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

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

* Department of Electrical & Electronic Engineering, Yonsei University, Seoul, Korea
(Tel : +82-2-2123-2773, E-mail: jsoh@control.yonsei.ac.kr, jbpark@yonsei.ac.kr)
**School of Electronic Engineering, Kyonggi University, Suwon, Korea
(Tel : +82-31-249-9801, E-mail: yhchoi@kyonggi.ac.kr)

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 advantages of neural network, fuzzy model and wavelet transform. The basic idea of our WFNN structure 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. In our control system, the control signals are directly obtained to minimize the difference between the reference track and the pose of mobile robot using the gradient descent(GD) method. In addition, an approach that uses adaptive learning rates for the training of WFNN controller is driven via a Lyapunov stability analysis to guarantee the 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 performance of the WFNN controller with those of the FNN, the WNN and the WFM controllers.

Keywords: Fuzzy Neural Network, Wavelet Transform, Fuzzy System, Lyapunov Stability, Mobile Robot Tracking Control.

1. INTRODUCTION

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]-[3]. Intelligent control techniques, based on neural networks and fuzzy logic, have also been developed for path tracking control of mobile robots[4][5]. 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 even instability as well. 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 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[6][7]. This provides a strong motivation for using FNNs in the modeling and control of nonlinear systems. And 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. The conventional fuzzy model can not give the satisfactory result for the transient signal. On the contrary, in the case of 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[8]. Therefore, we design a FNN structure based on wavelet, which merges these advantages of neural network, fuzzy model and wavelet. The basic idea of wavelet based fuzzy neural network(WFNN) is to realize the process of fuzzy reasoning of WFM 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 an approach that uses adaptive learning rates is driven via a Lyapunov stability analysis to guarantee the fast convergence. In this paper, we design the direct adaptive control system using the WFNN structure. Through computer simulations, we demonstrate the effectiveness and feasibility of the proposed control method and compare the control performance of the WFNN controller with those of the FNN, the WFM and the wavelet neural network(WNN) controllers.

2. WAVELET BASED FUZZY NEURAL NETWORK

In our network structure[10], the network output, $\hat{y}_c$, is as follows:

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

where, x_n is the network input, and the network weight set is $\boldsymbol{\gamma} = \{\mathbf{a}, \boldsymbol{\omega}, \mathbf{d}, \mathbf{m}\}$, which is tuned to minimize the model errors via the gradient descent(GD) method. In order to apply the GD method, the squared error function is defined as follows:

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

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:

$$\begin{aligned} \boldsymbol{\gamma}_p(k+1) &= \boldsymbol{\gamma}_p(k) + \Delta\boldsymbol{\gamma}_p(k) = \boldsymbol{\gamma}_p(k) - \eta \frac{\partial J}{\partial \boldsymbol{\gamma}_p(k)} \\ &= \boldsymbol{\gamma}_p(k) - \eta \frac{\partial J}{\partial \hat{\mathbf{Y}}} \frac{\partial \hat{\mathbf{Y}}}{\partial \boldsymbol{\gamma}_p(k)} = \boldsymbol{\gamma}_p(k) + \eta \cdot \mathbf{E} \cdot \hat{\boldsymbol{\upsilon}}_p \,, \end{aligned} \tag{3}$$

where, $\mathbf{E} = [(y_{r1} - \hat{y}_1)\ (y_{r2} - \hat{y}_2) \cdots (y_{rC} - \hat{y}_C)]$ and 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 Eq. (4), and each gradient of WFNN output $\hat{y}$ with respect to each weight is presented as in Eq. (4) to Eq. (7):

$$\hat{\boldsymbol{\upsilon}}_p = \frac{\partial \hat{\mathbf{Y}}}{\partial \boldsymbol{\gamma}_p(k)} = \left[\frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{a}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \boldsymbol{\omega}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{m}(k)}\ \frac{\partial \hat{\mathbf{Y}}}{\partial \mathbf{d}(k)} \right], \tag{4}$$

