자기 회귀 웨이블릿 신경 회로망을 이용한 다이나믹 시스템의 동정: 적응 학습률 기반 수렴성 분석

Identification of Dynamic Systems Using a Self Recurrent Wavelet Neural Network: Convergence Analysis Via Adaptive Learning Rates

유 성 진, 최 윤 호, 박 진 배*
(Sung Jin Yoo, Yoon Ho Choi, and Jin Bae Park)

Abstract : This paper proposes an identification method using a self recurrent wavelet neural network (SRWNN) for dynamic systems. The architecture of the proposed SRWNN is a modified model of the wavelet neural network (WNN). But, 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. Thus, in the proposed identification architecture, the SRWNN is used for identifying nonlinear dynamic systems. The gradient descent method with adaptive learning rates (ALRs) is applied to train the parameters of the SRWNN identifier (SRWNNI). The ALRs are derived from the discrete Lyapunov stability theorem, which are used to guarantee the convergence of an SRWNNI. Finally, through computer simulations, we demonstrate the effectiveness of the proposed SRWNNI.

Keywords : self-recurrent wavelet neural network, gradient descent method, adaptive learning rate, identification

I. Introduction

The neural networks have been shown to be of benefit for use in the identification and control of nonlinear systems[1]. Most people have used the multi-layer perceptron (MLP) and the back- propagation (BP) training algorithm or gradient descent (GD) algorithm with fixed learning rates (FLRs)[2] to solve the dynamical problems. This method has two disadvantages. First, the MLP is a static mapping and without the aid of tapped delays, it does not represent a dynamic system mapping. Second, it is difficult to find optimal learning rate in the BP algorithm with FLRs. To solve the first problem, the recurrent neural network (RNN) is proposed for identifying dynamic systems[3-5]. The RNN has some important abilities not found in MLP, such as attractor dynamics and information storage for later use. However, due to its multi-layered structure, the training processes converge too slowly. To solve the second problem, many researchers have proposed the learning methods such as genetic algorithm and evolutionary program which are the global optimization method[6,7]. However, the computational burden of these algorithms· degrades the performance of real-time control systems.

Recently the wavelet neural network (WNN), which absorbs the advantage of high resolution of wavelets and learning of MLP, is proposed to guarantee the fast convergence, which is used for the identification and control of the nonlinear dynamic systems[8-12]. The WNN has one hidden layer, which is suitable for approximating functions with local nonlinearities and fast variations because of their intrinsic properties of finite support and self-similarity. But, the WNN has a disadvantage that it can be used only for static problems due to its feedforward network structure. That is, the WNN is inefficient in solving temporal problems. Hence, we present a self recurrent wavelet neural network (SRWNN), which combines the properties of attractor dynamics of RNN and the fast convergence of WNN. The SRWNN, as a modified model of the WNN, has a mother wavelet layer composed of self-feedback neurons. Since a self-feedback neuron can store the past information of the network, it can capture the dynamic response of the system. This modification permits that the SRWNN can be applied well to the complex dynamic system, though the SRWNN has less wavelet nodes than the WNN. Thus, the structure of SRWNN can be simpler than that of the WNN. Accordingly, the SRWNN is more suitable than the WNN for real-time control application.

In this paper, we propose the design method of the identifier using SRWNN to solve the identification problem for dynamic systems. And also, unlike the existing GD algorithm, the GD algorithm with ALRs is used for training the SRWNNI. The ALRs are derived from the discrete Lyapunov stability theorem, which are used to guarantee the convergence of an identifier in the proposed system. Finally, we consider the dynamic nonlinear systems to show the effectiveness of the proposed SRWNN identifier (SRWNNI).

* 책임저자(Corresponding Author)
논문접수 : 2004. 11. 3., 채택확정 : 2005. 6. 23.
유성진, 박진배 : 연세대학교 전기전자공학과
(niceguy1201@control.yonsei.ac.kr/jbpark@yonsei.ac.kr)
최윤호 : 경기대학교 전자공학부(yhchoi@kyonggi.ac.kr)

This paper is organized as follows: In Section II, we describe the architecture of SRWNN. The identification mechanism and the learning algorithm of SRWNNI are presented in Section III. Section IV presents the stability analysis for the convergence of SRWNNI and then ALRs are established for training of SRWNNI. Simulation results are discussed in Section V. Section VI gives the conclusion of this paper.

II. Self Recurrent Wavelet Neural Network

This Section discusses the SRWNN structure. A schematic diagram of the proposed SRWNN structure is shown in Fig. 1, 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 ϕ_{jk} of each node is derived from its mother wavelet ϕ as follows:

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

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

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

where θ_{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 system is stored 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, θ_{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 $\theta_{jk} = 0$.

