Stable Predictive Control of Chaotic Systems Using Self-Recurrent Wavelet Neural Network

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

Abstract: In this paper, a predictive control method using self-recurrent wavelet neural network (SRWNN) is proposed for chaotic systems. Since the SRWNN has a self-recurrent mother wavelet layer, it can well attract the complex nonlinear system though the SRWNN has less mother wavelet nodes than the wavelet neural network (WNN). Thus, the SRWNN is used as a model predictor for predicting the dynamic property of chaotic systems. The gradient descent method with the adaptive learning rates is applied to train the parameters of the SRWNN based predictor and controller. The adaptive learning rates are derived from the discrete Lyapunov stability theorem, which are used to guarantee the convergence of the predictive controller. Finally, the chaotic systems are provided to demonstrate the effectiveness of the proposed control strategy.

Keywords: Self-recurrent wavelet neural network, predictive control, adaptive learning rate, gradient descent, chaos control.

1. INTRODUCTION

Chaos control is very active area of research and has sustained much interest due to potential applications in various areas. Chaos is a special feature of complex parametric nonlinear dynamical systems, which has the random-like behavior usually shown in statistical systems although it is associated with deterministic dynamics. Recently many researchers have managed to use modern powerful methods for controlling chaotic systems [1-3]. But most of them can be applied to control chaotic systems when the exact or at least the approximate mathematical model for chaotic systems is available. To solve this shortcoming, predictive control methods, which are considered as a kind of adaptive control strategy, were introduced for controlling unknown chaotic systems [4].

On the other hand, the intelligent techniques using network structures such as neural network (NN) and radial basis function network (RBFN) have been developed to control chaotic systems [5,6]. Also, some papers successfully applied these network structures to the model predictor of predictive control [7,8]. But they have some drawbacks, which come from their inherent characteristics. NN has some limitation such as slow convergence, settlement of local minima. In the case of RBFN, even if the RBFN can represent any function that is in the space spanned by the family of basis functions, the basis functions are generally not orthogonal. That is, RBFN representation is not unique and is probably not the most efficient [9]. To overcome these problems, the WNN, which absorbs the advantage of high resolution of wavelets and the advantages of learning and feedforward of neural networks, is proposed to guarantee the fast convergence and is used as a new powerful tool for function approximation, because of their intrinsic properties of finite support and self-similarity [9-11]. Since the basis functions of WNN are orthogonal, the WNN provides a unique and efficient representation for the given functions. However, 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 not the most suitable in solving temporal problems like predicting the behaviors of complex chaotic systems. Therefore, we develop a new structure, self-recurrent wavelet neural network (SRWNN), which combines the properties of attractor dynamics of recurrent neural network (RNN) [12,13] and the fast convergence of WNN to solve the control problem for chaotic systems. The proposed SRWNN, a modified model of the WNN, has a mother wavelet layer composed of self-

Manuscript received February 3, 2004; revised November 19, 2004; accepted December 21, 2004. Recommended by Editorial Board member Jin Young Choi under the direction of Editor-in-Chief Myung Jin Chung.

Sung Jin Yoo and Jin Bae Park are with the Department of Electrical and Electronic Engineering, Yonsei University, Seodaemun-Gu, Seoul 120-749, Korea (e-mails: {niceguy 1201, jbpark}@control.yonsei.ac.kr).

Yoon Ho Choi is with the School of Electronic Engineering, Kyonggi University, Kyonggi-Do, Suwon 442-760, Korea (e-mail: yhchoi@kuic.kyonggi.ac.kr).

* Corresponding author.

feedback neurons. Since a self-feedback neuron can store past information of the network, it can capture the dynamic response of the system. This modification allows the SRWNN to be applied well to the complex chaotic systems, though the SRWNN has less wavelet nodes than the WNN. Thus, the structure of the SRWNN can be simpler than that of the WNN. Accordingly, the SRWNN is more suitable in real-time control application than the WNN.

In this paper, we propose the design method of the SRWNN based controller using the predictive control scheme to solve the control problem for chaotic systems. Here, the SRWNN is used as the model predictor for chaotic systems. Also the back-propagation algorithm with adaptive learning rates is used for training the SRWNN. The adaptive learning rates are derived in the sense of discrete Lyapunov stability analysis, which are used to guarantee the convergence of the SRWNN predictor and controller in the proposed control system. Finally, we consider the chaotic nonlinear systems to show the effectiveness of the proposed SRWNN based predictive control.

