can be written as

$$\mathbf{G}_{\omega,c}(k) = \frac{\partial u_c}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}},$$

then

$$\left|\mathbf{G}_{\omega,c}(k)\right| = \left|\frac{\mu_j}{\sum_{j=1}^{R}\mu_j} O_{B_j}\right| \le \left|\frac{\mu_j}{\sum_{j=1}^{R}\mu_j}\right| \left|O_{B_j}\right| \le \left|O_{B_j}\right| \le \left\|O_B\right\|.$$

Since $\left|\frac{\mu_j}{\sum_{j=1}^{R}\mu_j}\right| < 1$ and $\left\|\frac{\mu_j}{\sum_{j=1}^{R}\mu_j}\right\| < \sqrt{R}$,

we obtain $\left\|\mathbf{G}_{\omega,c}(k)\right\| \le \sqrt{R}\left\|O_B\right\| \le R\left|O_{B_j}\right|_{\max}$ and have the maximum condition as follows:

$$\mathbf{G}^2_{\omega,c,\max}(k) = R^2\left|O_{B_j}\right|^2_{\max}. \tag{A1}$$

Hence, from Theorem 1 and (A1), (b) of Theorem 2 follows. □

Proof (c) and (d) of (33):

Let us define $\mathbf{G}_{m,d,c-out}(k)$ as $\mathbf{G}_{m,d,c-out}(k) \equiv \max_k \left\|\mathbf{G}_{m,d,c}(k)\right\|$. Then from (23) and the definition of Theorem 1, the gradient of WFNN output u_c with respect to weight $m_{k_n n}$ and $d_{k_n n}$ can be written as

$$\mathbf{G}_{m,d,c}(k) = \sum_{j=1}^{H} \frac{\partial B_{jc}\Phi_j}{\partial m_{k_n,n}, d_{k_n,n}(k)}$$
$$= \sum_{h=1}^{H}\left(\omega_{jc}\left(\frac{N\dot{U}M(m_{k_n n}, d_{k_n n})}{DEN} - \frac{DE\dot{N}(m_{k_n n}, d_{k_n n})NUM}{DEN^2}\right)\right).$$

Since

$$\dot{\phi}_{jn}(z_{jn}) = \left(z_{jn}^2 - 1\right)\exp\left(-\frac{1}{2}z_{jn}^2\right) < 1 \text{ and}$$
$$\dot{\exp}\left(-\frac{1}{2}z_{jn}^2\right) = -z_{jn}\exp\left(-\frac{1}{2}z_{jn}^2\right) < 1,$$

we obtain

$$\left\|\mathbf{G}_{m,c}(k)\right\| \le \cdots$$
$$\left\|\sum_{j=1}^{H}\left(\omega_{jc}\frac{N\dot{U}M(m_{k_n n})}{DEN}\right)\right\| + \left\|\sum_{j=1}^{H}\left(\omega_{jc}\frac{DE\dot{N}(m_{k_n n})NUM}{DEN^2}\right)\right\|$$
$$\le \sqrt{H}\left|\omega_{jc}\right|_{\max}\left(\frac{|DEN| + \sqrt{H}}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)$$

and

$$\left\|\mathbf{G}_{d,c}(k)\right\| \le \cdots$$
$$\left\|\sum_{j=1}^{H}\left(\omega_{jc}\frac{N\dot{U}M(d_{k_n n})}{DEN}\right)\right\| + \left\|\sum_{j=1}^{H}\left(\omega_{jc}\frac{DE\dot{N}(d_{k_n n})NUM}{DEN^2}\right)\right\|$$
$$\le \sqrt{H}\left|\omega_{jc}\right|_{\max}\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN| + \sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right).$$

Therefore, we obtain each maximum condition as follows:

$$\mathbf{G}^2_{m,c-out}(k) = H\left|\omega_{jc}\right|^2_{\max}\left(\frac{|DEN| + \sqrt{H}}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2, \tag{A2}$$

$$\mathbf{G}^2_{d,c-out}(k) = \cdots$$
$$H\left|\omega_{jc}\right|^2_{\max}\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN| + \sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2. \tag{A3}$$

While weights **a** and **ω** have an effect on only one connected output, weights **m** and **d** have an effect on all outputs. Therefore, for convergence according to the effect of weights **m** and **d**, additional expansion is needed.

