

# Path Tracking Control Using a Wavelet Based Fuzzy Neural Network for Mobile Robots

Joon Seop Oh\* and Yoon Ho Choi\*\*

\*Department of Electrical & Electronic Engineering, Yonsei University

\*\*School of Electronic Engineering, Kyonggi University

## Abstract

In this paper, we present a novel approach for the structure of Fuzzy Neural Network(FNN) based on wavelet function and apply this network structure to the solution of the tracking problem for mobile robots. Generally, the wavelet fuzzy model(WFM) has the advantage of the 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. However, it is very difficult to identify the fuzzy rules and to tune the membership functions of the fuzzy reasoning mechanism. Neural networks, on the other hand, utilize their learning capability for automatic identification and tuning. Therefore, we design a wavelet based FNN structure(WFNN) that merges these advantages of neural network, fuzzy model and wavelet transform. The basic idea of our wavelet based FNN is to realize the process of fuzzy reasoning of 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. And our network can automatically identify the fuzzy rules by modifying the connection weights of the networks via the gradient descent scheme. To verify the efficiency of our network structure, we evaluate the tracking performance for mobile robot and compare it with those of the FNN and the WFM.

**Key Words :** Wavelet Based Fuzzy Neural Network, Direct Adaptive control, Mobile Robot, Gradient Descent Method

## 1. Introduction

The localization and path tracking problems for mobile robots have been given great attention by automatic control researchers in the recent literatures. 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. Even though these intelligent control strategies have shown effectiveness, especially for nonlinear systems, they have certain drawbacks derived from their own characteristics. While conventional neural networks have good ability of self-learning, they also have some limitations such as slow convergence, the difficulty in reaching the global minima in the parameter space, and sometimes instability as well[7][8]. In the case of fuzzy logic, it is a human-imitating logic, but lacks the ability of self-learning and self-tuning. Therefore, in the research on the intelligent control, FNNs are devised to overcome these limitations and to combine the advantages of both neural networks and fuzzy logic[9][10]. This provides a strong motivation for using FNNs for modeling and controlling nonlinear systems. And the wavelet fuzzy model(WFM) has the advantage of the wavelet transform by constituting FBF, the conclusion part to equalize the linear combination of FBF with the linear combination of wavelet

functions and modifying fuzzy model to be equivalent to wavelet transform. The conventional fuzzy model cannot give the satisfactory result for the transient signal. On the contrary, in the wavelet fuzzy model, 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. Therefore, we design a FNN structure based on wavelet that merges these advantages of neural network, fuzzy model and wavelet. The basic idea of WFNN is to realize the process of fuzzy reasoning of 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 the fuzzy rules by modifying the connection weights of the networks using the gradient descent(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 gradient descent scheme to their learning. In this paper, we design the direct adaptive control system using 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. The control process is a dynamic on-line process that uses the WFNN trained by GD method. Through computer simulations, we demonstrate the effectiveness and the feasibility of the proposed control method.

## 2. Structure of Wavelet Based FNN

### 2.1 Wavelet Frames and Wavelet Networks

Wavelet networks were first presented in the framework of static modeling[14]-[17]. Generally, the property of wavelet functions can be expressed as follows: any function of  $L^2(R)$  can be approximated to any prescribed accuracy with a finite sum of wavelets. Therefore, wavelet networks can be considered as an alternative to neural and radial basis function networks. Wavelet frames, on the other hand, are constructed by simple operations for translation and dilation for a single fixed function called the mother wavelet. The following relation derives wavelet  $\phi_j(x)$  from its mother wavelet  $\phi(z_j)$ .

$$\phi_j(x) = \phi\left(\frac{x - m_j}{d_j}\right) = \phi(z_j) \quad (1)$$

where the translation factor  $m_j$  and the dilation factor  $d_j$  are real numbers in  $R$  and  $R_+^*$ , respectively. The family of functions generated by  $\phi$  can be defined as

$$\Omega_c = \frac{1}{\sqrt{d_j}} \phi\left(\frac{x - m_j}{d_j}\right), m_j \in R \text{ and } d_j \in R_+^* \quad (2)$$