This paper is organized as follows: In Section 2, we discuss the identification of chaotic systems using the SRWNN. Here, the architecture and training algorithm of the SRWNN is presented, and the stability of the SRWNN based predictor is analyzed. Section 3 presents the SRWNN based predictive control strategy and also the stability of controller is analyzed. Simulation results are discussed in Section 4. Section 5 gives the conclusion of this paper.

2. IDENTIFICATION OF CHAOTIC SYSTEMS USING THE SRWNN

This Section discusses the identification of chaotic systems using the SRWNN. We first describe the SRWNN structure, and then the identification method with the SRWNN is presented for chaotic systems. Finally, we derive the convergence theorems for selecting the appropriate learning rates to identify the chaotic systems.

2.1. The SRWNN structure

In this paper, we consider the SRWNN structure with multi-input and single-output. A schematic diagram of the proposed SRWNN structure is shown in Fig. 1, which has N_i inputs, one output, and $N_i \times N_w$ the mother wavelet nodes ("wavelons"). 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 wavelon layer. Each node of this layer consists of a wavelon with a self-feedback loop. In this paper, we select the first derivative of a gaussian function, $\varphi(x) = -x\exp(-\frac{1}{2}x^2)$ as a mother wavelet function. A wavelet φ_{jk} of each node is derived from its mother wavelet function φ as follows:

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

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

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

where θ_{jk} denotes the weight of the self-feedback loop. The input of this layer contains the memory term $\varphi_{jk}(n-1)$, 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 wavelons than the WNN, the SRWNN can attract well 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 wavelons as follows:

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

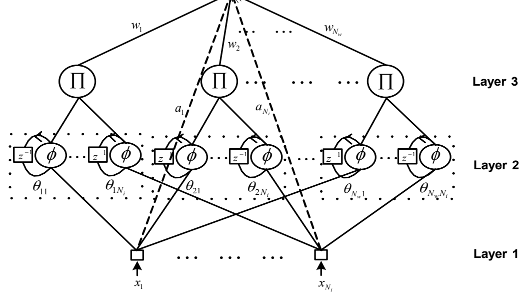


Fig. 1. Structure of the proposed SRWNN.

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:

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

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 nodes. W is the weighting vector of SRWNN:

$$W=\begin{bmatrix} a_k & m_{jk} & d_{jk} & \theta_{jk} & w_j \end{bmatrix}^T, \tag{5}$$

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.

2.2. Identification method for chaotic systems

This paper uses the series-parallel method for the identification of chaotic systems. The identification structure is shown in Fig. 2. The identification model for the chaotic 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 $e_I(n)$ between the actual system output and the SRWNN output is used to train the SRWNN. The SRWNN output will attract the output trajectories of chaotic systems. The current SRWNN output represents as follows [14,15]:

$$\begin{aligned} y_I(n)=f(&y_c(n-1),\ y_c(n-2),\ \ldots, \\ &y_c(n-N_s),\ u(n),\ u(n-1),\ \ldots,\ u(n-N_e)), \end{aligned} \tag{6}$$

where N_s and N_e indicate the number of the past outputs and the past input state variables, respectively. And also, $y_c(n)$ and $u(n)$ are the chaotic system output and the identification input, respectively.

In this paper, we use the gradient-descent (GD) method to train the SRWNN structure. Our goal is to minimize the following quadratic cost function:

$$J_I(n)=\frac{1}{2}[y_c(n)-y_I(n)]^2=\frac{1}{2}e_I^2(n), \tag{7}$$

where $y_c(n)$ is the chaotic system output and $y_I(n)$

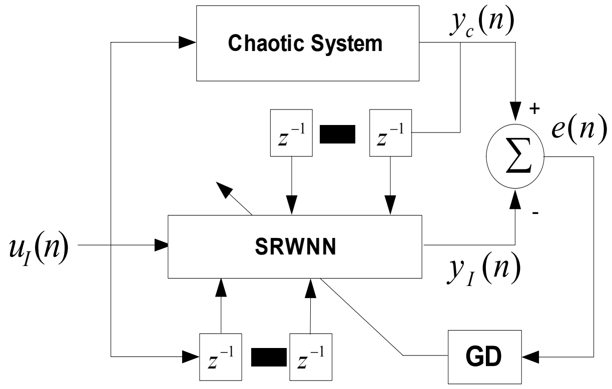


Fig. 2. Identification structure using the SRWNN.