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

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

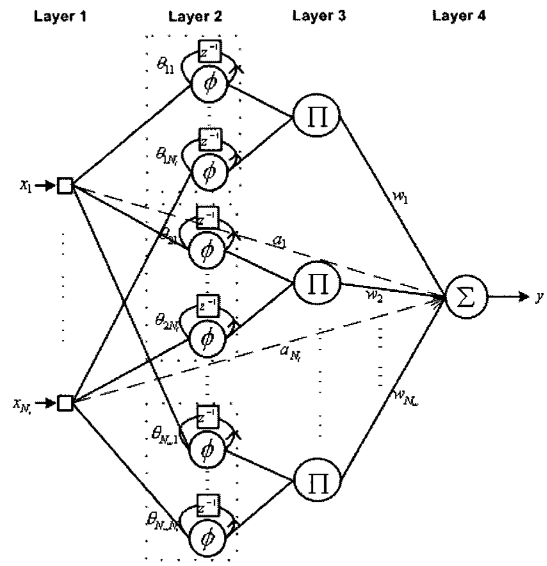


Fig. 1. SRWNN structure.

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 output of SRWNN is composed by each self-recurrent wavelet 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 nodes. W is the weighting vector of SRWNN represented by:

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

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

III. Identification Using SRWNN

1. Identification Architecture

We use the series-parallel method for the identification of dynamic system. The identification model of dynamic system is composed of the SRWNN and tapped delay lines. The current input, the past inputs, and the past outputs of the system are fed into the SRWNN and the error $e(n)$ between the actual system output and the SRWNN output is used to train the SRWNN. The SRWNN output will attract the output trajectories of the dynamic system. The current output of the SRWNNI represents as follows[12]:

$$y_I(n) = f(y(n-1), \ldots, y(n-N_s), u(n), \ldots, u(n-N_e)), \quad (6)$$

where, N_s and N_e indicates the number of the past outputs and the past input variables respectively. And $y(n)$ and $u(n)$ are the dynamic system output and the identification input respectively.

2. Training of the SRWNNI

This Subsection discusses the training method of the SRWNNI. Our goal is to minimize the following quadratic cost function:

$$J(n)=\frac{1}{2}[y(n)-y_I(n)]^2=\frac{1}{2}e^2(n), \tag{7}$$

where $y(n)$ is the plant output and $y_I(n)$ is the current output of the SRWNNI for the discrete time n. By using the GD method, the weight values of SRWNNI are adjusted so that the error is minimized after a given number of training cycles. The GD method may be defined as:

$$\begin{aligned} W(n+1) &= W(n)+\Delta W(n) \\ &= W(n)+\bar{\eta}\left(-\frac{\partial J(n)}{\partial W(n)}\right), \end{aligned} \tag{8}$$

where, $\bar{\eta}=diag[\eta_a, \eta_m, \eta_d, \eta_\theta, \eta_w]$ represents the learning rate matrix for weights of the SRWNNI and W is weighting vector, which is defined in Section II.

The partial derivative of the cost function with respect to $W(n)$ is

$$\frac{\partial J(n)}{\partial W(n)}=e(n)\frac{\partial e(n)}{\partial W(n)}=-e(n)\frac{\partial y_I(n)}{\partial W(n)}. \tag{9}$$

By applying the chain rule recursively, the error term for each layer is first calculated, and then the parameters in the corresponding layers are adjusted. The components of the weighting vector are

$$\frac{\partial y_I(n)}{\partial a_k(n)}=x_k(n), \tag{10}$$

$$\frac{\partial y_I(n)}{\partial m_{jk}(n)}=-\frac{w_j}{d_{jk}}\frac{\partial \Phi_j(X)}{\partial z_{jk}}, \tag{11}$$

$$\frac{\partial y_I(n)}{\partial d_{jk}(n)}=-\frac{w_j}{d_{jk}}z_{jk}\frac{\partial \Phi_j(X)}{\partial z_{jk}}, \tag{12}$$

$$\frac{\partial y_I(n)}{\partial \theta_{jk}(n)}=\frac{w_j}{d_{jk}}\phi_{jk}(n-1)\frac{\partial \Phi_j(X)}{\partial z_{jk}}, \tag{13}$$

$$\frac{\partial y_I(n)}{\partial w_j(n)}=\Phi_j(X), \tag{14}$$

where,

$$\frac{\partial \Phi_j(X)}{\partial z_{jk}}=\phi(z_{j1})\phi(z_{j2})\cdots\dot{\phi}(z_{jk})\cdots\phi(z_{jN_i}),$$

$$\dot{\phi}(z_{jk})=\frac{\partial \phi_j}{\partial z_{jk}}=(z_{jk}^2-1)\exp\left(-\frac{1}{2}z_{jk}^2\right).$$

IV. Convergence Analysis Via ALRs

Let us define a discrete Lyapunov function as

$$V(n)=\frac{1}{2}e^2(n), \tag{15}$$

where, $e(n)$ is the identification error. The change in the Lyapunov function is obtained by

$$\begin{aligned} \Delta V(n) &= V(n+1)-V(n) \\ &= \frac{1}{2}[e^2(n+1)-e^2(n)]. \end{aligned} \tag{16}$$

The error difference can be represented by[3]

$$\begin{aligned} \Delta e(n) &= e(n+1)-e(n) \\ &\approx \left[\frac{\partial e(n)}{\partial W_i(n)}\right]^T \Delta W_i(n), \end{aligned} \tag{17}$$

where, $W_i(n)$ is an arbitrary component of the weighting vector $W(n)$. And the corresponding changes of this is denoted by $\Delta W_i(n)$.

