2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Direct Adaptive Control Using Self Recurrent Wavelet Neural Network Via Adaptive Learning Rates for Stable Path Tracking of Mobile Robots

Sung Jin Yoo, Jin Bae Park, and Yoon Ho Choi

Abstract— **This paper proposes a direct adaptive control method for stable path tracking of mobile robots using self recurrent wavelet neural network (SRWNN). As the proposed SRWNN is a modified model of the wavelet neural network (WNN), the SRWNN includes the basic ability of the WNN such as fast convergence. Besides the SRWNN has a property, unlike the WNN, that the SRWNN can store the past information of the network because a mother wavelet layer of the SRWNN is composed of self-feedback neurons. Accordingly, the SRWNN can easily cope with the unexpected change of the system. For the control problem, two SRWNNs are used as each direct adaptive controller for generating two control inputs, the translational and rotational displacement of the mobile robot. Specially, the gradient-descent method with adaptive learning rates (ALRs) is applied to train the parameters of the SRWNN controllers. The ALRs are derived from the discrete Lyapunov stability theorem out of consideration for the model of mobile robots, 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.**

I. 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 control of mobile robots is a fundamentally important issue and has been studied by many researchers [1]–[7].

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, NNs have some drawbacks, which come from their inherent characteristics, such as slow convergence, settlement of local minima. To solve these defects, 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 the identification and control of nonlinear systems [8]–[13]. However, the WNN does not require prior knowledge about the plant to be controlled due to its feedforward structure. Therefore, the WNN cannot adapt rapidly under the circumstances, such as the operation environment of mobile robots, to change frequently the operating conditions and dynamics' parameters. To overcome these problems, we propose the self recurrent wavelet neural network (SRWNN), which is a modified the WNN. Since the SRWNN has a mother wavelet layer composed of self-feedback neurons, it can capture the past information of the network and adapt rapidly to sudden changes of the control environment. Due to these properties, the structure of SRWNN can be simpler than that of the WNN.

The training method of the network structure has usually used the back-propagation (BP) or gradient-descent (GD) method [10]–[15]. But since the BP method has a problem that the optimal learning rates cannot easily be found, the adaptive learning rates (ALRs), which can adapt rapidly the change of the plant, have been researched. The ALRs have usually been derived from the Lyapunov stability theorem and applied to the various networks such as the diagonal recurrent neural network [15], the recurrent fuzzy neural network [11], and the wavelet neural network [12]. But these works applied the ALRs only to the single-input single-output system. If the ALRs are applied to the multi-input multi-output system, they must be induced considering the relation between input and output of the plant. Specially, in case that the number of inputs and outputs of the plant are different, it is not easy to derive the ALR theorems. Accordingly, we propose the design method of the SRWNN based direct adaptive controller for the stable path tracking of the mobile robot, which has two inputs and three outputs, and develop the ALR theorems, which are suitable for the SRWNN tracking controller. Two SRWNNs are used as each controller in our control scheme for generating two control inputs that are the translational and rotational displacement of the mobile robot. The SRWNN controllers are trained by the GD method using the ALRs. The ALRs 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.

II. SELF RECURRENT WAVELET NEURAL NETWORK

A schematic diagram of the SRWNN structure is shown in Fig. 1, which has N_i inputs, one output, and $N_i \times N_w$ mother wavelets. N_w denotes the number of product nodes. The SRWNN structure consists of four layers.

Sung Jin Yoo and Jin Bae Park are with Department of Electrical and Electronic Engineering, Yonsei University, Seodaemun-Gu, Seoul, 120-749, Korea. niceguy1201@control.yonsei.ac.kr, jbpark@yonsei.ac.kr

Yoon Ho Choi is with School of Electronic Engineering, Kyonggi University, Suwon, Kyonggi-Do, 443-760, Korea. yhchoi@kyonggi.ac.kr

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 ϕ_{jk} of each node is derived from its mother wavelet ϕ as follows:

$$\phi_{jk}(z_{jk}) = \phi(\frac{u_{jk} - m_{jk}}{d_{jk}}), \quad \text{with} \quad z_{jk} = \frac{u_{jk} - m_{jk}}{d_{jk}}, \quad (1)$$