$=\hat{y}_c(n)$ is the current output of the SRWNN for the discrete time n. By using the GD method, the weight values of the SRWNN are adjusted so that the error is minimized after a given number of training cycles.

The gradient-descent method may be defined as:

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

where η_I represents the learning rate of the SRWNN and W is weighting vector, which is defined in Section 2.1.

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

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

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

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

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

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

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

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

where,

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

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

2.3. Stability analysis for identification

To analyze the stability for the SRWNN based identifier, let us define a discrete Lyapunov function as

$$V_I(n) = J_I(n) = \frac{1}{2}e_I^2(n), \tag{15}$$

where $e_I(n)$ is the identification error. The subscript I indicates the parameters related to the SRWNN based identifier.

The change in the Lyapunov function is obtained by

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

The error difference can be represented by [13,16]

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

where $\Delta W_I = \left[\Delta a_I \quad \Delta m_I \quad \Delta d_I \quad \Delta\theta_I \quad \Delta w_I\right]^T$ represent s the change of a weight vector of the SRWNN based identifier. Using (8) and (9), ΔW_I is obtained by

$$\Delta W_I = \bar{\eta}_I e_I(n)\frac{\partial y_I(n)}{\partial W_I}, \tag{18}$$

where $\bar{\eta}_I = diag\left[\eta_I^a,\ \eta_I^m,\ \eta_I^d,\ \eta_I^\theta,\ \eta_I^w\right]$ is the learning rates of the tuning parameters. And, $\frac{\partial y_I(n)}{\partial W_I} = \left[\frac{\partial y_I(n)}{\partial a_I} \ \frac{\partial y_I(n)}{\partial m_I} \ \frac{\partial y_I(n)}{\partial d_I} \ \frac{\partial y_I(n)}{\partial \theta_I} \ \frac{\partial y_I(n)}{\partial w_I}\right]^T$ denotes the Jacobian of the identifier output $y_I(n)$ with respect to the weighting vector of the SRWNN based identifier.

Theorem 1: Let $\bar{\eta}_I = diag\left[\eta_I^1,\ \eta_I^2,\ \eta_I^3,\ \eta_I^4,\ \eta_I^5\right] = diag\left[\eta_I^a,\ \eta_I^m,\ \eta_I^d,\ \eta_I^\theta,\ \eta_I^w\right]$ be the learning rates for the weights of the SRWNN based identifier and define $C_{I,max}$ as

$$\begin{aligned}C_{I,max} &= \left[C_{I,max}^1 \quad C_{I,max}^2 \quad C_{I,max}^3 \quad C_{I,max}^4 \quad C_{I,max}^5\right]^T\\ &= \Bigg[max_n\left|\frac{\partial y_I(n)}{\partial a_I}\right| \quad max_n\left|\frac{\partial y_I(n)}{\partial m_I}\right|\\ &\qquad max_n\left|\frac{\partial y_I(n)}{\partial d_I}\right| \quad max_n\left|\frac{\partial y_I(n)}{\partial \theta_I}\right| \quad max_n\left|\frac{\partial y_I(n)}{\partial w_I}\right|\Bigg].\end{aligned}$$