A family  $\Omega_c$  is said to be a frame of  $L^2(R)$  if the following equation is satisfied.

$$C f^2 \leq \sum_{j, \phi_j \in \Omega_c} \langle \phi_j, f \rangle^2 \leq C f^2, 0 < C < +\infty \quad (3)$$

where,  $f$  and  $\langle f, g \rangle$  denote the norm of function  $f$  and the inner product of functions  $f$  and  $g$ , respectively. Families of wavelet frames of  $L^2(R)$  are universal approximators. For the modeling of multivariable processes, multidimensional wavelets must be defined. A multidimensional wavelet function is represented with tensor product of single dimensional wavelet function as follows:

$$\phi(x) = \phi_1(x_1) \cdots \phi_n(x_n) \quad (4)$$

Assuming that single dimensional wavelet transform is separated into  $n$  orthogonal direction elements, Fourier transform of each term in Eq. (4) is substituted for itself.

$$\hat{\phi}(w) = \hat{\phi}_1(w_1) \cdots \hat{\phi}_n(w_n) \quad (5)$$

where,  $\hat{\phi}(w)$  is Fourier transform of  $\phi(x)$ .

$$\int \frac{\phi_i(w_i)}{w_i} dw_i < \infty \quad (6)$$

It is proven that admissibility condition of Eq. (6) must be satisfied. On condition of attenuation, Eq. (6) can satisfy the following for  $\phi_i(x_i)$  that converges into 0 for  $+\infty$  and  $-\infty$ .

$$\int \phi_i(x_i) dx_i = 0 \quad (7)$$

The condition of Eq. (3) should be satisfied to be wavelet frames as well. Therefore,  $\phi_i(x_i)$  which satisfied Eqs. (3) and (7) should be set as wavelet frame. In this paper, the first-ordered differential form of the Gaussian probability density function is employed as a mother wavelet function that satisfies both of the conditions as Eq. (8).

$$\phi(z) = -z \exp\left(-\frac{1}{2}z^2\right) \quad (8)$$

And we use multidimensional wavelets constructed as the product of  $N_i$  scalar wavelets as follow.

$$\Phi_j(x) = \prod_{k=1}^{N_i} \phi(z_{jk}) \text{ with } z_{jk} = \frac{x - m_{jk}}{d_{jk}} \quad (9)$$

### 2.2 Wavelet Based Fuzzy Neural Network

Generally, the wavelet fuzzy model has the advantage of the wavelet transform by constituting FBF and conclusion part to equalize the linear combination of FBF with the linear combination of wavelet functions. The conventional fuzzy model cannot give the satisfactory result for the transient signal. On the contrary, in the wavelet fuzzy model, the accurate fuzzy model can be obtained because the energy compaction by unconditional basis and the description of a transient signal by wavelet basis functions are distinguished. However, it is very difficult to identify the fuzzy rules and to tune the membership functions of the fuzzy reasoning mechanism. Neural networks, on the other hand, utilize their learning capability for automatic identification and tuning, but they have the following problems (i) they need accurate input-output data (ii) their learning process is time-consuming, to mention a few. Therefore, we design a FNN structure based on wavelet that merges these advantages of neural network, fuzzy modeling and wavelet transform. The basic idea of WFNN is to realize the process of fuzzy reasoning of 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 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 WFNN, which has  $N$  inputs  $(x_1, x_2, \dots, x_N)$ , and  $C$  outputs  $(y_1, y_2, \dots, y_c)$ , 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  $m, d, w$  and the numbers 1, -1 between the units denote connection weights of the network.