Using (8) and (9), $\Delta W_i(n)$ is obtained by

$$\Delta W_i(n)=\eta_i e(n)\frac{\partial y_I(n)}{\partial W_i(n)}, \tag{18}$$

where, η_i is an arbitrary diagonal element of the learning rate matrix $\bar{\eta}$ corresponding to the weight component $W_i(n)$.

Theorem 1: Let $\bar{\eta}=[\eta_1\ \eta_2\ \eta_3\ \eta_4\ \eta_5]=[\eta_a\ \eta_m\ \eta_d\ \eta_\theta\ \eta_w]$ be the learning rates for the weights of the SRWNNI and define $C_{\max}$ as

$$\begin{aligned} C_{\max} &= [C_{1,max}\ \ C_{2,max}\ \ C_{3,max}\ \ C_{4,max}\ \ C_{5,max}]^T \\ &= \Big[\max_n\left\|\frac{\partial y_I(n)}{\partial a(n)}\right\|\ \ \max_n\left\|\frac{\partial y_I(n)}{\partial m(n)}\right\|\ \ \max_n\left\|\frac{\partial y_I(n)}{\partial d(n)}\right\| \\ &\qquad \max_n\left\|\frac{\partial y_I(n)}{\partial \theta(n)}\right\|\ \ \max_n\left\|\frac{\partial y_I(n)}{\partial w(n)}\right\|\Big], \end{aligned}$$

where $\|\cdot\|$ represents the Euclidean norm. Then, the asymptotic convergence of the SRWNNI is guaranteed if η_i is chosen to satisfy

$$0<\eta_i<\frac{2}{(C_{i,max})^2}, \qquad i=1,\ldots,5. \tag{19}$$

Proof: From (15), $V(n)>0$. Thus, from (17) and (18), the change of the Lyapunov function is

$$\begin{aligned} \Delta V(n) &= \frac{1}{2}[e^2(n+1)-e^2(n)] \\ &= \Delta e(n)[e(n)+\frac{1}{2}\Delta e(n)] \\ &= \left[\frac{\partial e(n)}{\partial W_i(n)}\right]^T \eta_i e(n)\frac{\partial y_I(n)}{\partial W_i(n)} \\ &\quad \cdot\left\{e(n)+\frac{1}{2}\left[\frac{\partial e(n)}{\partial W_i(n)}\right]^T \eta_i e(n)\frac{\partial y_I(n)}{\partial W_i(n)}\right\} \\ &= -\left[\frac{\partial y_I(n)}{\partial W_i(n)}\right]^T \eta_i e(n)\frac{\partial y_I(n)}{\partial W_i(n)} \\ &\quad \cdot\left\{e(n)-\frac{1}{2}\left[\frac{\partial y_I(n)}{\partial W_i(n)}\right]^T \eta_i e(n)\frac{\partial y_I(n)}{\partial W_i(n)}\right\} \\ &= -e^2(n)\left[\eta_i\left\|\frac{\partial y_I(n)}{\partial W_i(n)}\right\|^2\right. \\ &\quad \left.\cdot\left(1-\frac{1}{2}\eta_i\left\|\frac{\partial y_I(n)}{\partial W_i(n)}\right\|^2\right)\right] \\ &= -e^2(n)\gamma, \end{aligned}$$

where,

$$\gamma=\left[\eta_i\left\|\frac{\partial y_j(n)}{\partial W_i(n)}\right\|^2\left(1-\frac{1}{2}\eta_i\left\|\frac{\partial y_j(n)}{\partial W_i(n)}\right\|^2\right)\right]$$
$$\geq\left[\eta_i\left\|\frac{\partial y_j(n)}{\partial W_i(n)}\right\|^2\left(1-\frac{1}{2}\eta_i(C_{i,\max})^2\right)\right].$$

If $\gamma>0$ is satisfied, $\Delta V(n)<0$. Accordingly, the asymptotic convergence of the SRWNNI is guaranteed.

Here, we obtain (19). This completes the proof of the theorem. ■

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

$$\eta_i=\frac{1}{(C_{i,\max})^2},\quad i=1,\ldots,5. \tag{20}$$

Proof: γ can represent as follows:

$$\gamma=\left[\eta_i\left\|\frac{\partial y_j(n)}{\partial W_i(n)}\right\|^2\left(1-\frac{1}{2}\eta_i\left\|\frac{\partial y_j(n)}{\partial W_i(n)}\right\|^2\right)\right]$$
$$\geq-\frac{1}{2}(C_{i,\max})^4\left(\eta_i-\frac{1}{(C_{i,\max})^2}\right)^2+\frac{1}{2}>0. \tag{21}$$

From (21), the learning rate which guarantees maximum convergence is (20). This completes the proof. ■

Theorem 2: Let η_a be the learning rate of the input direct weight for the SRWNNI. The asymptotic convergence is guaranteed if the learning rate η_a satisfies:

$$0<\eta_a<\frac{2}{N_i|x_{\max}|^2}, \tag{22}$$

where N_i denote the input number of the SRWNNI. $x_{\max}$ is maximum value of SRWNNI's inputs.