$$\hat{\upsilon}_{a_{nc}} = \frac{\partial \hat{y}_c}{\partial a_{nc}(k)} = x_n \,, \tag{5}$$

$$\hat{\upsilon}_{\omega_{jc}} = \frac{\partial \hat{y}_c}{\partial \omega_{jc}(k)} = \frac{\partial \sum_{j=1}^{R} y_{jc}}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}}, \tag{6}$$

$$\hat{\upsilon}_{m_{k_n n}, d_{k_n n}} = \frac{\partial \hat{y}_c}{\partial m_{k_n n}, d_{k_n n}(k)} = \frac{\partial \left(\sum_{j=1}^{H} B_{jc}\Phi_j \right)}{\partial m_{k_n n}, d_{k_n n}(k)}$$
$$= \sum_{h=1}^{H} \left(\omega_{jc} \left(\frac{N\dot{U}M(m_{k_n n}, d_{k_n n})}{DEN} - \frac{DE\dot{N}(m_{k_n n}, d_{k_n n})NUM}{DEN^2} \right) \right)_h, \tag{7}$$

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

$$N\dot{U}M(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}}$$
$$= -\frac{1}{d_{k_n n}} \left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})} \left(\left(O^2_{A_{k_n n}} - 1\right) \exp\left(-\frac{1}{2} O^2_{A_{k_n n}} \right) \right) \right),$$

$$DE\dot{N}(m_{k_n n}) = \frac{\partial z_{k_n n}}{\partial m_{k_n n}} \frac{\partial DEN}{\partial z_{k_n n}}$$
$$= -\frac{1}{d_{k_n n}} \sum_{h=1}^{H} \left(\frac{\prod_{n=1}^{N} O_{C_{k_n n}}}{O_{C_{k_n n}}} \left(-O_{A_{k_n n}} \exp\left(-\frac{1}{2} O^2_{A_{k_n n}} \right) \right) \right)_h,$$

$$N\dot{U}M(d_{k_n n}) = \frac{\partial z_{k_n n}}{\partial d_{k_n n}} \frac{\partial NUM}{\partial z_{k_n n}}$$
$$= -\frac{O^2_{A_{k_n n}}}{d_{k_n n}} \left(\frac{\prod_{n=1}^{N} \phi_{k_n n}(z_{k_n n})}{\phi_{k_n n}(z_{k_n n})} \left(\left(O^2_{A_{k_n n}} - 1\right) \exp\left(-\frac{1}{2} O^2_{A_{k_n n}} \right) \right) \right),$$

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

3. PATH TRACKING CONTROL FOR MOBILE ROBOT USING THE WFNN

3.1 Dynamic model of mobile robot

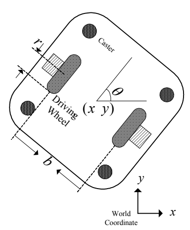

Fig. 1 Mobile robot model and world coordinate

The mobile robot used in this paper is composed of two driving wheels and four casters. And 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. 1. The equation for motion dynamics is as follows:

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \delta d_k cos(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta d_k sin(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta\theta_k \end{bmatrix}, \tag{8}$$

where, δd and $\delta\theta$ 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 WFNN are trained via the GD method. The overall control system is shown in Fig. 2. WFNN controller calculates the control input $\mathbf{U} = [u_d\ u_\theta]^T$ by training the inverse dynamics of plant iteratively. But, the updating of parameters of 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 plant.