Then, the asymptotic stability is guaranteed if η_I^i are chosen to satisfy

$$0 < \eta_I^i < \frac{2}{(C_{I,max}^i)^2}, \quad i = 1,..,5.$$

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

$$\begin{aligned}\Delta V_I(n) &= \frac{1}{2}[e_I^2(n+1) - e_I^2(n)]\\ &= \Delta e_I(n)[e_I(n) + \frac{1}{2}\Delta e_I(n)]\\ &= \left[\frac{\partial e_I(n)}{\partial W_I}\right]^T \bar{\eta}_I e_I(n)\frac{\partial y_I(n)}{\partial W_I}\\ &\quad \bullet\left\{e_I(n) + \frac{1}{2}\left[\frac{\partial e_I(n)}{\partial W_I}\right]^T \bar{\eta}_I e_I(n)\frac{\partial y_I(n)}{\partial W_I}\right\}\\ &= -\left[\frac{\partial y_I(n)}{\partial W_I}\right]^T \bar{\eta}_I e_I(n)\frac{\partial y_I(n)}{\partial W_I}\\ &\quad \bullet\left\{e_I(n) - \frac{1}{2}\left[\frac{\partial y_I(n)}{\partial W_I}\right]^T \bar{\eta}_I e_I(n)\frac{\partial y_I(n)}{\partial W_I}\right\}\\ &= -e_I^2(n)\sum_{i=1}^{5}\left[\eta_I^i\left(\frac{\partial y_I(n)}{\partial W_I^i}\right)^2\left\{1 - \frac{1}{2}\eta_I^i\left(\frac{\partial y_I(n)}{\partial W_I^i}\right)^2\right\}\right]\\ &= -\gamma_I e_I^2(n).\end{aligned}$$

If $\gamma_I > 0$, $\Delta V_I(n) < 0$. Accordingly, the asymptotic convergence of the SRWNN based identifier is guaranteed. Here, we obtain $0 < \eta_I^i < 2/(C_{I,max}^i)^2$, $i = 1,...,5$. This completes the proof of the theorem. □

Corollary 1: If the learning rates are chosen as $\eta_I = \eta_I^i (i = 1,...5)$, the maximum learning rate which guarantees convergence is $\eta_I^M = 1/(C_{I,max})^2$.

Proof: Let $C_{I,max}$ be defined as

$$C_{I,max} = max_n \|C_I(n)\| = max_n \left\| \left[\frac{\partial y_I(n)}{\partial a_I} \quad \frac{\partial y_I(n)}{\partial m_I} \quad \frac{\partial y_I(n)}{\partial d_I} \quad \frac{\partial y_I(n)}{\partial \theta_I} \quad \frac{\partial y_I(n)}{\partial w_I} \right] \right\|,$$

where $\|\cdot\|$ represents the Euclidean norm.

$$\begin{aligned}\gamma_I &= \eta_I \left\| \frac{\partial y_I(n)}{\partial W_I} \right\|^2 \left(1 - \frac{1}{2}\eta_I \left\| \frac{\partial y_I(n)}{\partial W_I} \right\|^2 \right) \\ &= \eta_I \|C_I(n)\|^2 \left(1 - \frac{1}{2}\eta_I \|C_I(n)\|^2 \right) \qquad (19) \\ &= -\frac{1}{2}\|C_I(n)\|^4 \left(\eta_I - \frac{1}{\|C_I(n)\|^2} \right)^2 + \frac{1}{2} > 0.\end{aligned}$$

From (19), the maximum learning rate which guarantees convergence is $\eta_I^M = 1/(C_{I,max})^2$. This completes the proof. □

Theorem 2: Let η_I^a be the learning rate of input direct weights for the SRWNN based identifier. The asymptotic convergence is guaranteed if the learning rate η_I^a satisfies:

$$0 < \eta_I^a < \frac{2}{N_i \, |x_{I,max}|^2},$$

where N_i is the input number of the SRWNN based identifier. $|x_{I,max}|$ denote the maximum value of the absolute values of the SRWNN based identifier's input.

Proof:

$$\begin{aligned}C_I^1(n) = \frac{\partial y_I(n)}{\partial a_I} &= \left[\frac{\partial y_I(n)}{\partial a_1} \quad \frac{\partial y_I(n)}{\partial a_2} \quad \cdots \quad \frac{\partial y_I(n)}{\partial a_{N_i}} \right] \\ &= [x_{I,1} \quad x_{I,2} \quad \cdots \quad x_{I,N_i}] \\ &= X,\end{aligned}$$

where X is the input vector of the SRWNN based identifier and $a_I = [a_1 \ a_2 \ \cdots \ a_{N_i}]^T$. Then we have $\|C_I^1(n)\| \le \sqrt{N_i} \, |x_{I,max}|$. Therefore, from Theorem 1, we find that $0 < \eta_I^a < 2/(C_{I,max}^1)^2 = 2/(N_i |x_{I,max}|^2)$ □

In order to prove Theorem 22, the following lemmas are used.