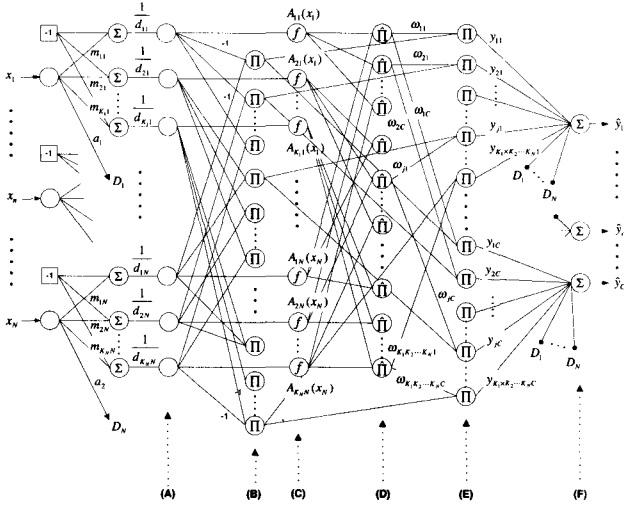


Fig. 1 Network structure of WFNN

WFNN can be divided into two parts according to the process of the fuzzy reasoning: the premise part and the consequence parts. The premise part consists of nodes (A), (C), (D) and the consequence parts consist of nodes (D) through (F). The grades of the membership functions in the premise are calculated in nodes (A) and (C). The 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 by constituting FBF and conclusion part. Therefore, the output node (F) is equivalent to wavelet transform. Consequently, in our WFNN structure, the output  $\hat{y}_c$  is calculated as follows:

$$\hat{y}_c = \sum_{n=1}^N a_{nc} x_n + \sum_{j=1}^R B_{jc} \Phi_j \quad (10)$$

where,

$$\begin{aligned} \Phi_j(x) &= \prod_{n=1}^N \phi_n(z) \\ &= \prod_{n=1}^N -\left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right) \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right) \end{aligned}$$

Mother wavelet:  $\phi(z) = -z \exp(-\frac{1}{2}z^2)$ ,  $z = \frac{x-m}{d}$

$k_n$ :  $k$ -th fuzzy variable of  $n$ -th input,  $N$ : input Num.,  
 $K_n$ : fuzzy variable Num of input  $n$ ,  $R$ : wavelet Num.

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

Node A:

$$O_A = \frac{x_n - m_{k_n,n}}{d_{k_n,n}} \quad (11)$$

Node B:

$$O_B = \prod_{n=1}^N O_{A_{k_n}} = \prod_{n=1}^N \left(-\left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)\right) \quad (12)$$

Node C:

$$\begin{aligned} O_C &= A_{k_n}(x_n) = \exp\left(-\frac{1}{2} O_A^2\right) \\ &= \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right) \end{aligned} \quad (13)$$

Node D:

$$\begin{aligned} I_D = \mu_j &= \prod_{n=1}^N O_{C_{k_n}} = \prod_{n=1}^N A_{k_n}(x_n) \\ &= \prod_{n=1}^N \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right) \\ O_D = \hat{\mu}_j &= \frac{\mu_j}{R} \\ &= \frac{\sum_{j=1}^R \mu_j}{R} \end{aligned} \quad (14)$$

$$\begin{aligned} &= \frac{\prod_{n=1}^N \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right)}{\sum_{n=1}^R \left(\prod_{n=1}^N \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right)\right)_j} \\ R &= \prod_{k=1}^N K_k \end{aligned}$$

Node E:

$$\begin{aligned} O_E = y_{jc} &= w_{jc} O_D O_B \\ &= w_{jc} \frac{\mu_j}{R} \prod_{n=1}^N \left(-\left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)\right) \\ &= w_{jc} \frac{\prod_{n=1}^N \left(-\left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)\right) \exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right)}{\sum_{j=1}^R \prod_{n=1}^N \left(\exp\left(-\frac{1}{2} \left(\frac{x_n - m_{k_n,n}}{d_{k_n,n}}\right)^2\right)\right)_j} \\ &= B_{jc} \Phi_j \quad \text{where, } B_{jc} = \frac{w_{jc}}{R} \frac{1}{\sum_{j=1}^R I_D} \end{aligned} \quad (15)$$