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

$$u_{jk}(n) = x_k(n) + \phi_{jk}(n-1) \cdot \alpha_{jk}, \quad (2)$$

where, α_{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 well the system with complex dynamics. Here, α_{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:

$$\begin{aligned} \Phi_j(x) &= \prod_{k=1}^{N_i} \phi(z_{jk}) \\ &= \prod_{k=1}^{N_i} \left[-z_{jk} exp\left(-\frac{1}{2} z_{jk}^2 \right) \right]. \end{aligned} \quad (3)$$

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:

$$y(n) = \sum_{j=1}^{N_w} w_j \Phi_j(x) + \sum_{k=1}^{N_i} a_k x_k, \quad (4)$$

where, w_j is the connection weight between product nodes and output nodes, and a_k is the connection weight between the input nodes and the output node. By using the direct term, the SRWNN has a number of advantages such as a direct linear feedthrough network, including initialization of network parameters based on process knowledge and enhanced extrapolation outside of examples the learning data sets [16].

The weighting vector W of SRWNN is represented by

$$W = [a_k \quad m_{jk} \quad d_{jk} \quad \alpha_{jk} \quad w_j]^T, \quad (5)$$

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 α_{jk} are given by 0. That is, there are no feedback units initially.

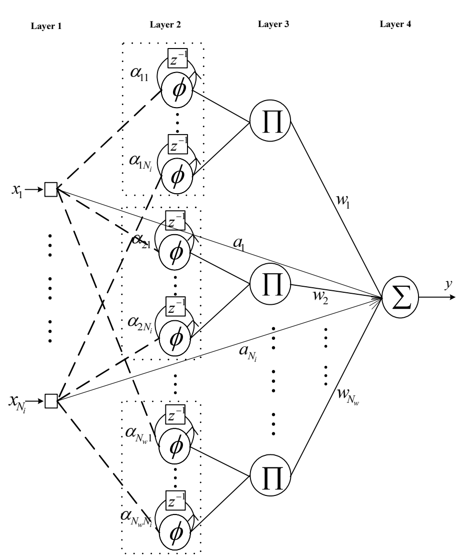


Fig. 1. The SRWNN structure

III. Control Design for Path Tracking of Mobile Robots

A. The mobile robot model

The model of mobile robots 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 mobile robot is represented by three states: the coordinates (x_c, y_c) of the midpoint between the two driving wheels and the orientation angle θ. The motion dynamics of the mobile robot in a global coordinate frame can then be expressed as follows [17]:

$$\begin{bmatrix} x_c(n+1) \\ y_c(n+1) \\ \theta(n+1) \end{bmatrix} = \begin{bmatrix} x_c(n) \\ y_c(n) \\ \theta(n) \end{bmatrix} + \mathbf{M}(n), \quad (6)$$

where,

$$\mathbf{M}(n) = \begin{bmatrix} \delta d(n) con(\theta(n) + \delta\theta(n)/2) \\ \delta d(n) sin(\theta(n) + \delta\theta(n)/2) \\ \delta\theta(n) \end{bmatrix}. \quad (7)$$

$\delta d = \frac{d_R + d_L}{2}$, 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.

B. SRWNN controller

In this subsection, we design the SRWNN based direct adaptive control system for path tracking of the mobile robot. Since the kinematics of the mobile robot given in (6) consists of two inputs and three outputs, two SRWNN controllers must be used for generating each control input δd and $\delta\theta$. The overall controller architecture based on direct adaptive control scheme is shown in Fig. 2. In this architecture, SRWNNC1 and SRWNNC2 denote two SRWNN controllers for controlling the control input δd and $\delta\theta$, respectively. And in order to consider the accurate position of the mobile robot in global coordinate frame, the sum of the squared past errors $\sqrt{e_1^2(n-1)+e_2^2(n-1)}$ is used as the input of the controllers. The past control signal $\delta d(n-1)$ and the sum of the squared past errors $\sqrt{e_1^2(n-1)+e_2^2(n-1)}$ are fed into the SRWNNC1 so that the current control input $\delta d(n)$ is generated. And also, $\delta\theta(n-1)$, $\sqrt{e_1^2(n-1)+e_2^2(n-1)}$ and $e_3(n-1)$ are used as the input of SRWNNC2 for generating the current control signal $\delta\theta(n)$. Accordingly, two cost functions must be defined to select optimal control signals.