Proof:

$$C_1(n)=\frac{\partial y_j(n)}{\partial a(n)}=X,$$

where $X=[x_1\ x_2\ \cdots\ x_{N_i}]^T$ is the input vector of the SRWNNI. Then we have $|C_1(n)|\leq\sqrt{N_i}|x_{\max}|$. Therefore, from Theorem 1, we obtain (22). ■

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

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

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

Theorem 3: Let η_m, η_d and η_θ be the learning rates of the translation, dilation and self-feedback weight for the SRWNNI, respectively. The asymptotic convergence is guaranteed if the learning rates satisfy:

$$0<\eta_m,\eta_\theta<\frac{2}{N_wN_i}\left[\frac{|d_{\min}|}{|w_{\max}|(2\exp(-0.5))}\right]^2, \tag{23}$$

$$0<\eta_d<\frac{2}{N_wN_i}\left[\frac{|d_{\min}|}{|w_{\max}|(2\exp(0.5))}\right]^2, \tag{24}$$

where N_w is the number of nodes in the product layer of the SRWNNI.

Proof:

1) The learning rate η_m of the translation weight m:

$$C_2(n)=\frac{\partial y_j(n)}{\partial m(n)}=\sum_{j=1}^{N_w}w_j\left(\frac{\partial\Phi_j(X)}{\partial m}\right)$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(\frac{\partial\phi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial m}\right)\right\} \tag{25}$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(2\exp(-0.5)\left(-\frac{1}{d}\right)\right)\right\}. \tag{26}$$

According to Lemma 2,

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

Thus, (25) is obviously smaller than (26). Then we have

$$\|C_2(n)\|<\sum_{j=1}^{N_w}w_j\sqrt{N_i}\left(\frac{-2\exp(-0.5)}{d_{\min}}\right)<\sqrt{N_w}\sqrt{N_i}|w_{\max}|\left|\frac{2\exp(-0.5)}{d_{\min}}\right|.$$

Accordingly, from Theorem 1, we find (23).

2) The learning rate η_d of the dilation weight d:

$$C_3(n)=\frac{\partial y_j(n)}{\partial d(n)}=\sum_{j=1}^{N_w}w_j\left(\frac{\partial\Phi_j(X)}{\partial d}\right)$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(\frac{\partial\phi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial d}\right)\right\} \tag{27}$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(2\exp(0.5)\left(\frac{1}{d}\right)\right)\right\}. \tag{28}$$

According to Lemmas 1 and 2,

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

Thus, (27) is obviously smaller than (28). Then we have

$$\|C_3(n)\|<\sum_{j=1}^{N_w}w_j\sqrt{N_i}\left(\frac{2\exp(0.5)}{d_{\min}}\right)<\sqrt{N_w}\sqrt{N_i}|w_{\max}|\left|\frac{2\exp(0.5)}{d_{\min}}\right|.$$

Accordingly, from Theorem 1, we find (24).

3) The learning rate η_θ of the self-feedback weight θ:

$$C_4(n)=\frac{\partial y_j(n)}{\partial\theta(n)}$$
$$=\sum_{j=1}^{N_w}w_j\left(\frac{\partial\Phi_j(X)}{\partial\theta}\right) \tag{29}$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(\frac{\partial\phi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial\theta}\right)\right\}$$
$$<\sum_{j=1}^{N_w}w_j\left\{\sum_{k=1}^{N_i}\max\left(2\exp(-0.5)\left(-\frac{\phi_{jk}(n-1)}{d}\right)\right)\right\}. \tag{30}$$

According to Lemma 2,

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

Thus, (29) is obviously smaller than (30). Then we have

$$\|C_4(n)\|<\sum_{j=1}^{N_w} w_j\sqrt{N_i}\left(\frac{2\exp(-0.5)}{d_{min}}\right)$$
$$<\sqrt{N_w}\sqrt{N_i}|w_{max}|\left|\frac{2\exp(-0.5)}{d_{min}}\right|.$$

Accordingly, from Theorem 1, we find (23). ■

Theorem 4: Let η_w be the learning rate for weights w of the SRWNNI. Then, asymptotic stability is guaranteed if the learning rate satisfies:

$$0<\eta_w<\frac{2}{N_w}. \tag{31}$$

Proof:

$$C_5(n)=\frac{\partial y_I(n)}{\partial w}=\Phi,$$

where $\Phi=[\Phi_1\Phi_2\cdots\Phi_{N_w}]^T$ is the output vector of the product layer of the SRWNNI. Then, since we have $\Phi_j\le 1$ for all j, $C_5(n)\le\sqrt{N_w}$. Accordingly, from Theorem 1, we find (31). ■