Node F:

$$\begin{aligned} O_F = \hat{y}_c &= \sum_{n=1}^N a_{nc} x_n + \sum_{j=1}^R y_{jc} \\ &= \sum_{n=1}^N a_{nc} x_n + \sum_{j=1}^R B_{jc} \Phi_j \end{aligned} \quad (16)$$

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). Where,  $\mu_j$  is the truth value of the  $j$ -th fuzzy rule and  $\hat{\mu}_j$  is the normalized value of  $\mu_j$ . Fuzzy system realizes the center of gravity defuzzification formula using  $\hat{\mu}_j$  in Eq. (14). The consequence parts consist of nodes (D) through (F) and the fuzzy reasoning is realized as:

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

where,  $R^j$  is the  $j$ -th fuzzy rule,  $A_{k_n,n}$  is fuzzy variable in the premise,  $\omega_{jc}$  is a constant. Consequently, the output value of node (F) includes the inferred values. The weights  $\omega_{jc}$  are modified to identify fuzzy rules using the gradient descent method. In order to apply the gradient descent method, the squared error function is defined as follows:

$$J = \frac{1}{2} ((y_{r1} - \hat{y}_1)^2 + (y_{r2} - \hat{y}_2)^2 + \dots + (y_{rc} - \hat{y}_c)^2) \quad (17)$$

where,  $\hat{y}_c$  is the output value of WFNN and  $y_{rc}$  is the desired value. Using the GD method, the parameter set,  $\gamma_c = \{a_n, \omega_{jc}\}$ , can be tuned as follows:

$$\begin{aligned} \gamma_c(k+1) &= \gamma_c(k) + \Delta\gamma_c(k) \\ &= \gamma_c(k) - \eta \frac{\partial J}{\partial \gamma_c(k)} \\ &= \gamma_c(k) - \eta \frac{\partial J}{\partial \hat{y}_c} \frac{\partial \hat{y}_c}{\partial \gamma_c(k)} \\ &= \gamma_c(k) + \eta (y_{rc} - \hat{y}_c) \hat{v}_\gamma \end{aligned} \quad (18)$$

where,  $\hat{v}_\gamma = \frac{\partial \hat{y}_c}{\partial \gamma_c(k)}$

$$\begin{aligned} \hat{v}_\omega &= \frac{\partial \hat{y}_c}{\partial \omega_{jc}(k)} = \frac{\partial \sum_{j=1}^R y_{jc}}{\partial \omega_{jc}(k)} \\ &= \frac{\prod_{n=1}^N \left( -\left( \frac{x_n - m_{k_n,n}}{d_{k_n,n}} \right) \right) \exp\left(-\frac{1}{2} \left( \frac{x_n - m_{k_n,n}}{d_{k_n,n}} \right)^2\right)}{\sum_{j=1}^R \left( \prod_{n=1}^N \exp\left(-\frac{1}{2} \left( \frac{x_n - m_{k_n,n}}{d_{k_n,n}} \right)^2\right) \right)_j} \\ \hat{v}_a &= \frac{\partial \hat{y}_c}{\partial a_c(k)} = x_n \end{aligned}$$

and  $\eta$  is called the learning rate.

### 3. Path Tracking Control for Mobile Robot Using WFNN

#### 3.1 Dynamic Model of Mobile Robot

The mobile robot used in this paper is composed of two driving wheels and four casters and 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.