C. Training algorithm

Let us define two cost functions as

$$J_1(n)=\frac{1}{2}e_1^2(n)+\frac{1}{2}e_2^2(n), \tag{8}$$

$$J_2(n)=\frac{1}{2}e_3^2(n), \tag{9}$$

where, $e_1(n)=x_r(n)-x_c(n)$, $e_2(n)=y_r(n)-y_c(n)$, and $e_3(n)=\theta_r(n)-\theta(n)$. Here, $x_r(n)$, $y_r(n)$, and $\theta_r(n)$ denote the current 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

$$\begin{aligned}W_{1,2}(n+1)&=W_{1,2}(n)+\Delta W_{1,2}(n)\\&=W_{1,2}(n)+\bar{\eta}_{1,2}\left(-\frac{\partial J_{1,2}(n)}{\partial W_{1,2}(n)}\right),\end{aligned} \tag{10}$$

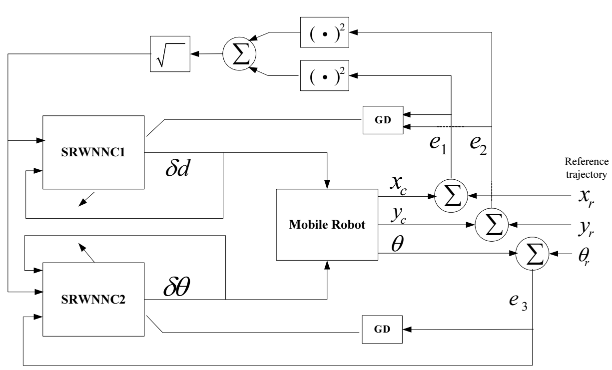


Fig. 2. The proposed control structure for the mobile robot

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 the weights of 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

$$\begin{aligned}\frac{\partial J_1(n)}{\partial W_1(n)}&=-e_1(n)\frac{\partial x_c(n)}{\partial W_1(n)}-e_2(n)\frac{\partial y_c(n)}{\partial W_1(n)}\\&=-\left[e_1(n)\frac{\partial x_c(n)}{\partial u_1(n)}+e_2(n)\frac{\partial y_c(n)}{\partial u_1(n)}\right]\frac{\partial u_1(n)}{\partial W_1(n)},\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{\partial J_2(n)}{\partial W_2(n)}&=-e_3(n)\frac{\partial \theta(n)}{\partial W_2(n)}\\&=-e_3(n)\frac{\partial \theta(n)}{\partial u_2(n)}\frac{\partial u_2(n)}{\partial W_2(n)},\end{aligned} \tag{12}$$

where, $u_1(n)=\delta d(n)$ and $u_2(n)=\delta\theta(n)$. And $\partial x_c(n)/\partial u_1(n)$, $\partial y_c(n)/\partial u_1(n)$, and $\partial\theta(n)/\partial u_2(n)$ denote the system sensitivity. They can be computed from (6). And also, the components of the Jacobian of control inputs u_1 and u_2 with respect to each weighting vector $W_{1,2}$ are computed by (4) as follows:

$$\frac{\partial u_{1,2}(n)}{\partial a_{1,2k}(n)}=x_{1,2k}, \tag{13}$$

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

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

$$\frac{\partial u_{1,2}(n)}{\partial \alpha_{1,2jk}(n)}=\frac{w_{1,2j}}{d_{1,2jk}}\phi_{1,2jk}(n-1)\frac{\partial \Phi_{1,2j}(x)}{\partial z_{1,2jk}}, \tag{16}$$

$$\frac{\partial u_{1,2}(n)}{\partial w_{1,2j}(n)}=\Phi_{1,2j}(x), \tag{17}$$

where,

$$\frac{\partial \Phi_{1,2j}}{\partial z_{1,2jk}}=\phi(z_{1,2j1})\phi(z_{1,2j2})\cdots\dot{\phi}(z_{1,2jk})\cdots\phi(z_{1,2jN_i}),$$

$$\dot{\phi}(z_{1,2jk})=\frac{\partial \phi_{1,2j}}{\partial z_{1,2jk}}=(z_{1,2jk}^2-1)\exp(-\frac{1}{2}z_{1,2jk}^2).$$

IV. Stability Analysis via ALRs

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 trained by GD method is related to select the appropriate learning rates. But, in the GD method using the static learning rates, it is difficult to choose optimal learning rates because appropriate learning rates are usually selected by trial and error. To solve this problem, we develop some convergence theorems for selecting appropriate learning rates adaptively.