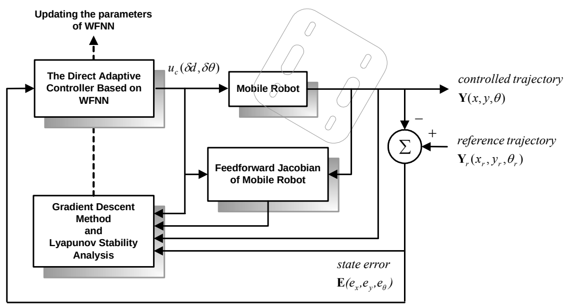

Fig. 2 Direct adaptive control system

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:

$$u_d = \sum_{n=1}^{3} a_{nd} e_n + \sum_{j=1}^{R} y_{jd} = \sum_{n=1}^{3} a_{nd} e_n + \sum_{j=1}^{R} B_{jd} \Phi_j,$$
$$u_\theta = \sum_{n=1}^{3} a_{n\theta} e_n + \sum_{j=1}^{R} y_{j\theta} = \sum_{n=1}^{3} a_{n\theta} e_n + \sum_{j=1}^{R} B_{j\theta} \Phi_j, \tag{9}$$

where,

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

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

Training Procedure :

The purpose of training the parameters of 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:

$$C = \frac{1}{2}((x_r - x)^2 + (y_r - y)^2 + (\theta_r - \theta)^2). \qquad (10)$$

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

$$\begin{aligned} \frac{\partial C}{\partial \boldsymbol{\gamma}_p} &= -e_x \frac{\partial x}{\partial \boldsymbol{\gamma}_p} - e_y \frac{\partial y}{\partial \boldsymbol{\gamma}_p} - e_\theta \frac{\partial \theta}{\partial \boldsymbol{\gamma}_p} \\ &= -e_x \frac{\partial x}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\gamma}_p} - e_y \frac{\partial y}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\gamma}_p} - e_\theta \frac{\partial \theta}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\gamma}_p} \\ &= -\mathbf{E}\, J(u) \frac{\partial \mathbf{U}}{\partial \boldsymbol{\gamma}_p}, \end{aligned} \qquad (11)$$

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:

$$J(u) = \left. \begin{bmatrix} cos(\theta_k + \frac{\partial \theta_k}{2}) & -\frac{\delta d_k}{2} sin(\theta_k + \frac{\delta \theta_k}{2}) \\ sin(\theta_k + \frac{\delta \theta_k}{2}) & \frac{\delta d_k}{2} cos(\theta_k + \frac{\delta \theta_k}{2}) \\ 0 & 1 \end{bmatrix} \right|_{\theta_k = \theta_{k-1}}. \qquad (12)$$

The partial derivative of the control input $\mathbf{U}$ with respect to the parameters of a WFNN controller can be calculated by using Eqs. (13) and (14).

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

$$\begin{aligned} \boldsymbol{\gamma}_p(k+1) &= \boldsymbol{\gamma}_p(k) + \Delta \boldsymbol{\gamma}_p(k) \\ &= \boldsymbol{\gamma}_p(k) - \eta \frac{\partial C}{\partial \boldsymbol{\gamma}_p} = \boldsymbol{\gamma}_p(k) - \eta \mathbf{E}\, J(u) \frac{\partial \mathbf{U}}{\partial \boldsymbol{\gamma}_p}, \end{aligned} \qquad (12)$$

where, η is the learning rate of a WFNN.

From Eqs (12) and (13), each gradient of the controller output u_c with respect to each weight is presented as follows:

$$\begin{aligned} &\frac{\partial u_c}{\partial a_{nc}} = e_n, \quad \frac{\partial u_c}{\partial \omega_{jc}} = \frac{\partial \sum_{j=1}^{R} y_{jc}}{\partial \omega_{jc}(k)} = \frac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}} \\ &\frac{\partial u_c}{\partial m_{k_n n}, d_{k_n n}(k)} = \frac{\partial \left(\sum_{j=1}^{H} B_{jc} \Phi_j \right)}{\partial m_{k_n n}, d_{k_n n}(k)} \\ &= \sum_{h=1}^{H} \left(\omega_{jc} \left(\frac{N\dot{U}M(m_{k_n n}, d_{k_n n})}{DEN} - \frac{D\dot{E}N(m_{k_n n}, d_{k_n n}) NUM}{DEN^2} \right) \right)_h \end{aligned} \qquad , (13)$$

and the detailed description is shown in Eq. (7).

