# Self-Recurrent Wavelet Neural Network Based Direct Adaptive Control for Stable Path Tracking of Mobile Robots

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

*Department of Electrical and Electronic Engineering, Yonsei University, Seodaemun-Gu, Seoul, 120-749, Korea
(Tel: +82-2-2123-2773, FAX: +82-2-362-4539, E-mail: niceguy1201@control.yonsei.ac.kr, jbpark@yonsei.ac.kr)
**School of Electronic Engineering, Kyonggi University, Suwon, Kyonggi-Do, 443-760, Korea
(Tel : +82-31-249-9801, E-mail: yhchoi@kuic.kyonggi.ac.kr)

**Abstract:** This paper proposes a direct adaptive control method using self-recurrent wavelet neural network (SRWNN) for stable path tracking of mobile robots. The architecture of the SRWNN is a modified model of the wavelet neural network (WNN). Unlike the WNN, since a mother wavelet layer of the SRWNN is composed of self-feedback neurons, the SRWNN has the ability to store the past information of the wavelet. For this ability of the SRWNN, the SRWNN is used as a controller with simpler structure than the WNN in our on-line control process. The gradient-descent method with adaptive learning rates (ALR) is applied to train the parameters of the SRWNN. The ALR are derived from discrete Lyapunov stability theorem, which are used to guarantee the stable path tracking of mobile robots. Finally, through computer simulations, we demonstrate the effectiveness and stability of the proposed controller.

**Keywords:** Direct adaptive control, Self-recurrent wavelet neural network, Gradient-descent method, Adaptive learning rate, Mobile robot

## 1. Introduction

In recent years, mobile robots have been used as the applications of many areas, such as room cleaning, disabled people assistance, and factory automation. These applications require mobile robots to have the ability to track stably the path. Thus, the stable path tracking problems of mobile robots are a fundamentally important issue and have been studied by many researchers.

In the meanwhile, neural network (NN) has been used as a good tool to control an autonomous mobile robot [1] [2] because no mathematical models are needed and it can easily be applied to nonlinear and linear systems. But, NN have some drawbacks, which come from their inherent characteristics, such as slow convergence, settlement of local minima. Accordingly, recently wavelet neural network (WNN), which absorbs the advantages of high resolution of wavelets and learning of NN, has been proposed to guarantee the fast convergence and is used for identifying and controlling the nonlinear system [3]. However, WNN does not require prior knowledge about the plant to be controlled due to its feedforward structure. Accordingly, WNN cannot adapt rapidly under the circumstances to change frequently the operating conditions and dynamics' parameters like to the operation environment of mobile robots. To overcome problems, we use the self-recurrent wavelet neural network (SRWNN) that we proposed in [4]. Since the SRWNN, a modified WNN, has a mother wavelet layer composed of self-feedback neurons, it can capture a past information of the network and adapt rapidly to sudden changes of the control environment. Due to these properties, the structure of the SRWNN can be simpler than that of the WNN.

In this paper, we propose the design method of the SRWNN based direct adaptive controller for the stable path tracking of the mobile robot. Two SRWNNs are used as each controller in our control scheme for generating two control inputs, the translational and rotational displacement of the robot. The SRWNN controllers are trained by the gradient-descent (GD) method using the adaptive learning rates (ALR). The suitable ALR of controllers for mobile robots are derived in the sense of discrete Lyapunov stability analysis, which are used to guarantee the convergence of the SRWNN controllers in the proposed control system.

This paper is organized as follows. In Section 2, we present some basics of mobile robots and SRWNN. Section 3 discusses the SRWNN based direct adaptive control strategy as applied to the tracking problem. The stability of the control system is analyzed and then the ALR are derived in Section 4. Section 5 presents a simulation result. Finally, Section 6 gives some concluding remarks.

## 2. Preliminaries

### 2.1. The model for mobile robot

The model of mobile robot used in this paper has two opposed drive wheels, mounted on the left and right sides of the robot, and a caster. In this model, the location of the robot be represented by three states, the coordinates $(x_c, y_c)$ of the midpoint between the two driving wheels and the orientation angle $\theta$, as shown in Fig. 1.

The motion dynamics of robot in a global coordinate frame can then be expressed as follows [5]:

$$\begin{bmatrix} x_c(n+1) \\ y_c(n+1) \\ \theta(n+1) \end{bmatrix} = \begin{bmatrix} x_c(n) \\ y_c(n) \\ \theta(n) \end{bmatrix} + \begin{bmatrix} \delta d(n) con(\theta(n) + \frac{\delta\theta(n)}{2}) \\ \delta d(n) sin(\theta(n) + \frac{\delta\theta(n)}{2}) \\ \delta\theta(n) \end{bmatrix} \quad (1)$$

where, $\delta d = \frac{d_R + d_L}{2}$ and $\delta\theta = \frac{d_R - d_L}{b}$ are used as control inputs. Here, $d_R$ and $d_L$ denote the distances, traveled by the right and the left wheel respectively. Also, $b$ is the distance between the wheels.