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

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

Theorem 3: Let η_I^m, η_I^d and η_I^θ be the learning rates of the translation, dilation and self-feedback weight for the SRWNN based identifier, respectively. The asymptotic convergence is guaranteed if the learning rates satisfy:

$$0 < \eta_I^m, \eta_I^\theta < \frac{2}{N_w N_i} \left[\frac{1}{|w_{I,max}| \left(\frac{2\exp(-0.5)}{|d_{I,min}|} \right)} \right]^2,$$

$$0 < \eta_I^d < \frac{2}{N_w N_i} \left[\frac{1}{|w_{I,max}| \left(\frac{2\exp(0.5)}{|d_{I,min}|} \right)} \right]^2,$$

where $|w_{I,max}|$ and $|d_{I,min}|$ are the maximum value of the absolute values of the output weight w_I and the dilation weight d_I, respectively.

Proof:

1) The learning rate η_I^m of the translation weight m_I:

$$\begin{aligned}C_I^2(n) &= \frac{\partial y_I(n)}{\partial m_I} \\ &= \sum_{j=1}^{N_w} w_{I,j} \left(\frac{\partial \Phi_j}{\partial m_I} \right) \\ &= \sum_{j=1}^{N_w} w_{I,j} \left\{ \sum_{k=1}^{N_i} \frac{\prod_{k=1}^{N_i} \varphi(z_{jk})}{\varphi(z_{jk})} \left(\frac{\partial \varphi(z_{jk})}{\partial z_{jk}} \frac{\partial z_{jk}}{\partial m_I} \right) \right\} \qquad (20) \\ &< \sum_{j=1}^{N_w} w_{I,j} \left\{ \sum_{k=1}^{N_i} max \left(\frac{\partial \varphi(z_{jk})}{\partial z_{jk}} \frac{\partial z_{jk}}{\partial m_I} \right) \right\} \\ &< \sum_{j=1}^{N_w} w_{I,j} \left\{ \sum_{k=1}^{N_i} max \left(2\exp(-0.5) \left(-\frac{1}{d_I} \right) \right) \right\}. \qquad (21)\end{aligned}$$

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, (20) is obviously smaller than (21). Then we have

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

Accordingly, from Theorem 1, we find that

$$0 < \eta_I^m < \frac{2}{(C_{I,max}^2)^2} = \frac{2}{N_w N_i}\left[\frac{1}{|\,w_{I,max}\,|\left(\frac{2\exp(-0.5)}{|d_{I,min}|}\right)}\right]^2.$$

2) The learning rate η_I^d of the dilation weight d_I :

$$C_I^3(n) = \frac{\partial y_I(n)}{\partial d_I}$$
$$= \sum_{j=1}^{N_w} w_{I,j}\left(\frac{\partial \Phi_j}{\partial d_I}\right)$$
$$= \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i}\frac{\prod_{k=1}^{N_i}\varphi(z_{jk})}{\varphi(z_{jk})}\left(\frac{\partial\varphi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial d_I}\right)\right\}$$
$$< \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i} max\left(\frac{\partial\varphi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial d_I}\right)\right\} \quad (22)$$
$$< \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i} max\left(2\exp(0.5)\left(\frac{1}{d_I}\right)\right)\right\}. \quad (23)$$

According to Lemmas 1 and 2,

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

Thus, (22) is obviously smaller than (23). Then we have

$$\left\|C_I^3(n)\right\| < \sum_{j=1}^{N_w} w_{I,j}\sqrt{N_i}\left(\frac{2\exp(0.5)}{d_{I,min}}\right)$$
$$< \sqrt{N_w}\sqrt{N_i}\,|\,w_{I,max}\,|\left|\frac{2\exp(0.5)}{d_{I,min}}\right|.$$

Accordingly, from Theorem 1, we find that

$$0 < \eta_I^d < \frac{2}{(C_{I,max}^3)^2} = \frac{2}{N_w N_i}\left[\frac{1}{|\,w_{I,max}\,|\left(\frac{2\exp(0.5)}{|d_{I,min}|}\right)}\right]^2.$$