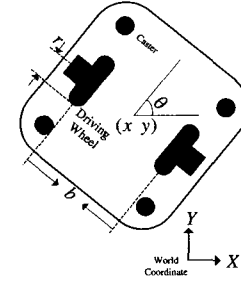


Fig. 2 Mobile robot model and world coordinate

We have the equation for motion dynamics as follows:

$$\begin{aligned} \begin{matrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{matrix} &= \begin{matrix} X_k \\ Y_k \\ \theta_k \end{matrix} + \begin{matrix} \delta d_k \cos\left(\theta_k + \frac{\delta \theta_k}{2}\right) \\ \delta d_k \sin\left(\theta_k + \frac{\delta \theta_k}{2}\right) \\ \delta \theta_k \end{matrix} \quad (19) \\ \delta d &= \frac{d_r - d_l}{2} \quad d\theta = \frac{d_r - d_l}{b} \end{aligned}$$

where,  $\delta d$  and  $\delta \theta$  are linear velocity and angular velocity.

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 expressed by  $u_k = [\delta d_k, \delta \theta_k]^T$ .

#### 3.2 The Direct Adaptive Control System Using WFNN

In our control system, we design the direct adaptive control system using the WFNN structure. The purpose of our control system is to minimize the state errors  $E(e_x, e_y, e_\theta)$  between reference trajectory and controlled trajectory of a mobile robot. For this purpose, we train the WFNN's parameters using the GD method. The overall control system is shown in Fig. 3. A WFNN controller calculates the control input  $u_k = [\delta d_k, \delta \theta_k]^T$  by training the inverse dynamics of plant iteratively. But, the updating of WFNN parameters through the variation rate  $J(\gamma, Y)$  in the GD method cannot be calculated directly. So, we train the parameters of WFNN through the transformation of the output error of plant. In our WFNN structure, inputs, multidimensional wavelets, and two outputs are considered as shown in Fig. 4.

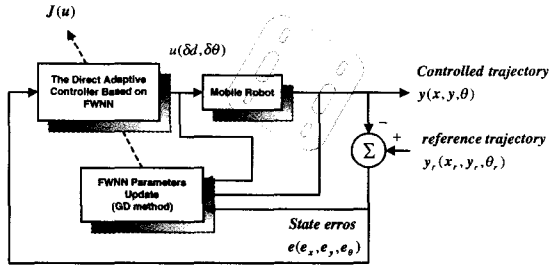


Fig. 3 Direct adaptive control system

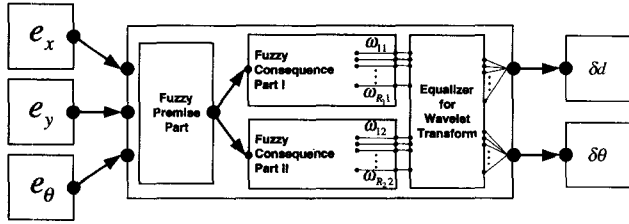


Fig. 4 WFNN structure for a mobile robot

In this structure, inputs are composed of errors between reference trajectory and controlled trajectory, and outputs are control variables. Each control variable is as shown in Eq. (20).

$$\begin{aligned} \delta d_k &= \sum_{n=1}^3 a_{nd} e_n + \sum_{j=1}^R \hat{y}_{jd} = \sum_{n=1}^3 a_{nd} e_n + \sum_{j=1}^R B_{jd} \Phi_j \\ \delta \theta_k &= \sum_{n=1}^3 a_{n\theta} e_n + \sum_{j=1}^R \hat{y}_{j\theta} = \sum_{n=1}^3 a_{n\theta} e_n + \sum_{j=1}^R B_{j\theta} \Phi_j \end{aligned} \quad (20)$$

where,

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

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

**Training Procedure :**

• Definition of the following cost function so as to train the WFNN controller based on direct adaptive control technique.

$$C = \frac{1}{2} \{ (x_r - x)^2 + (y_r - y)^2 + (\theta_r - \theta)^2 \} \quad (21)$$

• Calculation of the partial derivative of the cost function with respect to the parameter set of a WFNN controller.