Remark 1: From Corollary 1, the learning rates of the SRWNNI for guaranteeing the maximum convergence are

$$\eta_{a,M}=\frac{1}{N_i|x_{max}|^2},$$

$$\eta_{m,M}=\eta_{\theta,M}=\frac{1}{N_wN_i}\left[\frac{|d_{min}|}{|w_{max}|(2\exp(-0.5))}\right]^2,$$

$$\eta_{d,M}=\frac{1}{N_wN_i}\left[\frac{|d_{min}|}{|w_{max}|(2\exp(0.5))}\right]^2,$$

$$\eta_{w,M}=\frac{2}{N_w}.$$

V. Simulation results

To visualize the validity of the proposed SRWNN based identification method, we present two simulation results for nonlinear dynamic systems and compare the performance of the SRWNN with that of the WNN structure models introduced in [11]. The first example involves identifying a nonlinear dynamic system, and the second example involves identifying a Lorenz system, which is a chaotic nonlinear.

Table 1. Performance comparison in example 1.

	SRWNN (ALR)	SRWNN (FLR)	WNN (ALR)	WNN (FLR)
No. of Product layers	1	2	5	5
Parameters	17	30	50	50
Learning rates	Adaptive	0.05	Adaptive	0.9
Time Step	1000	1000	1000	1000
MSE error	0.00032	0.0013	0.0023	0.0362

1. Identification of Nonlinear Dynamic System

This subsection considers the following dynamic plant with time delay inputs[5]:

$$y(n+1)=0.72y(n)+0.0025y(n-1)u(n-1)$$
$$+0.01u^2(n-1)+0.2u(n-3).$$

Here the checking input signal $u(n)$ was used to determine the identification results,

$$u(n)=\begin{cases}\sin(\frac{\pi n}{25}), & 0<n<250\\ 1.0, & 250\le n<500\\ -1.0, & 500\le n<750\\ 0.3\sin(\frac{\pi n}{25})+0.1\sin(\frac{\pi n}{32}) & \\ +0.6\sin(\frac{\pi n}{10}) & 750\le n<1000.\end{cases}$$

In this example, to examine the performance of the SRWNNI, we compare the performance of the SRWNN trained by ALRs and FLRs with that of the WNN trained by ALRs and FLRs respectively. Figure 2 shows the outputs of the plant and SRWNN with ALRs. The experimental results demonstrate the perfect identification capability of the SRWNN model. The errors of four case, SRWNN with ALRs and FLRs, WNN trained by ALRs and FLRs, are shown in Fig. 3. The figure reveals that the SRWNN model with ALRs have the smallest identification error and fastest convergence among four cases. The initial values of ALRs are chosen as $\eta_a=\eta_m=\eta_d=\eta_\theta=\eta_w=0.05$. And for remaining time steps, the ALRs are computed by Remark 1 every time step. Figure 4 shows the time evolution of the ALRs. In Fig. 4, the optimal learning rates of SRWNNI are found adaptively reducing the identification error. And the simulation environments and performance results of the SRWNN with ALRs are compared with those of other methods.

As shown in Table 1, the number of adjustable parameters and MSE in the SRWNN trained by ALRs are smaller than in other cases under the same time step.

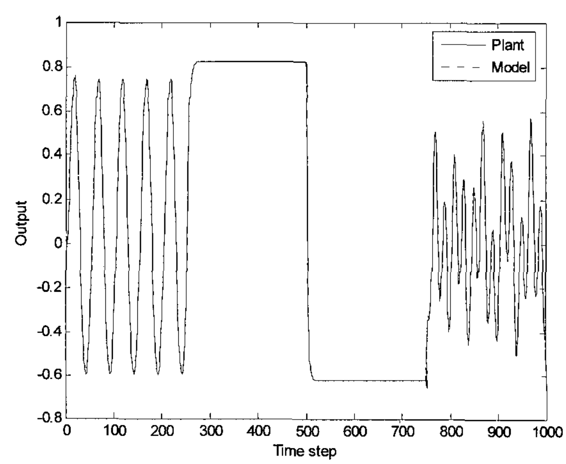


Fig. 2. Outputs of the plant and the SRWNN with ALRs.

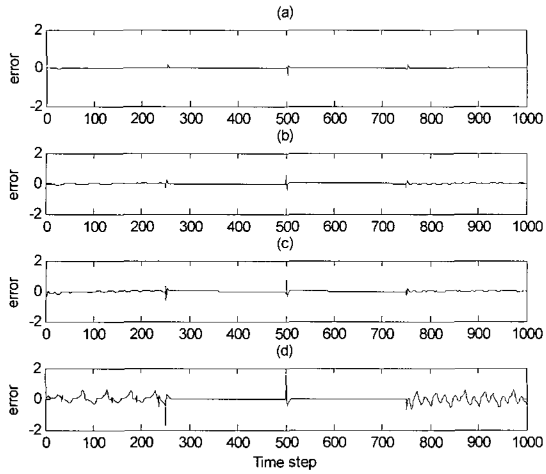


Fig. 3 Errors between the desired output and various model output. (a) SRWNN with ALRs. (b) SRWNN with FLRs. (c) WNN with ALRs (d) WNN with FLRs.