3) The learning rate η_I^θ of the self-feedback weight θ_I :

$$C_I^4(n) = \frac{\partial y_I(n)}{\partial \theta_I}$$
$$= \sum_{j=1}^{N_w} w_{I,j}\left(\frac{\partial \Phi_j}{\partial \theta_I}\right)$$
$$= \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i}\frac{\prod_{k=1}^{N_i}\varphi(z_{jk})}{\varphi(z_{jk})}\left(\frac{\partial\varphi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial \theta_I}\right)\right\}$$
$$< \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i} max\left(\frac{\partial\varphi(z_{jk})}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial \theta_I}\right)\right\} \quad (24)$$
$$< \sum_{j=1}^{N_w} w_{I,j}\left\{\sum_{k=1}^{N_i} max\left(2\exp(-0.5)\left(\frac{\varphi_{jk}(n-1)}{d_I}\right)\right)\right\}. \quad (25)$$

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, (24) is obviously smaller than (25).

$$\left\|C_I^4(n)\right\| < \sum_{j=1}^{N_w} w_{I,j}\sqrt{N_i}\left(\frac{2\exp(-0.5)}{d_{I,min}}\right)$$
$$< \sqrt{N_w}\sqrt{N_i}\,|\,w_{I,max}\,|\left|\frac{2\exp(-0.5)}{d_{I,min}}\right|.$$

Accordingly, from Theorem 1, we find that

$$0 < \eta_I^\theta < \frac{2}{(C_{I,max}^4)^2} = \frac{2}{N_w N_i}\left[\frac{1}{|\,w_{I,max}\,|\left(\frac{2\exp(-0.5)}{|d_{I,min}|}\right)}\right]^2.$$ □

Theorem 4: Let η_I^w be the learning rate for weights w_I of the SRWNN based identifier. Then, asymptotic stability is guaranteed if the learning rate satisfies:

$$0 < \eta_I^w < \frac{2}{N_w},$$

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

Proof:

$$C_I^5(n) = \frac{\partial y_I(n)}{\partial w_I} = \Phi,$$

where, $\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_{N_w} \end{bmatrix}^T$ is the output vector of the product layer of the SRWNN based identifier. Then, since we have $\Phi_j \le 1$ for all j, $|C_I^5(n)| \le \sqrt{N_w}$. Accordingly, from Theorem 1, we find that $0 < \eta_I^w < 2/N_w$. □

Remark 1: From Corollary 1, the maximum learning rates of the SRWNN based identifier are

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

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

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

$$\eta_I^{w,M} = 1/N_w.$$

3. SRWNN BASED PREDICTIVE CONTROL

In this Section, we propose the SRWNN based predictive control method for chaotic systems. Suppose that a chaotic plant is given without precise mathematical description of its structure and parameters and that the given plant, although uncertain, has an inherent unstable periodic orbit and the system is currently in the chaotic state. The object is to design a controller, which, when being added to the plant in a feedback configuration, is used to drive the closed-loop system response to move out of the chaotic attractor and then to converge to the unstable periodic orbit or the equilibrium point.

First, we describe the overall architecture and the strategy of predictive control, and then develop the condition for the learning rate to guarantee the convergence of the proposed controller.

3.1. Predictive control using the SRWNN

1) SRWNN based one-step prediction: We first describe the one-step ahead predictive control scheme. Assume that the output data of the SRWNN are available on-line for generating the control signal. In

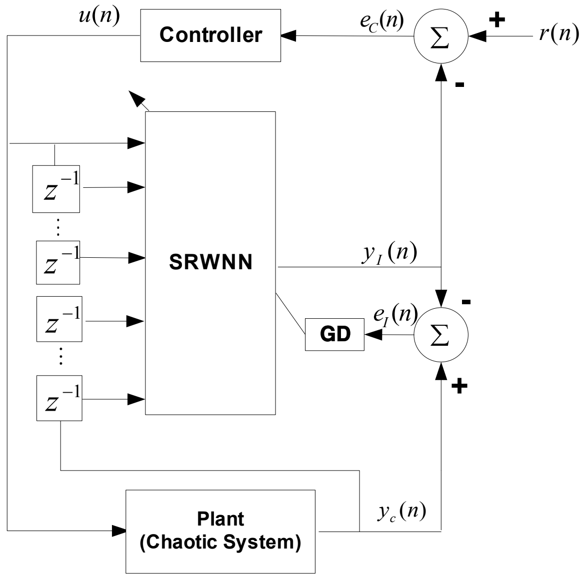


Fig. 3. SRWNN based predictive control architecture.

our design method, an on-line prediction unit based on the SRWNN is employed and a nonlinear feedback controller based on a predictive control scheme is implemented. The overall configuration of the SRWNN based predictive control system is shown in Fig. 3, where the SRWNN output $y_I(n)$ is controlled to track the reference signal $r(n)$.