$$\begin{aligned} \frac{\partial C}{\partial \gamma_c} &= -e_x \frac{\partial x}{\partial \gamma_c} - e_y \frac{\partial y}{\partial \gamma_c} - e_\theta \frac{\partial \theta}{\partial \gamma_c} \\ &= -e_\theta \frac{\partial x}{\partial u} \frac{\partial u}{\partial \gamma_c} - \dots = -E_Y J(u) \frac{\partial u}{\partial \gamma_c} \end{aligned} \quad (22)$$

where,  $e_x = x_r - x$ ,  $e_y = y_r - y$ ,  $e_\theta = \theta_r - \theta$  and

$J(u) = \frac{\partial Y}{\partial u}$  is the feedforward Jacobian of plant and is as follows.

$$J(u) = \begin{bmatrix} \cos\left(\theta_k + \frac{\delta\theta_k}{2}\right) & -\frac{\delta d_k}{2} \sin\left(\theta_k + \frac{\delta\theta_k}{2}\right) \\ \sin\left(\theta_k + \frac{\delta\theta_k}{2}\right) & \frac{\delta d_k}{2} \cos\left(\theta_k + \frac{\delta\theta_k}{2}\right) \\ 0 & 1 \end{bmatrix} \quad (23)$$

The partial derivative of the control input  $u$  with respect to the parameters of a WFNN controller can be calculated by using Eqs. (24) and (25).

• Updating of WFNN parameters. The minimization is performed by the following iterative GD method.

$$\gamma_c(n+1) = \gamma_c(n) - \eta \frac{\partial C}{\partial \gamma_c} \quad (24)$$

where,  $\eta$  is the learning rate of a WFNN.

From Eqs (22) and (23),  $\frac{\partial u}{\partial \gamma_c}$  is the gradient of the controller output  $u$ , with respect to parameters set  $\gamma_c$ .

$$\begin{aligned} \frac{\partial u_c}{\partial \omega_{jc}} &= \frac{\partial \sum_{j=1}^R y_{jc}}{\partial \omega_{jc}} \\ &= \frac{\prod_{n=1}^3 \left( -\left( \frac{e_n - m_{k_n n}}{d_{k_n n}} \right) \right) \exp\left( -\frac{1}{2} \left( \frac{e_n - m_{k_n n}}{d_{k_n n}} \right)^2 \right)}{\sum_{r=1}^R \prod_{n=1}^3 \exp\left( -\frac{1}{2} \left( \frac{e_n - m_{k_n n}}{d_{k_n n}} \right)^2 \right)_j} \end{aligned} \quad (25)$$

$$\frac{\partial u_c}{\partial a_{nc}} = e_n$$

where, subscript  $c$  and  $n$  denote control input and input of WFNN, respectively.

## 4. Simulation Results

In this section, we present simulation results to validate the control performance of proposed WFNN controller for the path tracking of mobile robots. The control objective for our tracking system is to minimizing the difference between the reference track and the pose of mobile robot that is controlled through a WFNN controller. Because the characteristic of network structure is very susceptible to several simulation environments such as initial value of network weight, sampling time, learning rate, etc., in this simulation, we use same parameters as shown in Table 1, where initial values of network weight are determined randomly. And the membership function used in this simulation is as shown in Fig. 5. This simulation considers the tracking of a trajectory generated by the following displacements:

Linear velocity  $\delta d = 20\text{cm/sec}$ , Angular velocity  $\delta\theta = 0^\circ/\text{sec}$  ( $0 < t \leq 5$ )  
 Linear velocity  $\delta d = 30\text{cm/sec}$ , Angular velocity  $\delta\theta = 59.3^\circ/\text{sec}$  ( $5 < t \leq 10$ )  
 Linear velocity  $\delta d = 30\text{cm/sec}$ , Angular velocity  $\delta\theta = -59.3^\circ/\text{sec}$  ( $10 < t \leq 15$ )  
 Linear velocity  $\delta d = 20\text{cm/sec}$ , Angular velocity  $\delta\theta = 0^\circ/\text{sec}$  ( $15 < t \leq 20$ )