4. STABILITY OF THE WFNN CONTROLLER

In the update rule of Eq. (3), selection of the values for the learning rate η has the significant effect on the control performance. Generally, if η is too big, the system is unstable. And 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 the 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 $\boldsymbol{\gamma}_p$ of the WFNN. $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ are defined as $\mathbf{G}_{p,c}(k) = \frac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)}$ 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, subscript p and c denote each weight and output, respectively. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as follows:

$$0 < \eta_{p,c} < \frac{2}{\mathbf{G}^2_{p,c,\max}(k) \left(J^2_{x,c} + J^2_{y,c} + J^2_{\theta,c} \right)}. \qquad (14)$$

Proof:

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

$$V(k) = \frac{1}{2} \mathbf{E}^T \mathbf{E}(k), \qquad (15)$$

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

$$\begin{aligned} &\Delta V(k) = V(k+1) - V(k) \\ &= \frac{1}{2} (e_x^2(k+1) - e_x^2(k) + e_y^2(k+1) - e_y^2(k) + e_\theta^2(k+1) - e_\theta^2(k)) \end{aligned} ,(16)$$

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

$\Delta e_y(k) \approx \left[\frac{\partial e_y(k)}{\partial \boldsymbol{\gamma}_p(k)} \right]^T \Delta \boldsymbol{\gamma}_p(k)$, $\Delta e_\theta(k) \approx \left[\frac{\partial e_\theta(k)}{\partial \boldsymbol{\gamma}_p(k)} \right]^T \Delta \boldsymbol{\gamma}_p(k)$.

From Eqs. (27), (28) and (29), $\Delta \boldsymbol{\gamma}_p(k)$ is defined as

$$\begin{aligned} &\Delta \boldsymbol{\gamma}_p(k) = -\eta_{p,c} \frac{\partial C}{\partial \boldsymbol{\gamma}_p(k)} \\ &= \eta_{p,c} \left(e_x(k) \frac{\partial x(k)}{\partial u_c(k)} + e_y(k) \frac{\partial y(k)}{\partial u_c(k)} + e_\theta(k) \frac{\partial \theta(k)}{\partial u_c(k)} \right) \left[\frac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)} \right] \end{aligned} , (17)$$

and the error difference can be represented by

$$\begin{aligned} &\Delta e_x(k) \approx \left[\frac{\partial e_x(k)}{\partial \boldsymbol{\gamma}_p(k)} \right]^T \Delta \boldsymbol{\gamma}_p(k) \\ &= -\left[\frac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)} \right]^T \frac{\partial x(k)}{\partial u_c(k)} \eta_{p,c} \left(e_x(k) \frac{\partial x(k)}{\partial u_c(k)} + e_y(k) \frac{\partial y(k)}{\partial u_c(k)} + e_\theta(k) \frac{\partial \theta(k)}{\partial u_c(k)} \right) \left[\frac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)} \right] \\ &= -\eta_{p,c} \left\| \frac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)} \right\|^2 \frac{\partial x(k)}{\partial u_c(k)} \left(e_x(k) \frac{\partial x(k)}{\partial u_c(k)} + e_y(k) \frac{\partial y(k)}{\partial u_c(k)} + e_\theta(k) \frac{\partial \theta(k)}{\partial u_c(k)} \right), \end{aligned} \qquad (18)$$

where, $\Delta e_y(k)$ and $\Delta e_\theta(k)$ have the same description. Let $J_{s,c}$ be a element of the feedforward Jacobian for the state of a mobile robot with respect to the control input, where, subscript s and c denote one state among three state of a mobile robot and the control input u_c, respectively. From Eqs. (16) - (18), $\Delta V(k)$ can be represented as

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

Let us define $\mathbf{G}_{p,c}(k)$ and $\mathbf{G}_{p,c,\max}(k)$ as $\mathbf{G}_{p,c}(k) = \dfrac{\partial u_c(k)}{\partial \boldsymbol{\gamma}_p(k)}$ and $\mathbf{G}_{p,c,\max}(k) \equiv \max_k \|\mathbf{G}_{p,c}(k)\|$, respectively.