Here, we use the SRWNN to predict future values of the chaotic system. Our object is to look for an optimal control signal $u(n)$ to minimize the following performance function:

$$J_C = \frac{1}{2}e_C^2(n+1), \tag{26}$$

where $e_C(n+1) = r(n+1) - y_I(n+1)$ is the control error.

To minimize J_C, $u(n)$ is recursively calculated via the GD method.

$$\begin{aligned} u(n+1) &= u(n) + \Delta u(n) \\ &= u(n) + \eta_C\left(-\frac{\partial J_C}{\partial u(n)}\right), \end{aligned} \tag{27}$$

where η_C is the learning rate of the control input. We can see that the controller relies on the SRWNN based predictor. Thus to improve the controller performance, it is necessary that the SRWNN output well approaches the chaotic system output. In this aspect, the SRWNN, which has an ability to store the past information for predicting the future output, is a suitable tool. Differentiating the cost function with respect to $u(n)$, it can be obtained by

$$\frac{\partial J_C}{\partial u(n)} = -\left[\frac{\partial y_I(n+1)}{\partial u(n)}\right]^T e_C(n+1), \tag{28}$$

where $\frac{\partial y_I(n+1)}{\partial u(n)}$ is the gradient of the SRWNN model with respect to $u(k)$. It can be analytically evaluated by using the known SRWNN structure, (4) as follows:

$$\begin{aligned}\frac{\partial y_I(n+1)}{\partial u_C(n)} &= \frac{\partial y_I(n+1)}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial u_C(n)} \\ &= \left[\sum_{j=1}^{N_w}\frac{w_j}{d_{jk}}\frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k\right]_{k=N_s+1},\end{aligned} \tag{29}$$

where $\partial \mathbf{x}/\partial u_C(n)$ is the column vector as follows:

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial u_C(n)} &= \left[\frac{\partial y(n)}{\partial u_C(n)} \quad \cdots \quad \frac{\partial y(n-N_s)}{\partial u_C(n)} \quad \frac{\partial u_C(n)}{\partial u_C(n)}\right. \\ &\qquad \left.\cdots \quad \frac{\partial u_C(n-N_e)}{\partial u_C(n)}\right] \\ &= \begin{bmatrix}0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0\end{bmatrix}^T,\end{aligned} \tag{30}$$

where $\mathbf{x}$ is the inputs of the SRWNN based model predictor.

2) Extension to multi-step prediction: Let the prediction algorithm described above be extended to a multi-step prediction control scheme, and redefine the cost function by the vectors:

$$\mathbf{J_C} = \frac{1}{2}\left[\mathbf{E_C}^T(n+1)\mathbf{E_C}(n+1)\right], \tag{31}$$

where

$$\begin{aligned}\mathbf{Y_I}(n+1) &= \left[y_I(n+1) \quad y_I(n+2) \quad \cdots \quad y_I(n+N)\right]_{N\times 1}^T, \\ \mathbf{R}(n+1) &= \left[r(n+1) \quad r(n+2) \quad \cdots \quad r(n+N)\right]_{N\times 1}^T, \\ \mathbf{E_C}(n+1) &= \left[e_C(n+1) \quad e_C(n+2) \quad \cdots \quad e_C(n+N)\right]_{N\times 1}^T \\ &= \left[r(n+1)-y_I(n+1) \quad r(n+2)-y_I(n+2)\right. \\ &\qquad \left.\cdots \quad r(n+N)-y_I(n+N)\right]_{N\times 1}^T.\end{aligned}$$

Then, we define the control signal as:

$$\mathbf{U}(n) = \left[u(n) \quad u(n+1) \quad \cdots \quad u(n+N-1)\right]_{N\times 1}^T,$$

where N denotes the number of prediction steps.