Table 1. The parameters

Membership Function	11
Learning Rate	0.01
Gain	0.2
Initial Weights	(5,5,0)

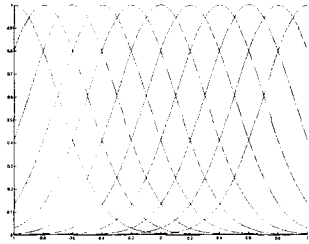


Fig. 5 Membership functions

Fig. 6 shows the reference path and the controlled path using WFNN controller for a mobile robot. Also, Fig. 7 shows the control errors for path tracking of a mobile robot. Figs. 8 and 9 show the control inputs from WFNN controller and the feed forward Jacobian of a mobile robot system, respectively.

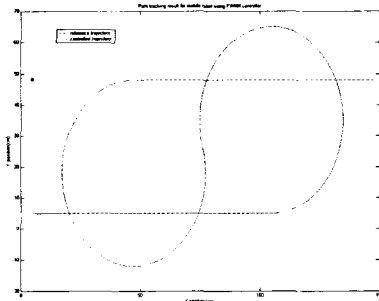


Fig. 6 Controlled path using WFNN controller

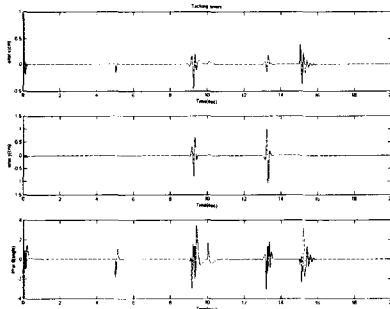


Fig. 7 Path tracking errors

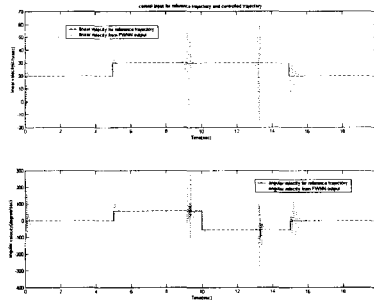


Fig. 8 Reference input and control input

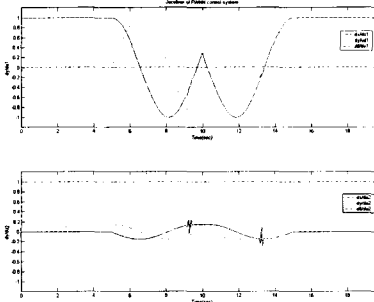


Fig. 9 Feedforward Jacobian of each control input

From these figures and Table 2, we confirm that direct adaptive control system using our WFNN controller works well although the tracking error was occurred in case that a direction is changed. Also, from various simulations, we identified that the learning rate has an effect on the control performance compared with other network parameters. So, the control performance is repeatedly confirmed using various learning rates. Also, in order to evaluate the performance of the proposed WFNN structure, we compare the control results of a WFNN controller with those of FNN and WFM controllers.

Table 2. The simulation result

Maximum control error	$x$	0.0381cm
	$y$	0.0432cm
	$\theta$	0.25°

Figs. 10, 11 and 12 show the control performance according to learning rate using WFNN, FNN and WFM controller, respectively. From Fig. 11, the change of MSEs with respect to learning rate is relatively small although the magnitude of MSEs is large as a whole. On the contrary, from Fig. 12, the change of MSEs with respect to learning rate is large. But, the magnitude of MSEs is very small if the learning rate is chosen properly. From the result of these two figures, FNN controller has the advantage that learning rate is not sensitive to control performance, while, the mobile robot can navigate more accurately using WFM controller.