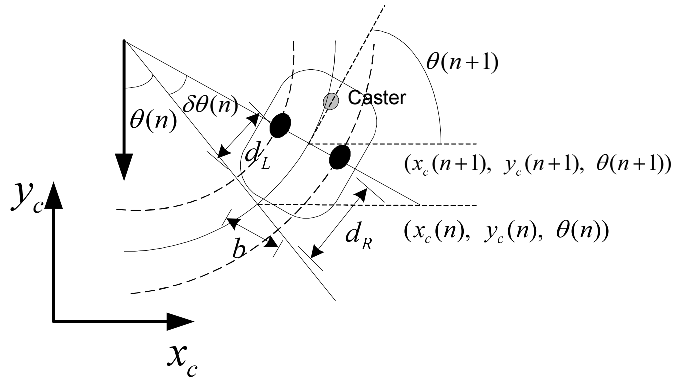


Fig. 1. The representation of the state vector of the mobile robot in a global coordinate frame

### 2.2. Self-recurrent wavelet neural network

A schematic diagram of the SRWNN structure is shown in Fig. 2, which has $N_i$ inputs, one output, and $N_i \times N_w$ mother wavelets. The SRWNN structure consists of four layers.

*The layer 1* is an input layer. This layer accepts the input variables and transmits the accepted inputs to the next layer directly.

*The layer 2* is a mother wavelet layer. Each node of this layer has a mother wavelet and a self-feedback loop. In this paper, we select the first derivative of a gaussian function, $\phi(x) = -x\exp(-\frac{1}{2}x^2)$ as a mother wavelet function. A wavelet $\phi_{jk}$ of each node is derived from its mother wavelet $\phi$ as follows:

$$\phi_{jk}(z_{jk}) = \phi(\frac{u_{jk} - m_{jk}}{d_{jk}}), \quad \text{with} \quad z_{jk} = \frac{u_{jk} - m_{jk}}{d_{jk}} \tag{2}$$

where, $m_{jk}$ and $d_{jk}$ are the translation factor and the dilation factor of the wavelets, respectively. The subscript $jk$ indicates the $k$th input term of the $j$th wavelet. In addition, the inputs of this layer for discrete time $n$ can be denoted by

$$u_{jk}(n) = x_k(n) + \phi_{jk}(n-1) \cdot \alpha_{jk} \tag{3}$$

where, $\alpha_{jk}$ denotes the weight of the self-feedback loop. The input of this layer contains the memory term $\phi_{jk}(n-1)$, which can store the past information of the network. That is, the current dynamics of the system is conserved for the next sample step. Thus, even if the SRWNN has less mother wavelets than the WNN, the SRWNN can attract nicely the system with complex dynamics. Here, $\alpha_{jk}$ is a factor to represent the rate of information storage. These aspects are the apparent dissimilar point between the WNN and the SRWNN. And also, the SRWNN is a generalization system of the WNN because the SRWNN structure is the same as the WNN structure when $\alpha_{jk} = 0$.

*The layer 3* is a product layer. The nodes in this layer are given by the product of the mother wavelets as follows:

$$\begin{aligned}\Phi_j(\mathrm{x}) &= \prod_{k=1}^{N_i} \phi(z_{jk}) \\ &= \prod_{k=1}^{N_i} \left[ -\left(\frac{u_{jk} - m_{jk}}{d_{jk}}\right) exp\left( -\frac{1}{2}\left(\frac{u_{jk} - m_{jk}}{d_{jk}}\right)^2 \right) \right]\end{aligned} \tag{4}$$

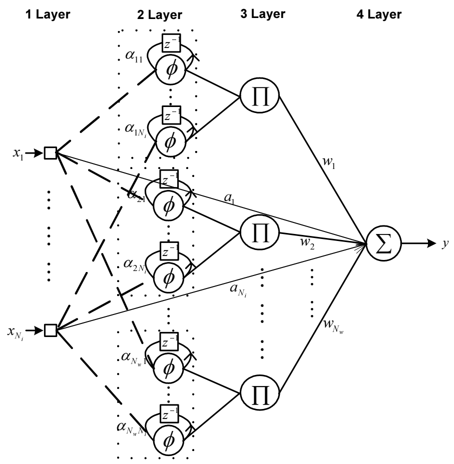


Fig. 2. The proposed SRWNN structure

*The layer 4* is an output layer. The node output is a linear combination of consequences obtained from the output of the layer 3. In addition, the output node accepts directly input values from the input layer. Therefore, the SRWNN output $y(n)$ is composed by self-recurrent wavelets and parameters as follows:

$$y(n) = \sum_{j=1}^{N_w} w_j \Phi_j(\mathrm{x}) + \sum_{k=1}^{N_i} a_k x_k \tag{5}$$

where, $w_j$ is the connection weight between product nodes and output nodes, and $a_j$ is the connection weight between the input nodes and the output node. The weighting vector $W$ of SRWNN is represented by

$$W = [a_k \quad m_{jk} \quad d_{jk} \quad \alpha_k \quad w_j]^T \tag{6}$$

where, the initial values of tuning parameters $a_k$, $m_{jk}$, $d_{jk}$, and $w_j$ are given randomly in the range of [-1 1] but $d_{jk} > 0$. And also, the initial values of $\alpha_{jk}$ are given by 0. That is, there are no feedback units initially.