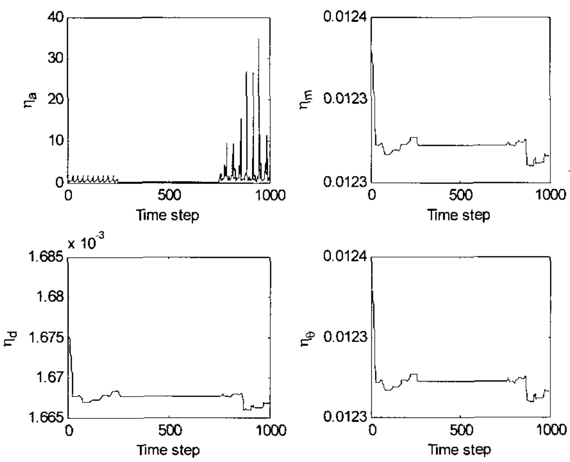


Fig. 4. ALRs of SRWNNI for the dynamic system.

2. Identification of Chaotic System

We consider the Lorenz system, the continuous-time chaotic nonlinear system. In this simulation, to demonstrate the performance of the proposed method, we compare the performance of SRWNN trained by ALRs and FLRs with that of the WNN trained by ALRs and FLRs. The state equation of this chaotic system with x, y, and z as state variables is expressed by[13]

$$\begin{bmatrix} \dot{x}(n) \\ \dot{y}(n) \\ \dot{z}(n) \end{bmatrix} = \begin{bmatrix} a(y(n)-x(n)) \\ bx(n)-y(n)-x(n)z(n) \\ x(n)y(n)-cz(n) \end{bmatrix}$$

which, with $a=10$, $b=28$, and $c=8/3$ produces a strange chaotic attractor, as shown Fig. 5. In the identification of the Lorenz system, we used three SRWNNI to identify three states (x, y, z) respectively. And we define the initial states of the plant as $(0.1673, 0.5651, 0.9854)$, and these of WNN and SRWNN models as $(0, 0, 0)$. The initial values of ALRs of all SRWNNIs and WNNIs are chosen as $\eta_a = \eta_m =$

Table 2. Performance comparison in example 2.

	SRWNN (ALR)	SRWNN (FLR)	WNN (ALR)	WNN (FLR)
No. of Product layers	1	2	5	5
Parameters	9	18	27	27
Learning rates	Adaptive	0.0006	Adaptive	0.0001
Time Step	2000	2000	2000	2000
MSE error of x_state	0.0018	0.0750	0.0041	0.2009
MSE error of y_state	0.0009	0.0318	0.0088	0.5250
MSE error of z_state	0.0018	0.0622	0.0401	0.1413

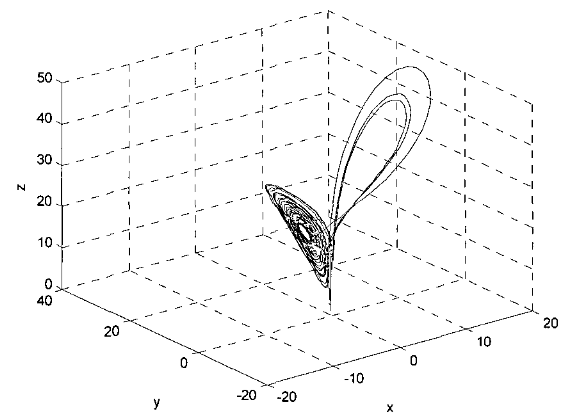


Fig. 5. The chaotic attractor of lorenz system (check data).

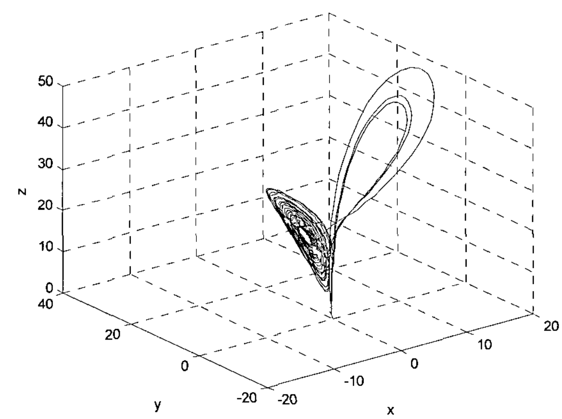


Fig. 6. Identification result of the SRWNNI with ALRs for the lorenz system.