Let us define a discrete Lyapunov function as

$$V(n) = \frac{1}{2}\sum_{\zeta=1}^{3} e_\zeta^2(n), \tag{18}$$

where, $e_1(n)$, $e_2(n)$, and $e_3(n)$ are the control errors. The change in the Lyapunov function is obtained by

$$\Delta V(n) = V(n+1) - V(n) = \frac{1}{2}\sum_{\zeta=1}^{3}[e_\zeta^2(n+1) - e_\zeta^2(n)]. \tag{19}$$

Three error differences can be represented by [15]

$$\Delta e_1(n) = e_1(n+1) - e_1(n) \approx \left[\frac{\partial e_1(n)}{\partial W_1^i(n)}\right]^T \Delta W_1^i(n), \tag{20}$$

$$\Delta e_2(n) = e_2(n+1) - e_2(n) \approx \left[\frac{\partial e_2(n)}{\partial W_1^i(n)}\right]^T \Delta W_1^i(n), \tag{21}$$

$$\Delta e_3(n) = e_3(n+1) - e_3(n) \approx \left[\frac{\partial e_3(n)}{\partial W_2^i(n)}\right]^T \Delta W_2^i(n), \tag{22}$$

where, $W_1^i(n)$ and $W_2^i(n)$ are an arbitrary component of the weighting vectors $W_1(n)$ and $W_2(n)$, respectively. And the corresponding changes of them are denoted by $\Delta W_1^i(n)$ and $\Delta W_2^i(n)$. Using (10)∼(12), ΔW_1 and ΔW_2 are obtained by

$$\Delta W_1^i(n) = \eta_1^i\left[e_1(n)\frac{\partial x_c(n)}{\partial u_1(n)} + e_2(n)\frac{\partial y_c(n)}{\partial u_1(n)}\right]\frac{\partial u_1(n)}{\partial W_1^i(n)}, \tag{23}$$

$$\Delta W_2^i(n) = \eta_2^i e_3(n)\frac{\partial \theta(n)}{\partial u_2(n)}\frac{\partial u_2(n)}{\partial W_2^i(n)}, \tag{24}$$

where η_1^i and η_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 $W_2^i(n)$, respectively.

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 SRWNNC1 and SRWNNC2 respectively, and define $\mathbf{C}_{1,2,max}$ as

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

where $\|\cdot\|$ represents the Euclidean norm. Then, the asymptotic convergence of SRWNNC1 and SRWNNC2 is guaranteed if $\eta_{1,2}^i$ are chosen to satisfy

$$0 < \eta_1^i < 2/[(S_x^2 + S_y^2)(C_{1,max}^i)^2], \tag{25}$$

$$0 < \eta_2^i < 2/[(S_\theta C_{2,max}^i)^2], \tag{26}$$

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

Proof: See the Appendix A. ■

Corollary 1: From conditions of Theorem 1, the maximum learning rates which guarantee the convergence are

$$\eta_1^{i,M} = 1/[(S_x^2 + S_y^2)(C_{1,max}^i)^2], \tag{27}$$

$$\eta_2^{i,M} = 1/[(S_\theta C_{2,max}^i)^2]. \tag{28}$$

Proof: ρ and γ can represent as follows:

$$\rho \geq -\frac{1}{2}(C_{1,max}^i)^4(S_x^2 + S_y^2) \cdot \left(\eta_{1c} - \frac{1}{(C_{1,max}^i)^2(S_x^2 + S_y^2)}\right)^2 + \frac{1}{2(S_x^2 + S_y^2)}, \tag{29}$$

$$\gamma \geq -\frac{1}{2}(C_{2,max}^i)^4 S_\theta^2\left(\eta_2^i - \frac{1}{(C_{2,max}^i)^2 S_\theta^2}\right)^2 + \frac{1}{2S_\theta^2}. \tag{30}$$

From (29) and (30), we obtain the maximum learning rates which guarantee the convergence (see (27) and (28)). This completes the proof. ■

Theorem 2: Let η_1^a and η_2^a be the learning rates for the input direct weights of SRWNNC1 and SRWNNC2 respectively. The asymptotic convergence of SRWNNC1 and SRWNNC2 is guaranteed if the learning rates η_1^a and η_2^a satisfy:

$$0 < \eta_1^a < 2/[(S_x^2 + S_y^2)N_{1i}|x_{1max}|^2], \tag{31}$$

$$0 < \eta_2^a < 2/[S_\theta^2 N_{2i}|x_{2max}|^2], \tag{32}$$

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

Proof:

$$C_1^1(n) = \frac{\partial u_1(n)}{\partial a_1(n)} = \mathbf{X},$$

where $\mathbf{X} = [x_{1,1} \quad x_{1,2} \quad \cdots \quad x_{1,N_{1i}}]^T$ is the input vector of SRWNNC1. Then, we have $\|C_1^1(n)\| \leq \sqrt{N_{1i}}|x_{1,max}|$. And also, $C_2^1(n)$ can be determined by the same method as $C_1^1(n)$. Therefore, from Theorem 1 , we obtain (31) and (32). ■