## 3. Control design for mobile robots

### 3.1. SRWNN Controller

In this section, we design the SRWNN based direct adaptive control system for path tracking of mobile robot. Since the kinematics of the mobile robot consists of 2 inputs and 3 outputs, 2 SRWNN controllers must be used for generating each control input $\delta d$ and $\delta\theta$. The overall controller architecture based on direct adaptive control scheme is shown in Fig. 3. In this architecture, the SRWNNC1 and SRWNNC2 denote two SRWNN controllers for controlling the control input $\delta d$ and $\delta\theta$, respectively. The past control signal $\delta d(n-1)$, the past errors $e_x(n-1)$, and $e_y(n-1)$ are fed into the SRWNNC1 so that the current input $\delta d(n)$ is generated. And also, the inputs of the SRWNNC2, such as the past control signal $\delta\theta(n-1)$, the past errors $e_x(n-1)$, $e_y(n-1)$, and $e_\theta(n-1)$, are used for generating the current control signal $\delta\theta(n)$. Accordingly, two cost functions must be defined to select optimal control signals.

### 3.2. Training algorithm

Let us define cost functions as

$$J_1(n) = \frac{1}{2}e_x^2(n) + \frac{1}{2}e_y^2(n) \tag{7}$$

$$J_2(n) = \frac{1}{2}e_\theta^2(n) \tag{8}$$

where, $e_x(n) = x_r(n) - x_c(n)$, $e_y(n) = y_r(n) - y_c(n)$, and $e_\theta(n) = \theta_r(n) - \theta(n)$. Here, $x_r$, $y_r$, and $\theta_r$ denote the states of the mobile robot for the reference trajectory.

By using the GD method, the weight values of SRWNNC1 and SRWNNC2 are adjusted so that cost functions are minimized after a given number of training cycles. The GD method for each cost functions may be defined as

$$\begin{aligned} W_{1,2}(n+1) &= W_{1,2}(n) + \Delta W_{1,2}(n) \\ &= W_{1,2}(n) + \bar{\eta}_{1,2}\left(-\frac{\partial J_{1,2}(n)}{\partial W_{1,2}(n)}\right) \end{aligned} \tag{9}$$

where, $W_{1,2}$ denote each weighting vectors of the SRWNNC1 and SRWNNC2, respectively. $\bar{\eta}_{1,2} = diag[\eta_{1,2}^a,\ \eta_{1,2}^m,\ \eta_{1,2}^d,\ \eta_{1,2}^\alpha,\ \eta_{1,2}^w]$ are learning rate matrices for weights of the SRWNNC1 and SRWNNC2. The gradient of cost functions $J_1$ and $J_2$ with respect to weighting vectors $W_1$ and $W_2$ of the controllers, respectively, are

$$\begin{aligned} \frac{\partial J_1(n)}{\partial W_1(n)} &= -e_x(n)\frac{\partial x_c(n)}{\partial W_1(n)} - e_y(n)\frac{\partial y_c(n)}{\partial W_1(n)} \\ &= -\left[e_x(n)\frac{\partial x_c(n)}{\partial u_1(n)} + e_y(n)\frac{\partial y_c(n)}{\partial u_1(n)}\right]\frac{\partial u_1(n)}{\partial W_1(n)} \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial J_2(n)}{\partial W_2(n)} &= -e_\theta(n)\frac{\partial \theta(n)}{\partial W_2(n)} \\ &= -e_\theta(n)\frac{\partial \theta(n)}{\partial u_2(n)}\frac{\partial u_2(n)}{\partial W_2(n)} \end{aligned} \tag{11}$$

where, $u_1(n) = \delta d(n)$ and $u_2(n) = \delta\theta(n)$. And $\frac{\partial x_c(n)}{\partial u_1(n)}$, $\frac{\partial y_c(n)}{\partial u_1(n)}$, and $\frac{\partial \theta(n)}{\partial u_2(n)}$ denote the system sensitivity. It can be computed from Eq. (1). And also, the components of Jacobian of the control inputs $u_1$ and $u_2$ with respect to each weighting vector $W_{1,2}$ are computed by Eq. (5) as follows:

$$\frac{\partial u_{1,2}(n)}{\partial a_{1,2k}(n)} = x_{1,2k} \tag{12}$$

$$\frac{\partial u_{1,2}(n)}{\partial m_{1,2jk}(n)} = -\frac{w_{1,2j}}{d_{1,2jk}}\frac{\partial \Phi_{1,2j}(\mathrm{x})}{\partial z_{1,2jk}} \tag{13}$$