Since

$$
\begin{aligned}
\rho(k) &= \eta_{p,c}\|\mathbf{G}_{p,c}(k)\|^2\left[1-\frac{1}{2}\eta_{p,c}\|\mathbf{G}_{p,c}(k)\|^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right] \\
&= \eta_{p,c}\|\mathbf{G}_{p,c}(k)\|^2\left[1-\frac{1}{2}\frac{\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\|\mathbf{G}_{p,c}(k)\|^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}{\mathbf{G}_{p,c,\max}^2(k)}\right], \quad (19) \\
&\geq \eta_{p,c}\|\mathbf{G}_{p,c}(k)\|^2\left[1-\frac{1}{2}\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right] > 0
\end{aligned}
$$

we obtain

$$
0 < \eta_{p,c} < \frac{2}{\mathbf{G}_{p,c,\max}^2(k)\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)} \quad \blacksquare \text{ Q.E.D.} \qquad (20)
$$

Remark 1: The convergence is guaranteed as long as Eq. (19) is satisfied, i.e.:

$$
\eta_{p,c}\left[1-\frac{1}{2}\eta_{p,c}\mathbf{G}_{p,c,\max}^2(k)\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right] > 0\,. \qquad (21)
$$

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

$$
\eta_{p,c,\max} = \frac{1}{\mathbf{G}_{p,c,\max}^2(k)\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}, \qquad (22)
$$

which is the 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, $\left\{\dfrac{\partial u_c(k)}{\partial \mathbf{a}(k)}, \dfrac{\partial u_c(k)}{\partial \boldsymbol{\omega}(k)}, \dfrac{\partial u_c(k)}{\partial \mathbf{m}(k)}, \dfrac{\partial u_c(k)}{\partial \mathbf{d}(k)}\right\}$, of WFNN output u_c with respect to the weight set. Then the convergence is guaranteed if $\eta_{p,c}$ is chosen as

$$
\begin{aligned}
&(a)\quad 0 < \eta_{a,c} < \frac{2}{N|e_n|_{\max}^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}, \\
&(b)\quad 0 < \eta_{\omega,c} < \frac{2}{R^2|O_{B_j}|_{\max}^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}, \\
&(c)\quad 0 < \eta_{m,c} < \frac{2}{\sqrt{C}\left|H|\omega_{jc}|_{\max}^2\left(\dfrac{|DEN|+\sqrt{H}}{|d_{k_n n}|_{\min}|DEN|^2}\right)^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right|_{\max}}, \\
&(d)\quad 0 < \eta_{d,c} < \frac{2}{\sqrt{C}\left|H|\omega_{jc}|_{\max}^2\left(\dfrac{|O_{A,k_n n}^2|_{\max}\big(|DEN|+\sqrt{H}\big)}{|d_{k_n n}|_{\min}|DEN|^2}\right)^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right|_{\max}}.
\end{aligned}
\qquad (23)
$$

Proof : (a)

Let us define $\mathbf{G}_{a,c,\max}(k)$ as $\mathbf{G}_{a,c,\max}(k) \equiv \max_k\|\mathbf{G}_{a,c}(k)\|$. Then from Eq. (20), we obtain $0 < \eta_{a,c} < \dfrac{2}{\mathbf{G}_{a,c,\max}^2(k)\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}$. And from the definition of Theorem 1, the maximum condition can be obtained as

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

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

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

■ Q.E.D.