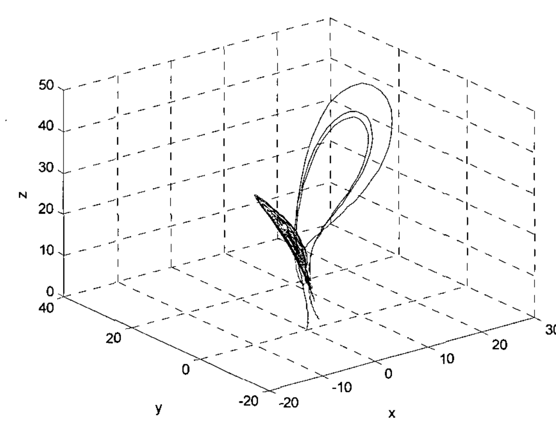


Fig. 7. Identification result of the SRWNNI with FLRs for the lorenz system.

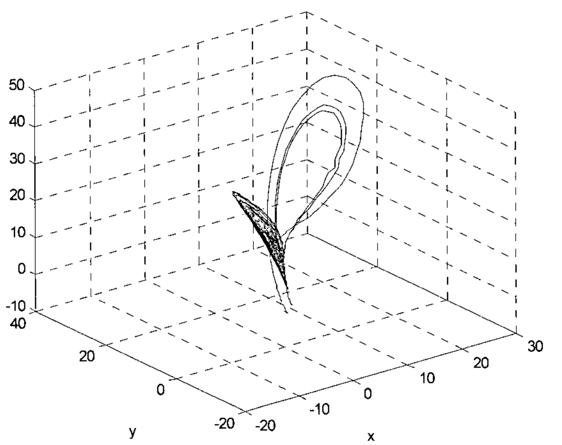


Fig. 8. Identification result of the WNNI with ALRs for the lorenz system.

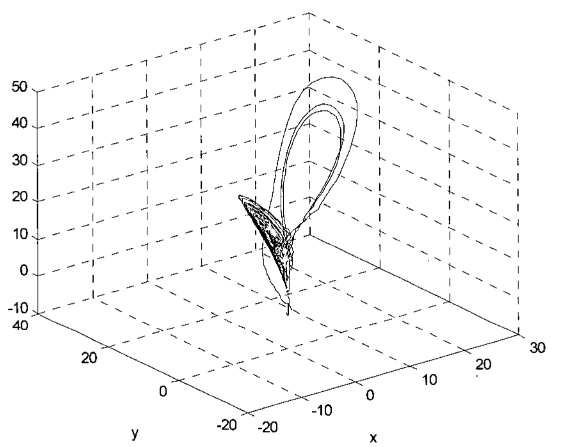


Fig. 9. Identification result of the WNNI with FLRs for the lorenz system.

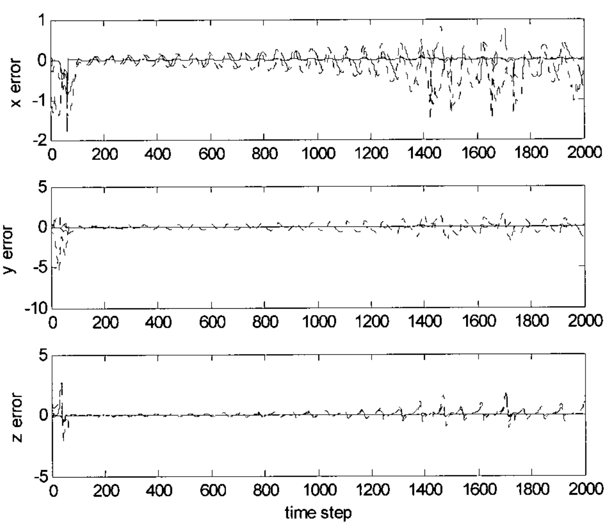


Fig. 11. Identification errors of SRWNN with ALRs(solid line), SRWNN with FLRs(dash-dotted line), WNN with ALRs(dashed line), and WNN with FLRs(dotted line) for the lorenz system.

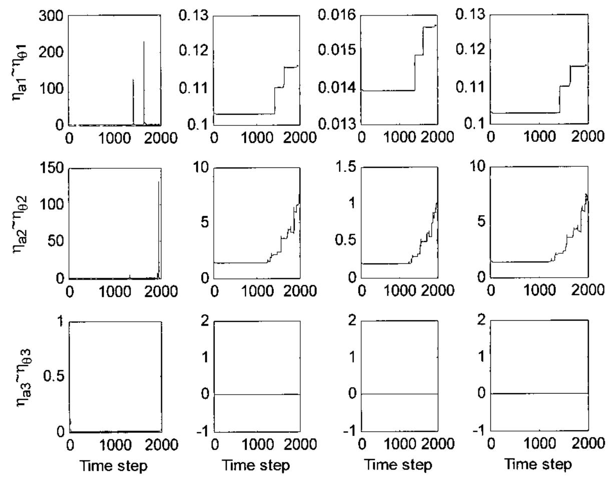


Fig. 12. ALRs of SRWNNIs for the lorenz system.