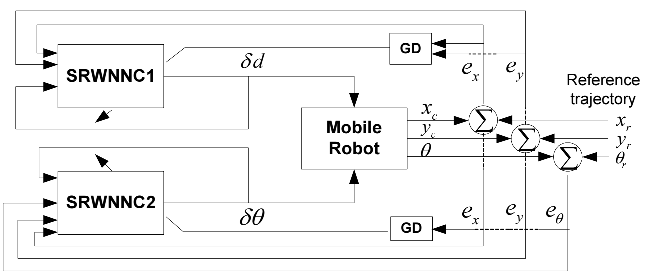


Fig. 3. The proposed control structure for mobile robot

$$\frac{\partial u_{1,2}(n)}{\partial d_{1,2jk}(n)} = -\frac{w_{1,2j}}{d_{1,2jk}}z_{1,2jk}\frac{\partial \Phi_{1,2j}(\mathrm{x})}{\partial z_{1,2jk}} \tag{14}$$

$$\frac{\partial u_{1,2}(n)}{\partial \alpha_{1,2jk}(n)} = \frac{w_{1,2j}}{d_{1,2jk}}\phi_{1,2jk}(n-1)\frac{\partial \Phi_{1,2j}(\mathrm{x})}{\partial z_{1,2jk}} \tag{15}$$

$$\frac{\partial u_{1,2}(n)}{\partial w_{1,2j}(n)} = \Phi_{1,2j}(\mathrm{x}) \tag{16}$$

where, $\frac{\partial \Phi_{1,2j}}{\partial z_{1,2jk}} = \phi(z_{1,2j1})\phi(z_{1,2j2})\cdots\dot{\phi}(z_{1,2jk})\cdots\phi(z_{1,2jN_i})$ and $\dot{\phi}(z_{1,2jk}) = \frac{\partial \phi_{1,2j}}{\partial z_{1,2jk}} = (z_{1,2jk}^2 - 1)\exp(-\frac{1}{2}z_{1,2jk}^2)$.

## 4. Stability analysis

In this section, we analyze the stability of the proposed controller for stable path tracking of mobile robots. Though the cost function of each controller is designed differently, one Lyapunov function for analyzing the stability of the unified control system is defined out of consideration for two control inputs. The convergence of the SRWNNC1 and SRWNNC2, which are trained with GD method, is related to select the appropriate learning rates. To solve this problem, we develop some convergence theorems for selecting appropriate learning rates adaptively.

Let us define a discrete Lyapunov function as

$$V(n) = \frac{1}{2}[e_x^2(n) + e_y^2(n) + e_\theta^2(n)] \tag{17}$$

where, $e_x(n)$, $e_y(n)$, and $e_\theta(n)$ are the control errors. The change in the Lyapunov function is obtained by

$$\begin{aligned} \Delta V(n) &= V(n+1) - V(n) \\ &= \frac{1}{2}[e_x^2(n+1) - e_x^2(n) \\ &\quad + e_y^2(n+1) - e_y^2(n) + e_\theta^2(n+1) - e_\theta^2(n)] \end{aligned} \tag{18}$$

Three error differences can be represented by [6]

$$\begin{aligned} \Delta e_x(n) &= e_x(n+1) - e_x(n) \\ &\approx \left[\frac{\partial e_x(n)}{\partial W_1^i(n)}\right]^T \Delta W_1^i(n) \end{aligned} \tag{19}$$

$$\begin{aligned} \Delta e_y(n) &= e_y(n+1) - e_y(n) \\ &\approx \left[\frac{\partial e_y(n)}{\partial W_1^i(n)}\right]^T \Delta W_1^i(n) \end{aligned} \tag{20}$$

$$\begin{aligned} \Delta e_\theta(n) &= e_\theta(n+1) - e_\theta(n) \\ &\approx \left[\frac{\partial e_\theta(n)}{\partial W_2^i(n)}\right]^T \Delta W_2^i(n) \end{aligned} \tag{21}$$

where, $W_1^i(n)$ and $W_2^i(n)$ are an arbitrary component of the weighting vectors $W_1(n)$ and $W_2(n)$. And the corresponding changes of them are denoted by $\Delta W_1^i(n)$ and $\Delta W_2^i(n)$. Using Eq. (9)∼(11), $\Delta W_{1,2}$ are obtained by

$$\Delta W_1^i(n) = \eta_1^i\left[e_x(n)\frac{\partial x_c(n)}{\partial u_1(n)} + e_y(n)\frac{\partial y_c(n)}{\partial u_1(n)}\right]\frac{\partial u_1(n)}{\partial W_1^i(n)} \tag{22}$$

$$\Delta W_2^i(n) = \eta_2^i e_\theta(n)\frac{\partial\theta(n)}{\partial u_2(n)}\frac{\partial u_2(n)}{\partial W_2^i(n)}. \tag{23}$$

where $\eta_1^i$ and $\eta_2^i$ are an arbitrary diagonal element of the learning rate matrices $\bar{\eta}_1$ and $\bar{\eta}_2$ corresponding to the weight component $W_1^i(n)$ and $W_2^i(n)$.