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 SRWNNC1 and SRWNNC2, respectively. The asymptotic convergence is guaranteed if the learning rates satisfy:

$$0 < \eta_1^m, \eta_1^\alpha < \frac{2}{(S_x^2 + S_y^2)N_{1w}N_{1i}} \cdot \left[\frac{|d_{1,min}|}{|w_{1,max}|2\exp(-0.5)}\right]^2, \tag{33}$$

$$0 < \eta_2^m, \eta_2^\alpha < \frac{2}{S_\theta^2 N_{2w}N_{2i}}\left[\frac{|d_{2,min}|}{|w_{2,max}|2\exp(-0.5)}\right]^2, \tag{34}$$

$$0 < \eta_1^d < \frac{2}{(S_x^2 + S_y^2)N_{1w}N_{1i}} \left[\frac{|d_{1,min}|}{|w_{1,max}|2\exp(0.5)} \right]^2, \quad (35)$$

$$0 < \eta_2^d < \frac{2}{S_\theta^2 N_{2w}N_{2i}} \left[\frac{|d_{2,min}|}{|w_{2,max}|2\exp(0.5)} \right]^2, \quad (36)$$

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

Proof: See the Appendix B. ■

Theorem 4: Let η_1^w and η_2^w be the learning rates for the weight w_1 of SRWNNC1 and the weight w_2 of SRWNNC2, respectively. Then, the asymptotic convergence is guaranteed if the learning rates satisfy:

$$0 < \eta_1^w < 2/[(S_x^2 + S_y^2)N_{1w}],$$
$$0 < \eta_2^w < 2/(S_\theta^2 N_{2w}).$$

Proof:

$$C_1^5(n) = \frac{\partial u_1(n)}{\partial w_1} = \Phi_1,$$

where $\Phi_1 = [\Phi_{1,1} \quad \Phi_{1,2} \quad \ldots \quad \Phi_{1,N_w}]^T$ is the output vector of the product layer of SRWNNC1. Then, since we have $\Phi_{1,j} \leq 1$ for all j, $\|C_1^5(n)\| \leq \sqrt{N_{1w}}$. 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 $0 < \eta_1^w < 2/((S_x^2 + S_y^2)N_{1w})$ and $0 < \eta_2^w < 2/(S_\theta^2 N_{2w})$. ■

Remark 1: From Corollary 1, the maximum learning rates of the SRWNNC1 and SRWNNC2 can be defined by using Theorems 2, 3, and 4.

V. Simulation Results

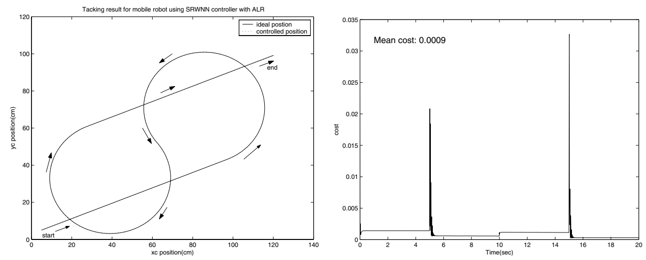
To visualize the validity of the proposed SRWNN controllers based on direct adaptive control scheme, we present simulation results for the stable path tracking of the mobile robot. The design parameters of our control system are chosen as $b = 60$, $N_{1w} = N_{2w} = 1$, $N_{1i} = 2$, and $N_{2i} = 3$ in all simulations. That is, the structures of the SRWNNC1 and SRWNNC2 are designed very simply. The ALRs defined in Remark 1 are used for training the SRWNNC1 and SRWNNC2. The sampling time is 0.01 s. To evaluate the performance of the controller, we define the cost and the mean cost as follows:

$$\text{Cost}(n) = e_1^2(n) + e_2^2(n) + e_3^2(n),$$
$$\text{Mean cost} = \frac{1}{T}\sum_{n=1}^{T}[e_1^2(n) + e_2^2(n) + e_3^2(n)],$$

where, T is the total number of samples.