Remark 2: The maximum learning rates of WFNN, which guarantee the fast convergence, are as shown in Eq. (24).

$$
\begin{aligned}
\eta_{a,c,\max} &= \frac{1}{N|e_n|_{\max}^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}, \\
\eta_{\omega,c,\max} &= \frac{1}{R^2|O_{B_j}|_{\max}^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)}, \\
\eta_{m,c,\max} &= \frac{1}{\sqrt{C}\left|H|\omega_{jc}|_{\max}^2\left(\dfrac{|DEN|+\sqrt{H}}{|d_{k_n n}|_{\min}|DEN|^2}\right)^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right|_{\max}}, \\
\eta_{d,c,\max} &= \frac{1}{\sqrt{C}\left|H|\omega_{jc}|_{\max}^2\left(\dfrac{|O_{A,k_n n}^2|_{\max}\big(|DEN|+\sqrt{H}\big)}{|d_{k_n n}|_{\min}|DEN|^2}\right)^2\big(J_{x,c}^2+J_{y,c}^2+J_{\theta,c}^2\big)\right|_{\max}}.
\end{aligned}
\qquad (24)
$$

5. SIMULATIONS

In this section, we present simulation results to validate the control performance of the proposed WFNN controller for the path tracking of mobile robots. Generally, the characteristic of 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 control procedure is 0.01sec. In the update rule of GD method, selection of the values for the learning rate η has the 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 controller are three state errors, $\mathbf{E}(e_x, e_y, e_\theta)$. The simulation environments and results are as shown in Table 1. This simulation considers the tracking of a trajectory generated by the following displacements:

Table 1. The simulation environments and results

	MF number of each input	Wavelet (Rule Num.)	Parameter	Learning rate	MSE		
					state x [cm]	state y [cm]	state θ [$\circ$]
Our WFNN	3	27	78	**Adaptively** (initial value : 0.1)	**0.0002695**	**0.0003747**	**0.000053**
Our WFNN	3	27	78	Experimentally fixed : 0.08	0.003814	0.004329	0.002589
WFM[8]	16	16	80	Experimentally fixed : 0.011	0.05734	0.07925	0.3254
FNN[7]	4	128	152	Experimentally fixed : 0.044	0.4186	0.9527	1.08903
WNN[9]	*	11	94	Experimentally fixed : 0.214	0.009312	0.007823	0.05426

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

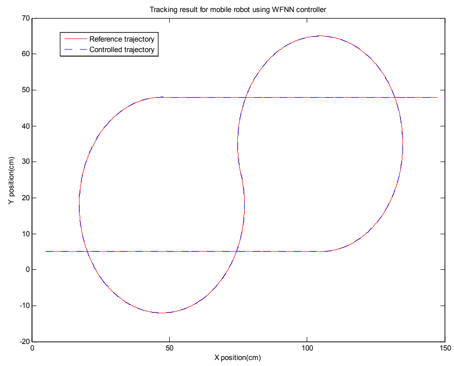

Fig. 3 Controlled path using a WFNN controller

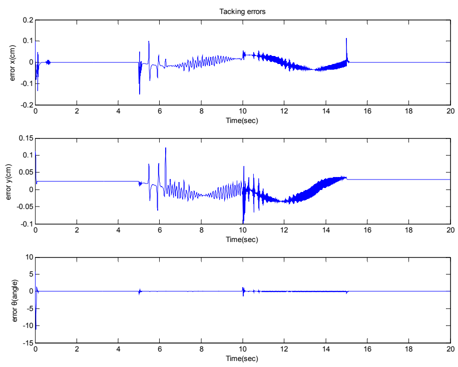

Fig. 4 Path tracking errors

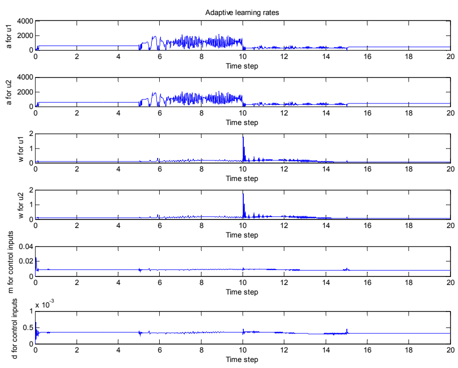

Fig. 5 Adaptive learning rates for the WFNN weights

Figure 3 shows the reference path and controlled path of a mobile robot using a WFNN controller. And Figs. 4 and 5 show the control errors for path tracking of a mobile robot and the adaptive learning rates for the fast convergence and stability, respectively. As a result, if the control errors are changed then the learning rates are changed too for the fast convergence and accuracy. In our simulations, 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 shown in Table 1. From these figures and Table 1, we confirm that the WFNN controller works better than other controllers that use the FNN, the WFM and the 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 function, the number of rule and the learning rate, is experimentally determined via many simulations.