**Theorem 1:** Let $\bar{\eta}_{1,2} = [\eta_{1,2}^1 \quad \eta_{1,2}^2 \quad \eta_{1,2}^3 \quad \eta_{1,2}^4 \quad \eta_{1,2}^5] = [\eta_{1,2}^a \quad \eta_{1,2}^m \quad \eta_{1,2}^d \quad \eta_{1,2}^\alpha \quad \eta_{1,2}^w]$ be the learning rates for the weights of the SRWNNC1 and SRWNNC2 and define $\mathbf{C_{1,2,max}}$ as

$$\begin{aligned}\mathbf{C_{1,2,max}} &= [C_{1,2,max}^1 \quad C_{1,2,max}^2 \quad C_{1,2,max}^3 \quad C_{1,2,max}^4 \quad C_{1,2,max}^5]^T \\ &= \left[max_n\left|\frac{\partial u_{1,2}(n)}{\partial a_{1,2}(n)}\right| \quad max_n\left|\frac{\partial u_{1,2}(n)}{\partial m_{1,2}(n)}\right|\right. \\ &\left. max_n\left|\frac{\partial u_{1,2}(n)}{\partial d_{1,2}(n)}\right| \quad max_n\left|\frac{\partial u_{1,2}(n)}{\partial \alpha_{1,2}(n)}\right| \quad max_n\left|\frac{\partial u_{1,2}(n)}{\partial w_{1,2}(n)}\right|\right]^T\end{aligned}$$

Then, the asymptotic convergence of the SRWNNC1 and SRWNNC2 is guaranteed if $\eta_{1,2}^i$ are chosen to satisfy

$$0 < \eta_1^i < \frac{2}{(S_x^2+S_y^2)(C_{1,max}^i)^2} \tag{24}$$

$$0 < \eta_2^i < \frac{2}{(S_\theta C_{2,max}^i)^2} \tag{25}$$

where $i = 1, ...5$, $S_x = \frac{\partial x_c(n)}{\partial u_1(n)}$, $S_y = \frac{\partial y_c(n)}{\partial u_1(n)}$, and $S_\theta = \frac{\partial \theta(n)}{\partial u_2(n)}$.

**Proof:** From Eq. (17), $V(n) > 0$. Using Eq. (19) $\sim$ (23), the change in the Lyapunov function is

$$\begin{aligned}\Delta V(n) &= V(n+1) - V(n) \\ &= \frac{1}{2}[e_x^2(n+1) - e_x^2(n) \\ &\quad + e_y^2(n+1) - e_y^2(n) + e_\theta^2(n+1) - e_\theta^2(n)] \\ &= \Delta e_x(n)\left[e_x(n) + \frac{1}{2}\Delta e_x(n)\right] \\ &\quad + \Delta e_y(n)\left[e_y(n) + \frac{1}{2}\Delta e_y(n)\right] \\ &\quad + \Delta e_\theta(n)\left[e_\theta(n) + \frac{1}{2}\Delta e_\theta(n)\right] \\ &= -\left[\frac{\partial u_1(n)}{\partial W_1^i(n)}\right]^T \eta_1^i(e_x(n)S_x + e_y(n)S_y)\frac{\partial u_1(n)}{\partial W_1^i(n)} \\ &\quad \cdot\left[(e_x(n)S_x + e_y(n)S_y) - \frac{1}{2}\left[\frac{\partial u_1(n)}{\partial W_1^i(n)}\right]^T\right. \\ &\quad \left.\cdot\ \eta_1^i(e_x(n)S_x + e_y(n)S_y)\frac{\partial u_1(n)}{\partial W_1^i(n)}(S_x^2+S_y^2)\right] \\ &\quad - \left[\frac{\partial u_2(n)}{\partial W_2^i(n)}\right]^T S_\theta\eta_2^i e_\theta(n) S_\theta\frac{\partial u_2(n)}{\partial W_2^i(n)} \\ &\quad \cdot\left[e_\theta - \frac{1}{2}\left[\frac{\partial u_2(n)}{\partial W_2^i(n)}\right]^T S_\theta\eta_2^i e_\theta(n) S_\theta\frac{\partial u_2(n)}{\partial W_2^i(n)}\right] \\ &= -(e_x(n)S_x + e_y(n)S_y)^2\left[\eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\right. \\ &\quad \left.\cdot\left(1 - \frac{1}{2}(S_x^2+S_y^2)\eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\right)\right] \\ &\quad - e_\theta^2(n)S_\theta^2\left[\eta_2^i\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\left(1 - \frac{1}{2}\eta_2^i S_\theta^2\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\right)\right] \\ &= -(e_x(n)S_x + e_y(n)S_y)^2\rho - e_\theta^2(n)\gamma\end{aligned}$$

where,

$$\rho \geq \eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\left(1 - \frac{1}{2}(S_x^2+S_y^2)\eta_1^i(C_{1,max}^i)^2\right)$$

$$\gamma \geq \eta_2^i\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\left(1 - \frac{1}{2}\eta_2^i S_\theta^2(C_{2,max}^i)^2\right)$$

If $\rho > 0$ and $\gamma > 0$ are satisfied, $\Delta V(n) < 0$. Thus, the asymptotic convergence of the proposed control system are guaranteed. Here, we obtain Eqs. (24) and (25). This completes the proof of the theorem. ■

**Theorem 2:** Let $\eta_1^a$ and $\eta_2^a$ be the learning rates for the input direct weights of the SRWNNC1 and the SRWNNC2. The asymptotic convergences of the SRWNNC1 and the SRWNNC2 are guaranteed if the learning rates $\eta_1^a$ and $\eta_2^a$ satisfies:

$$0 < \eta_1^a < \frac{2}{(S_x^2+S_y^2)N_{1i}|x_{1max}|^2} \tag{26}$$

$$0 < \eta_2^a < \frac{2}{S_\theta^2 N_{2i}|x_{2max}|^2} \tag{27}$$

where, $N_{1i}$ and $N_{2i}$ denote the input number of the SRWNNC1 and the SRWNNC2, respectively. $x_{1max}$ and $x_{2max}$ are the maximum value of each controller's input, respectively.