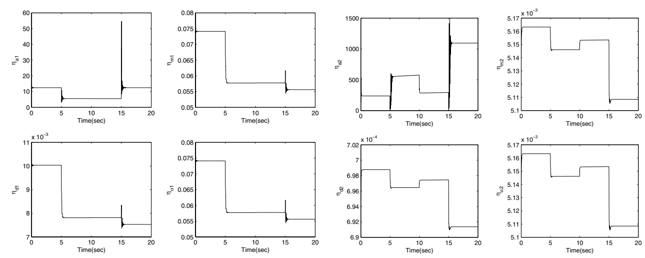
To examine the tracking performance for both the straight line and curved line, the reference trajectory is generated by the following control inputs:

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



(a) (b)

Fig. 3. (a) Tracking result for the mixed line (b) Tracking cost for the mixed line



(a) (b)

Fig. 4. The ALRs for the mixed line (a) SRWNNC1 (b) SRWNNC2

The departure posture of the mixed line is $(5, 5, \pi/8)$ and this trajectory has the variation of both the translational and rotational velocities. Fig. 3(a) presents the tracking control result using the proposed control system. The cost, as shown in Fig. 3(b), converges rapidly to almost zero after t = 5, 10, and 15 s because the SRWNN controllers adapt to the reference trajectory varying every five seconds. The ALRs for training SRWNNC1 and SRWNNC2 are shown in Figs. 4(a) and 4(b), respectively. Note that the new optimal learning rates are found rapidly by the ALRs algorithms in the instant of changing both the translational and rotational velocities of the reference trajectory. Since the translational velocity for generating the reference trajectory is held at t = 10 s, the ALRs for training the SRWNNC1 do not vary at t = 10 s, see Fig. 4(a). But the ALRs for learning the SRWNNC2 vary at t = 5, 10, and 15 s because the rotational velocity for generating the reference trajectory varies every five seconds, see Fig. 4(b). These results are caused since the SRWNNC1 and the SRWNNC2 are designed for controlling the x-y position and the angle of a mobile robot, respectively. Through all simulation results, note that SRWNNC1 and SRWNNC2 using the ALRs can adapt stably to the variation of the complex path.

VI. Concluding Remarks

A SRWNN-based adaptive direct control scheme has been proposed for the stable path tracking of mobile robots. In the control scheme, two SRWNN controllers have been designed for generating the control inputs, the translational and rotational displacement of the mobile robot. Two SRWNN controllers have been trained by the GD method with the ALRS. Since the SRWNN has the ability to store the past information of the network, it can adapt rapidly to the dynamic environment of mobile robots. By using the discrete Lyapunov theorem, the stability for the whole control scheme has been carried out and the ALRs have been also established for the stable path tracking of the mobile robot. Simulation results have shown that the proposed control system has an on-line adapting ability for controlling the mobile robot though the desired trajectories are complex.

APPENDIX

A. The proof of Theorem 1

From (18), $V(n)>0$. Using (19) $\sim$ (24), the change in the Lyapunov function is

$$\begin{aligned}\Delta V(n) &= \sum_{\zeta=1}^{3}\Delta e_{\zeta}(n)\left[e_{\zeta}(n)+\frac{1}{2}\Delta e_{\zeta}(n)\right]\\ &= -(e_1(n)S_x+e_2(n)S_y)^2\left[\eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\right.\\ &\quad\left.\cdot\left(1-\frac{1}{2}(S_x^2+S_y^2)\eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\right)\right]\\ &\quad -e_3^2(n)S_\theta^2\left[\eta_2^i\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\right.\\ &\quad\left.\cdot\left(1-\frac{1}{2}\eta_2^iS_\theta^2\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\right)\right]\\ &= -(e_1(n)S_x+e_2(n)S_y)^2\rho-e_3^2(n)\gamma,\end{aligned}$$

where,

$$\rho\geq\eta_1^i\left\|\frac{\partial u_1(n)}{\partial W_1^i(n)}\right\|^2\left(1-\frac{1}{2}(S_x^2+S_y^2)\eta_1^i(C_{1,max}^i)^2\right), \quad (37)$$

$$\gamma\geq\eta_2^i\left\|\frac{\partial u_2(n)}{\partial W_2^i(n)}\right\|^2\left(1-\frac{1}{2}\eta_2^iS_\theta^2(C_{2,max}^i)^2\right). \quad (38)$$

If $\rho>0$ and $\gamma>0$ are satisfied, $\Delta V(n)<0$. Thus, the asymptotic convergence of the proposed control system is guaranteed. Here, we obtain (25) and (26). This completes the proof. ■

B. The proof of Theorem 3

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 $|g(t)|<1,\forall g\in\mathbb{R}$.