Let us define $\mathbf{G}^2_{m,d,c,\max}(k)$ as $\mathbf{G}^2_{m,d,c,\max}(k) \equiv \max_k \left\|\mathbf{G}^2_{m,d,c-out}(k)\left(J^2_{x,c} + J^2_{y,c} + J^2_{\theta,c}\right)\right\|$. Here $\mathbf{G}^2_{m,d,c-out}(k)$ is the maximum condition for each output u_c according to the effect of the weight $\{\mathbf{m}, \mathbf{d}\}$ and $\mathbf{G}^2_{m,d,c,\max}(k)$ is the maximum condition for output $\mathbf{U}$. Then we obtain

$$\mathbf{G}^2_{m,d,c,\max}(k) \le \sqrt{C}\left|\mathbf{G}^2_{m,d,c-out}(k)\left(J^2_{x,c} + J^2_{y,c} + J^2_{\theta,c}\right)\right|_{\max}.$$

Thus,

$$\mathbf{G}^2_{m,c,\max}(k) = \sqrt{C}\cdots$$

$$\cdot\left| H\left|\omega_{jc}\right|^2_{\max}\left(\frac{|DEN|+\sqrt{H}}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)\right|_{\max}, \quad \text{(A4)}$$

$$\mathbf{G}^2_{d,c,\max}(k) = \sqrt{C}\left|H\left|\omega_{jc}\right|^2_{\max}\cdots\right.$$

$$\left.\cdot\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN|+\sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2\left(J^2_{x,c}+J^2_{y,c}+J^2_{\theta,c}\right)\right|_{\max}. \quad \text{(A5)}$$

Therefore, if the maximum conditions in (A2) and (A3) are substituted by (A4) and (A5), respectively from Theorem 1, (A4) and (A5), (c) and (d) of Theorem 2 follow. □

REFERENCES

[1] R. M. Desantis, “Path-tracking for car-like robots with single and double steering,” *IEEE Trans. on Vehicular Technology*, vol. 44, no. 2, pp. 366-377, 1995.

[2] X. Yang, K. He, M. Guo, and B. Zhang, “An intelligent predictive control approach to path tracking problem of autonomous robot,” *Proc. of IEEE Conf. on Systems, Man, and Cybernetics*, vol. 4, pp. 350-355, 1998.

[3] C. C. Wit, H. Khennouf, C. Samson, and O. J. Sordalen, “Nonlinear control design for mobile robots,” *Recent Trends in Mobile Robots, World Scientific*, pp. 121-156, 1993.

[4] Z. P. Jiang and H. Nijmeijer, “Tracking control of mobile robots: A case study in backstepping,” *Automatica*, vol. 33. no. 7, pp. 1393-1399, 1997.

[5] J. M. Yang and J. H. Kim, “Sliding mode motion control of nonholonomic mobile robots,” *IEEE Control Systems*, vol. 19, no. 2, pp. 15-23, 1990.

[6] G. Dongbing and H. Huosheng, “Wavelet neural work based predictive control for mobile robots,” *Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 3544-3549, 2000.

[7] M. L. Corradini, G. Ippoliti, S. Longhi, and S. Michelini, “Neural networks inverse model approach for the tracking problem of mobile robot,” *Proc. of RAAD* 2000, pp. 17-22, 2000.

[8] M. L. Corradini, G. Ippoliti, and S. Longhi, “The tracking problem of mobile robots: Experimental results using a neural network approach,” *Proc. of WAC*, pp. 33-37, 2000.

[9] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On identification of structures in premise of a fuzzy model using a fuzzy neural network," *Proc. of IEEE Int. Conf. on Fuzzy Systems*, pp. 661-666. 1993.

[10] T. Hasegawa, S. Horikawa, T. Furuhashi, and Y. Uchikawa, “On design of adaptive fuzzy neural networks and description of its dynamical behavior,” *Fuzzy Sets and Systems,* vol. 71, no. 1, pp. 3-23, 1995.