**Proof:**

$$C_1^1(n) = \frac{\partial u_1(n)}{\partial a_1(n)} = \sum_{k=1}^{N_{1i}} x_{1k} < \sqrt{N_{1i}}|x_{1max}|$$

And also, $C_2^1(n)$ can be determined by the same method as $C_1^1(n)$. Therefore, from Theorem 1 , we obtain (26) and (27). ■

In order to prove Theorem 3, the following lemmas are used.

**Lemma** 1: Let $f(t) = t\exp(-t^2)$. Then $|f(t)| < 1, \forall f \in \mathbb{R}$.

**Lemma** 2: Let $g(t) = t^2\exp(-t^2)$. Then $|g(t)| < 1, \forall g \in \mathbb{R}$.

**Theorem 3:** Let $\eta_{1,2}^m$, $\eta_{1,2}^d$ and $\eta_{1,2}^\alpha$ be the learning rates of the translation, dilation and self-feedback weights for the SRWNNC1 and the SRWNNC2, respectively. The asymptotic convergence is guaranteed if the learning rates satisfy:

$$0 < \eta_1^m, \eta_1^\alpha < \frac{2}{(S_x^2+S_y^2)N_{1w}N_{1i}}\left[\frac{1}{|w_{1,max}|\left(\frac{2\exp(-0.5)}{|d_{1,min}|}\right)}\right]^2 \tag{28}$$

$$0 < \eta_2^m, \eta_2^\alpha < \frac{2}{S_\theta^2 N_{2w}N_{2i}}\left[\frac{1}{|w_{2,max}|\left(\frac{2\exp(-0.5)}{|d_{2,min}|}\right)}\right]^2 \tag{29}$$

$$0 < \eta_1^d < \frac{2}{(S_x^2+S_y^2)N_{1w}N_{1i}}\left[\frac{1}{|w_{1,max}|\left(\frac{2\exp(0.5)}{|d_{1,min}|}\right)}\right]^2 \tag{30}$$

$$0 < \eta_2^d < \frac{2}{S_\theta^2 N_{2w}N_{2i}}\left[\frac{1}{|w_{2,max}|\left(\frac{2\exp(0.5)}{|d_{2,min}|}\right)}\right]^2 \tag{31}$$

where, $N_{1w}$ and $N_{2w}$ are the number of nodes in the product layer of the SRWNNC1 and the SRWNNC2, respectively.

**Proof:**

1) The learning rate $\eta_1^m$ of the translation weight $m_1$:

$$C_1^2(n) = \frac{\partial u_1(n)}{\partial m_1(n)}$$
$$= \sum_{j=1}^{N_{1w}} w_{1,j}\left(\frac{\partial \Phi_{1j}}{\partial m_1}\right)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(\frac{\partial \phi(z_{1,jk})}{\partial z_{1,jk}}\frac{\partial z_{1,jk}}{\partial m_1}\right)\right\} \quad (32)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(2\exp(-0.5)\left(-\frac{1}{d_1}\right)\right)\right\} \quad (33)$$
$$< \sqrt{N_{1w}}\sqrt{N_{1i}}|w_{1,max}|\left|\frac{2\exp(-0.5)}{d_{1,min}}\right|$$

According to Lemma 2,

$$\left|\left(\frac{1}{2}z_{1,jk}^2 - \frac{1}{2}\right)\exp\left\{-\left(\frac{1}{2}z_{1,jk}^2 - \frac{1}{2}\right)\right\}\right| < 1$$

Thus, eq. (32) is clearly smaller than eq. (33). And, $C_2^2(n)$ can be determined by the same method as $C_1^2(n)$. Accordingly, from Theorem 1, we can find Eqs. (28) and (29).

2) The learning rate $\eta_1^d$ of the dilation weight $d_1$ :

$$C_1^3(n) = \frac{\partial u_1(n)}{\partial d_1(n)}$$
$$= \sum_{j=1}^{N_{1w}} w_{1,j}\left(\frac{\partial \Phi_{1,j}}{\partial d_1}\right)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(\frac{\partial \phi(z_{1,jk})}{\partial z_{1,jk}}\frac{\partial z_{1,jk}}{\partial d_1}\right)\right\} \quad (34)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(2\exp(0.5)\left(\frac{1}{d_1}\right)\right)\right\} \quad (35)$$
$$< \sqrt{N_{1w}}\sqrt{N_{1i}}|w_{1,max}|\left|\frac{2\exp(0.5)}{d_{1,min}}\right|$$

According to Lemma 1 and Lemma 2,

$$\left|z_{1,jk}\exp\left(-z_{1,jk}^2\right)\right| < 1$$

$$\left|\left(\frac{1}{2} - \frac{1}{2}z_{1,jk}^2\right)\exp\left\{-\left(\frac{1}{2} - \frac{1}{2}z_{1,jk}^2\right)\right\}\right| < 1$$

Thus, eq. (34) is clearly smaller than eq. (35). And, $C_2^3(n)$ can be determined by the same method as $C_1^3(n)$. Accordingly, from Theorem 1, we can find Eqs. (30) and (31).