6. CONCLUSION

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 neural network, fuzzy model and wavelet transform as a controller. The control signals were 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 WFNN controller was driven via a Lyapunov stability analysis to guarantee the 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 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 the tracking errors are occurred in case that the direction is changed.

APPENDIX

Proof (b) of Eq. (23):

Let us define $\mathbf{G}_{\omega,c,\max}(k)$ as $\mathbf{G}_{\omega,c,\max}(k) \equiv \max_k \left\| \mathbf{G}_{\omega,c}(k) \right\|$. And then from Eq. (13) and the definition of Theorem 1, the gradient of WFNN output u_c with respect to weight ω_{jc} can be written as $\mathbf{G}_{\omega,c}(k) = \dfrac{\partial u_c}{\partial \omega_{jc}(k)} = \dfrac{\Phi_j}{\sum_{j=1}^{R} I_{D_j}}$, 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:

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

Hence, from Theorem 1 and Eq. (A1), (b) of Theorem 2 follows ■ Q.E.D.

Proof (c) and (d) of Eq. (23):

Let us define $\mathbf{G}_{m,d,c-out}(k)$ as $\mathbf{G}_{m,d,c-out}(k) \equiv \max_k \left\|\mathbf{G}_{m,d,c}(k)\right\|$. And then from Eq. (13) 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

$$\mathbf{G}_{m,d,c}(k) = \sum_{j=1}^{H}\frac{\partial B_{jc}\Phi_j}{\partial m_{k_n,n}, d_{k_n,n}(k)}$$

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

Since

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

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

we obtain

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

$$\le \sqrt{H}\left|\omega_{jc}\right|_{\max}\left(\frac{|DEN| + \sqrt{H}}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right),$$

and

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

$$\le \sqrt{H}\left|\omega_{jc}\right|_{\max}\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN| + \sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right).$$

Therefore, we obtain each maximum condition as follows:

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

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

While the weight $\mathbf{a}$ and $\boldsymbol{\omega}$ have an effect on only one connected output, the weight $\mathbf{m}$ and $\mathbf{d}$ have an effect on all output. Therefore, for the convergence according to the effect of the weight $\mathbf{m}$ and $\mathbf{d}$, the 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

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

$$\mathbf{G}^2_{d,c,\max}(k) = \sqrt{C}\left|H\left|\omega_{jc}\right|^2_{\max}\left(\frac{\left|O^2_{A,k_n n}\right|_{\max}\left(|DEN| + \sqrt{H}\right)}{\left|d_{k_n n}\right|_{\min}|DEN|^2}\right)^2\left(J^2_{x,c} + J^2_{y,c} + J^2_{\theta,c}\right)\right|_{\max}. \quad \text{(A5)}$$

Therefore, if the maximum condition Eqs. (A2) and (A3) are substituted by Eqs. (A4) and (A5), respectively, from Theorem 1, Eqs. (A4) and (A5), (c) and (d) of Theorem 2 follow ■ Q.E.D.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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, pp. 17-22, 2000.

[5] 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.

[6] 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.

[7] 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.

[8] C. K. Lin and S. D. Wang, "Fuzzy modeling using wavelet transforms," Electronics Letters, Vol. 32, pp. 2255-2256, 1996.

[9] 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.

[10] J. S. Oh and Y. H. Choi, "Path tracking control using a wavelet based fuzzy neural network for mobile robots," International Journal of Fuzzy Logic and Intelligent Systems, Vol. 4, No. 1, pp. 111-118, 2004.

[11] 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.