1) The learning rate η_1^m of the translation weight m_1:

$$\begin{aligned}C_1^2(n) &= \sum_{j=1}^{N_{1w}}w_{1,j}\left(\frac{\partial\Phi_{1,j}}{\partial m_1}\right)\\ &< \sum_{j=1}^{N_{1w}}w_{1,j}\left\{\sum_{k=1}^{N_{1i}}max\left(\frac{\partial\phi(z_{1,jk})}{\partial z_{1,jk}}\frac{\partial z_{1,jk}}{\partial m_1}\right)\right\} \quad (39)\\ &< \sum_{j=1}^{N_{1w}}w_{1,j}\left\{\sum_{k=1}^{N_{1i}}max\left(2\exp(-0.5)\left(-\frac{1}{d_1}\right)\right)\right\}. \quad (40)\end{aligned}$$

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, (39) is obviously smaller than (40). Then, we have

$$\begin{aligned}\|C_1^2(n)\| &< \sum_{j=1}^{N_{1w}}w_{1,j}\sqrt{N_{1i}}\left(\frac{-2\exp(-0.5)}{d_{1,min}}\right)\\ &< \sqrt{N_{1w}}\sqrt{N_{1i}}|w_{1,max}|\left|\frac{2\exp(-0.5)}{d_{1,min}}\right|.\end{aligned}$$

And also, $C_2^2(n)$, $C_1^3(n)$, $C_2^3(n)$, $C_1^4(n)$, and $C_2^4(n)$ can be determined by the same method as $C_1^2(n)$. Accordingly, from Theorem 1, we can find (33), (34), (35), and (36). ■

REFERENCES

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

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

[3] S. X. Yang, H. Yang, and M. Q. H. Meng, "Neural dynamics based full-state tracking control of a mobile robot," *Proc. of the IEEE Int'l Conf. Robotics and Automation,* 2004, pp. 4614-4619.

[4] C. D. Sousa, E. M. Hemerly, and R. K. H. Galvão, "Adaptive control for mobile robot using wavelet networks," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 32, no. 4, 2002, pp. 493-504.

[5] T. Hu, S. X. Yang, F. Wang, and G. S. Mittal, "A neural network controller for a nonholonomic mobile robot with unknown robot parameters," *Proc. of the IEEE Int'l Conf. Robotics and Automation,* 2002, pp. 3540-3545.

[6] S. Yildirim, "Robot trajectory control using neural networks," *Electronics Letters,* vol. 38, no. 19, 2002, pp. 1111-1113.

[7] Z. P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: a case study in backsteepping," *Automatica,* vol. 33, no 7, 1997, pp. 1393-1399.

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

[9] J. Zhang, G. Walter, Y. Miao, and W. N. W. Lee, "Wavelet neural networks for function learning," *IEEE Trans. Signal Processing,* vol. 43, no. 6, 1995, pp. 1485-1497.

[10] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input-output modeling," *Neurocomputing,* vol. 20, Issuses 1-3, 1998, pp. 173-188.

[11] C. H. Lee, and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural network," *IEEE Trans. Fuzzy Systems,* vol. 8, no. 4, 2000, pp. 349-366.

[12] R. J. Wai, and J. M. Chang, "Intelligent control of induction servo motor drive via wavelet neural network," *Electric Power Systems Research*, vol. 61, 2002, pp. 67-76.

[13] J. T. Choi, Y. H. Choi, and J. B. Park, "Stable intelligent control of chaotic systems via wavelet neural network," *Proc. of ICCAS2003,* Gyeongju, Korea, 2003, pp. 316-321.

[14] K. B. Kim, J. B. Park, and Y. H. Choi and G. Chen, "Control of chaotic nonlinear systems using radial basis network approximatiors," *Information Science,* vol. 130, 2000, pp. 165-183.

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

[16] D. Haesloop and B. R. Holt, "A neural network structure for system identification," *Proc. of ACC,* 1990, pp. 2460-2465.

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