3) The learning rate $\eta_1^\alpha$ of the self-feedback weight $\alpha_1$ :

$$C_1^4(n) = \frac{\partial u_1(n)}{\partial \alpha_1(n)}$$
$$= \sum_{j=1}^{N_{1w}} w_{1,j}\left(\frac{\partial \Phi_{1,j}}{\partial \alpha_1}\right)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(\frac{\partial \phi(z_{1,jk})}{\partial z_{1,jk}}\frac{\partial z_{1,jk}}{\partial \alpha_1}\right)\right\} \quad (36)$$
$$< \sum_{j=1}^{N_{1w}} w_{1,j}\left\{\sum_{k=1}^{N_{1i}} max\left(2\exp(-0.5)\left(\frac{\phi_{1,jk}(n-1)}{d_1}\right)\right)\right\} \quad (37)$$
$$< \sqrt{N_{1w}}\sqrt{N_{1i}}|w_{1,max}|\left|\frac{2\exp(-0.5)}{d_{1,min}}\right|$$

According to Lemma 2,

$$\left|\left(\frac{1}{2}z_{1,jk}^2 - \frac{1}{2}\right)\exp\left\{-\left(\frac{1}{2}z_{1,jk}^2 - \frac{1}{2}\right)\right\}\right| < 1$$

Thus, eq. (36) is clearly smaller than eq. (37). And, $C_2^4(n)$ can be determined by the same method as $C_1^4(n)$. Accordingly, from Theorem 1, we can find Eqs. (28) and (29). ■

**Theorem 4:** Let $\eta_1^w$ and $\eta_2^w$ be the learning rates for the weight $w_1$ of the SRWNNC1 and the weight $w_2$ of the SRWNNC2. Then, the asymptotic convergence is guaranteed if the learning rates satisfy:

$$0 < \eta_1^w < \frac{2}{N_{1w}}$$
$$0 < \eta_2^w < \frac{2}{N_{2w}}$$

**Proof:**

$$C_1^5(n) = \frac{\partial u_1(n)}{\partial w_1} = \sum_{j=1}^{N_{1w}} \Phi_{1,j} \quad (38)$$

Then, since we have $\Phi_{1,j} \le 1$ for all $j$, $|C_1^5(n)| \le \sqrt{N_{1w}}$. And also, $C_2^5(n)$ can be determined by the same method as $C_1^5(n)$. Accordingly, from Theorem 1, we find that $0 < \eta_1^w < 2/N_{1w}$ and $0 < \eta_2^w < 2/N_{2w}$. ■

## 5. Simulation result

To visualize the validity of the proposed SRWNN controllers based on indirect adaptive control scheme, we present a simulation result for the stable path tracking of the mobile robot. The design parameters of our control system are chosen as $b = 60$, $N_{1w} = N_{2w} = 1$, $N_{1i} = 3$, and $N_{2i} = 4$. That is, the structures of the SRWNNC1 and SRWNNC2 are designed very simply. The ALR is used for training the SRWNNC1 and SRWNNC2. The sampling time is 0.01 and the departure posture is $(5, 5, \pi/8)$. In this simulation, to examine the tracking performance of both the curved line and straight line, the reference trajectory is generated by the following control inputs.

$$u_1 = 20cm/sec, \quad u_2 = 0rad/sec \quad (0 \le t < 5)$$
$$u_1 = 30cm/sec, \quad u_2 = 1rad/sec \quad (5 \le t < 10)$$
$$u_1 = 30cm/sec, \quad u_2 = -1rad/sec \quad (10 \le t < 15)$$
$$u_1 = 20cm/sec, \quad u_2 = 0rad/sec \quad (15 \le t \le 20)$$

Figure 4 presents the tracking control result for the proposed control system and Fig. 5 shows the control errors. The ALR for training the SRWNNC1 and SRWNNC2 are shown in Figs. 6 and 7. The learning rates of the weight $w_{1,2}$ are chosen as 1 by Theorem 4. Note that the optimal learning rates during the path change are found by the ALR algorithm. In Fig. 4, we can observe that SRWNNC1 and SRWNNC2 using the ALR can adapt stably to the variation of the complex path. The control performance measure is tabulated in Table 1 using the mean-squared error (MSE) as the performance index. From the results of Table 1, we can observe that the proposed controllers using the ALR have a good performance.

## 6. Conclusion

A SRWNN-based adaptive direct control scheme for mobile robots has been proposed. In control scheme, two SRWNN controllers have been designed for generating the control inputs. The structures of two SRWNNs have been trained by the GD method. Since the SRWNN has the ability for storing the past information of the network, it can adapt rapidly to changes of the operation environment of mobile robots. Using the discrete Lyapunov theorem, stability of the whole control scheme has been carried out and the ALR has been also established for the stable path tracking of the mobile robot. A simulation result has shown that the proposed control system has an on-line adapting ability for controlling the mobile robot.