$\eta_d = \eta_\theta = \eta_w = 0.05$. And for remaining time steps, the ALRs are computed every time step by Remark 1. Figures 6, 7, 8, and 9 show the phase plane of Lorenz system after training for the WNN and SRWNN models respectively. The identification errors of all states for the WNN and SRWNN are compared in Fig. 10. Figure 11 shows the time evolution of the ALRs for weights of three SRWNNIs. In Fig. 10, the optimal learning rates of SRWNNI are found adaptively reducing the identification error. Note that the WNN model is inappropriate for Lorenz systems because of its static mapping as shown in Fig. 9. Moreover, the comparison in Table 2 shows that the number of adjustable parameters and MSE in the SRWNN trained by ALRs are smaller than in other cases under the same time step.

VI. Conclusion

This paper has proposed the self-recurrent wavelet neural network based identification method for the nonlinear dynamic systems. The SRWNN, which is a generalized network of the WNN, consists of four layers including a mother wavelet layer with self-feedback neurons. Since the self-feedback units act as memory elements, the SRWNN has the capability of temporarily storing information. Thus, the SRWNNI can be used as a good tool to identify the dynamic nonlinear systems. Based on the Lyapunov approach, the convergence theorems for SRWNNI were proven and the optimal adaptive learning rates were also established. Finally, the proposed SRWNN has been applied to a dynamic nonlinear system and the Lorenz system, which is a representative continuous- time chaotic system respectively. Simulation results show that the SRWNN has the three advantages. First, the SRWNN has the simpler network structure than the WNN. Second, the SRWNNI successfully can approximate a dynamic system mapping as accurately as desired. Third, the ALRs have better performance than the FLRs.

References

[1] K. S. Narendra, and K. Parthasarathy, "Identification and control of dynamic system using neural network," *IEEE Trans. on Neural Networks,* vol. 1, no. 1, pp. 4-27, 1990.

[2] D. Rumalhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in D. Rumelhart, and J. McClelland (Eds.), *Parallel Distributed Processing,* Cambrige, MA:MIT Press, vol. 1, pp. 318-362, 1986.

[3] 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.

[4] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," *Proc. of Int. Joint Conference on Neural Networks,* vol. 1, pp. 365-372, 1989.

[5] R. J. Williams and D. Zipser, "A learning algorithm for continually running recurrent neural network," *Neural Computing,* vol. 1, no. 2, pp. 270-280, 1989.

[6] S. H. Ling, H. K. Lam, F. H. F. Leung, and P. K. S. Tam, "A novel GA-based neural network for short-term load forecasting," *Proc. of Int. Joint Conference on Neural Networks,* vol. 3, pp. 2761-2766, 2002.

[7] S. Yao, C. Wei, and Z. He, "Evolving wavelet neural networks for function approximation," Electronics Letters, vol. 32, no. 4, pp. 360, 1996.

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

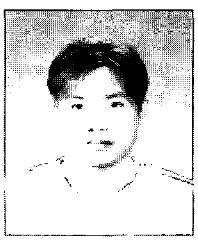
[9] J. T. Choi, Y. H. Choi, and J. B. Park, "Stable intelligent control of chaotic systems via wavelet neural network," *Proc. of Int. Conf. on Control, Automation and Systems,* pp. 316-321, 2003.

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

[11] L. Cao, Y. Hong, H. Fang, and G. He, "Predicting chaotic time series with wavelets networks," *Physica D,* vol. 85, no. 1-2, pp. 225-238, 1995.

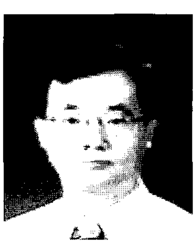
[12] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input-output modeling," *Neurocomputing,* vol. 20, no. 1-3, pp. 173-188, 1998.

[13] S. H. Strogatz, *Nonlinear Dynamics and Chaos,* Addison-Wesley, 1994.



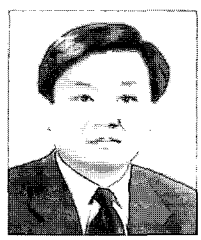
Sung Jin Yoo

received the B.S. and M.S. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2003 and 2005 respectively. Currently, he is pursuing a Ph.D. degree in the Dept. of Electrical and Electronic Engineering at Yonsei University, Seoul, Korea. His research interests include nonlinear adaptive control, neural network, robotic systems, and chaos control.



Jin Bae Park

received the B.S. degree in Electrical Engineering from Yonsei University, Seoul, Korea, in 1977 and the M.S. and Ph.D. degrees in Electrical Engineering from Kansas State University, Manhattan, in 1985 and 1990, respectively. Since 1992 he has been with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, where he is currently a Professor. His research interests include robust control and filtering, nonlinear control, mobile robot, fuzzy logic control, neural networks, genetic algorithms, and Hadamard-transform spectroscopy. He had served as vice-president for the Institute of Control, Automation, and Systems Engineers. He is serving as an Editor for the International Journal of Control, Automation, and Systems.



Yoon Ho Choi

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