[11] J. T. Choi and Y. H. Choi, “Fuzzy neural network based predictive control of chaotic nonlinear systems,” *IEICE Trans. on Fundamentals*, vol. E87-A, no. 5, pp. 1270-1279, 2004.

[12] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, “Training wavelet networks for nonlinear dynamic input-output modeling,” *Neurocomputing,* vol. 20. pp. 173-188, 1998.

[13] T. Kugarajah and Q. Zhang, “Multidimensional wavelet frames,” *IEEE Trans. on Neural Network*, vol. 6, no. 6, pp. 1552-1556, 1995.

[14] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet transform,” *IEEE Trans. on Pattern Anal. Mach. Intelligence*, vol. 11, no. 7, 674-693, 1989.

[15] Y. C. Pati and P. S. Krishnaprasad, “Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations,” *IEEE Trans. on Neural Network*, vol. 4, no. 1, pp. 73-85, 1998.

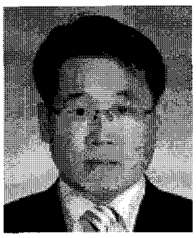
[16] C. K. Lin and S. D. Wang, “Fuzzy modeling using wavelet transforms,” *Electronics Letters*, vol. 32, pp. 2255-2256, 1996.

[17] C. K. Lin and S. D. Wang, “Constructing a fuzzy model from wavelet transforms,” *Proc. of Fuzzy Systems Symposium, Soft Computing in Intelligent Systems and Information Processing*, pp. 394-399, 1996.

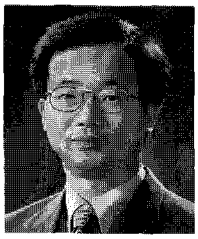
[18] Q. Zhang and A. Benveniste, “Wavelet networks,” *IEEE Trans. on Neural Network*, vol. 3, no. 6, pp. 889-898, 1992.

[19] Q. Zhang, “Using wavelet network in nonparametric estimation,” *IEEE Trans. on Neural Network*, vol. 8, no. 2, pp. 227-236, 1997.

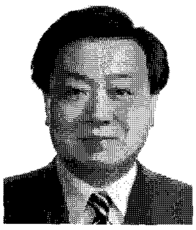
[20] J. S. Oh, S. J. You, J. B. Park, and Y. H. Choi, "The modeling of chaotic nonlinear system using wavelet based fuzzy neural network," *Proc. of Int. Conf. on Control, Automation and Systems*, pp. 635-639, 2004.



Joon Seop Oh received the B.S. and M.S. degrees in Electronic Engineering from Kyonggi University, Suwon, Korea, in 1997 and 1999, respectively. Currently, he is pursuing a Ph.D. degree in the Dept. of Electrical and Electronic Engineering at Yonsei University, Seoul, Korea. He has worked in several areas including those of mobile robot navigation, map construction, fuzzy logic control, neural networking, and Kalman filtering.



Jin Bae Park received the B.S. degree in Electrical Engineering from Yonsei University, Seoul, Korea, and the M.S. and Ph.D. degrees in Electrical Engineering from Kansas State University, Manhattan, in 1977, 1985 and 1990, respectively. Since 1992 he has been with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, where he is currently a Professor. His research interests include robust control and filtering, nonlinear control, mobile/biped robots, fuzzy logic control, neural networks, genetic algorithms, and Hadamard-transform spectroscopy. He has also served as the Director for the Transactions of the Korean Institute of Electrical Engineers (KIEE) (1998-2003) and as the Vice-President for the Institute of Control, Automation, and Systems Engineers (2004).



Yoon Ho Choi received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea, in 1980, 1982 and 1991, respectively. Since 1993, he has been with the School of Electronic Engineering at Kyonggi University, where he is currently a Professor. From 2000 to 2002, he was with the Department of Electrical Engineering at Ohio State University, where he was a Visiting Professor. He served as Director for the Institute of Control, Automation and Systems Engineers from 2003-2004. He is currently serving as the Director for The Information and Control Society of KIEE. His research interests include intelligent control, mobile robot, biped robot, web-based control systems and wavelet transform.