Table 1. The tracking control errors

| $x_c$ MSE | $y_c$ MSE | $\theta$ MSE |
|---|---|---|
| 0.0008 | 0.0009 | 0.00007 |

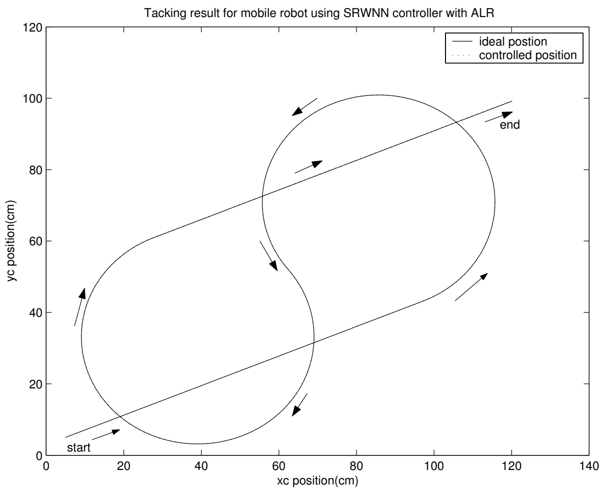


Fig. 4. Tracking result of the mobile robot

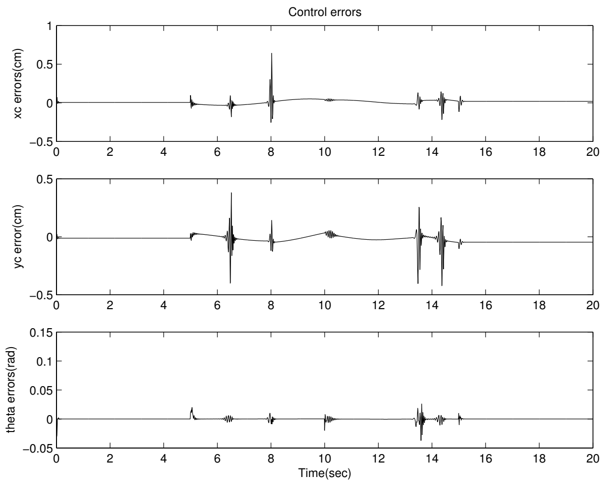


Fig. 5. The tracking control errors

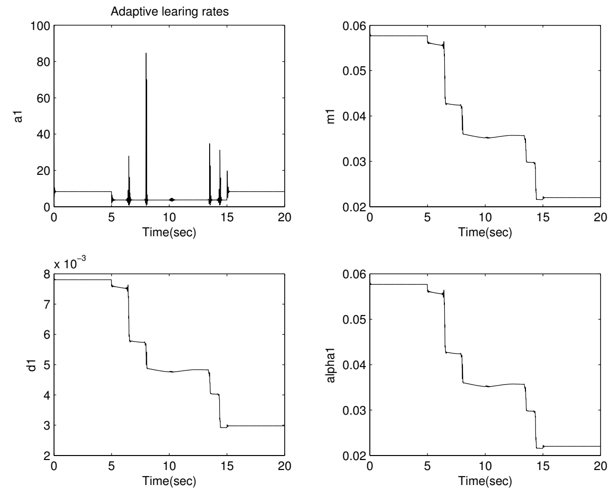


Fig. 6. The ALR of the SRWNNC1

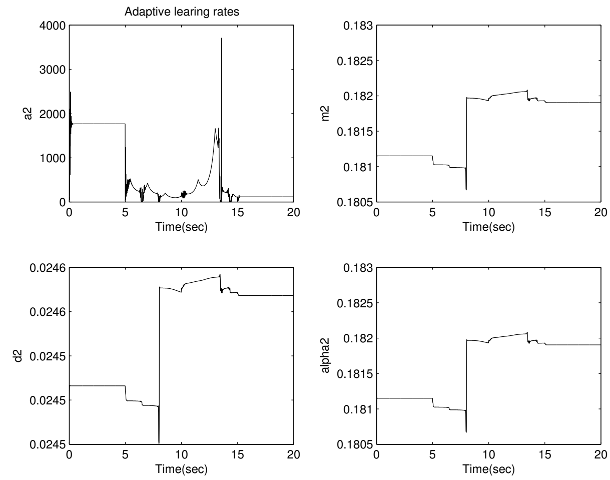


Fig. 7. The ALR of the SRWNNC2

## References


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

[2] K. Watanabe, J. Tang, S. Koga, and T. Fukuda, "A fuzzy-gaussian neural network and its application to mobile robot control", *IEEE Trans. on Control Systems Tech.,* vol. 4, no. 2, pp. 193-199, March, 1996.

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

[4] S. J. You, Y. H. Choi, and J. B. Park, "Generalized predictive control for chaotic systems using a self-recurrent wavelet neural network", *Proc. of KIEE Information and Control Conf.,* pp. 421-424, 2003.

[5] C. M. Wang, "Location estimation and uncertainty analysis for mobile robots", *Proc. of the Int. Conf. on Robotics and Automation,* pp. 1230-1235, 1988.

[6] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamics systems control," *IEEE Trans. on Neural Networks,* vol. 6, no. 1, pp. 144-156, 1995.