### 5. Conclusions

In this paper, we have proposed a FNN structure based on wavelet that merges the advantages of neural network, fuzzy model and wavelet transform. In addition, a WFNN controller based on direct adaptive control scheme has also been presented for the solution of the tracking problem for mobile robots. Through computer simulations, we have confirmed that direct adaptive control system using our WFNN controller works well although the tracking error was occurred in case that a direction is changed.

Also, in order to evaluate the performance of the proposed WFNN structure, we have compared the control results of WFNN controller with those of FNN and WFM controllers. As a result, it is shown that the tracking performance of a WFNN controller is better than those of FNN and WFM.

### 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, Singapore, 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 networks", *Proc. of the 2nd 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.

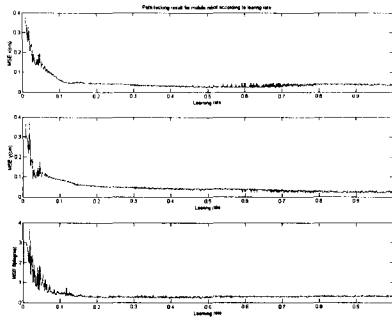


Fig. 10 Control results for a WFNN controller

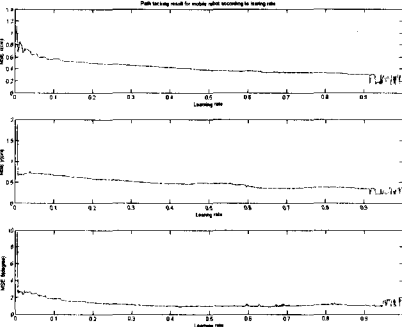


Fig. 11 Control results for a FNN controller

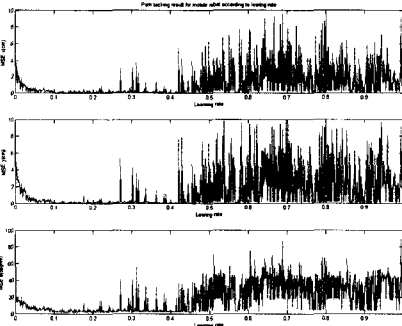


Fig. 12 Control results for a WFM controller

From the result of Fig. 10, we confirmed that if the WFNN controller based structure is applied to direct adaptive control system, it has these two advantages that the control performance is not sensitive to learning rate with small MSEs. Table 3 shows the average MSE with respect to learning rate for each controller structure.

Table 3. Average MSE with respect to learning rate for each controller structure

		MSE		
		$x$	$y$	$\theta$
Controller	WFNN	0.0462 [cm]	0.0472 [cm]	0.3667 [°]
	FNN	0.4173 [cm]	0.4768 [cm]	1.272 [°]
	WFM	1.1682 [cm]	1.5236 [cm]	12.0222 [°]

- [11] 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.
  - [12] C. K. Lin and S. D. Wang, "Fuzzy modeling using wavelet transforms", *Electronics Letters*, Vol. 32, Issue 24, pp. 2255-2256, 1996.
  - [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. 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] Q. Zhang and A. Benveniste, "Wavelet networks", *IEEE Trans. on Neural Network*, Vol. 3, No. 6, pp. 889-898, 1992.
  - [17] Q. Zhang, "Using wavelet network in nonparametric estimation", *IEEE Trans. on Neural Network*, Vol. 8, No. 2, pp. 227-236, 1997.
- 



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

Phone : +82-31-249-9801  
Fax : +82-31-249-9796  
E-mail : yhchoi@kuic.kyonggi.ac.kr



**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 mobile

robot navigation, map construction, fuzzy logic control, neural network, and Kalman filter. His current research interests are in the design and synthesis of wavelet neural network for tracking control of mobile robot.

Phone : +82-2-2123-2773  
Fax : +82-2-362-4539  
E-mail : jsOh@control.yonsei.ac.kr