To minimize $\mathbf{J_C}$, the control input vector $\mathbf{U}(n)$ is updated via the GD method:

$$\begin{aligned}\mathbf{U}(n+1) &= \mathbf{U}(n) + \Delta\mathbf{U}(n) \\ &= \mathbf{U}(n) + \eta_C\left(-\frac{\partial \mathbf{J_C}}{\partial \mathbf{U}(n)}\right),\end{aligned} \tag{32}$$

where η_C is the learning rate of the control input. And $\frac{\partial J_C}{\partial \mathbf{U}(n)}$ is expressed by the following equation:

$$\frac{\partial \mathbf{J_C}}{\partial \mathbf{U}(n)} = -\left[\frac{\partial \mathbf{Y_I}(n+1)}{\partial \mathbf{U}(n)}\right]^T \mathbf{E_C}(n+1), \tag{33}$$

where

$$\mathbf{G} \triangleq \frac{\partial \mathbf{Y_I}(n+1)}{\partial \mathbf{U}(n)} = \begin{pmatrix} \frac{\partial y_I(n+1)}{\partial u(n)} & 0 & 0 & \cdots & 0 \\ \frac{\partial y_I(n+2)}{\partial u(n)} & \frac{\partial y_I(n+2)}{\partial u(n+1)} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_I(n+N)}{\partial u(n)} & \frac{\partial y_I(n+N)}{\partial u(n+1)} & \frac{\partial y_I(n+N)}{\partial u(n+2)} & \cdots & \frac{\partial y_I(n+N)}{\partial u(n+N-1)} \end{pmatrix}_{N\times N} \tag{34}$$

From (32) and (33), the change of the control input is

$$\Delta\mathbf{U}(n) = \eta_C \mathbf{G}^T \mathbf{E_C}(n+1), \tag{35}$$

where $\mathbf{G}$ is a jacobian matrix. Using (35), while the optimal control input is calculated, the main computation is the matrix $\frac{\partial \mathbf{Y}_I(n+1)}{\partial \mathbf{U}(n)}$. The computing procedure for this matrix is presented in Appendix.

3.2. Stability analysis for control

Similarly to Section 2.3, let us define a discrete Lyapunov function as

$$V_C(n) = \frac{1}{2}\mathbf{E_C}^T(n+1)\mathbf{E_C}(n+1), \tag{36}$$

where $\mathbf{E}_C(n+1)$ is the control error vector. The change in the Lyapunov function is obtained by

$$\begin{aligned}\Delta V_C(n) &= V_C(n+1) - V_C(n) \\ &= \frac{1}{2}\left[\mathbf{E_C}^T(n+2)\mathbf{E_C}(n+2)\right. \\ &\qquad \left.-\mathbf{E_C}^T(n+1)\mathbf{E_C}(n+1)\right].\end{aligned} \tag{37}$$

The error difference can be represented by

$$\Delta\mathbf{E_C}(n+1) \approx \left(\frac{\partial \mathbf{E_C}(n+1)}{\partial \mathbf{U}(n)}\right)\Delta\mathbf{U}(n) = -\left(\frac{\partial \mathbf{Y_I}(n+1)}{\partial \mathbf{U}(n)}\right)\Delta\mathbf{U}(n), \tag{38}$$

where $\Delta\mathbf{U}(n)$ is denoted by (35). Using (33) and (38), $\Delta\mathbf{E_C}(n+1)$ is obtained by

$$\Delta\mathbf{E_C}(n+1) = -\mathbf{G}\eta_C\mathbf{G}^T\mathbf{E_C}(n+1). \tag{39}$$

Lemma 3: Let a positive symmetric matrix $M_{N\times N}$ as:

$$M = \alpha I - A,$$

where $A_{N\times N}$ is a positive symmetric matrix and $\alpha \geq 0$ is a real number. I is a identity matrix. Then, the eigenvalues of M can represent as follows:

$$eig(M) = eig(\alpha I) - eig(A),$$

where

$$eig(M) = \begin{bmatrix}\lambda_{M,1} & \lambda_{M,2} & \cdots & \lambda_{M,N}\end{bmatrix}^T,$$
$$eig(A) = \begin{bmatrix}\lambda_{A,1} & \lambda_{A,2} & \cdots & \lambda_{A,N}\end{bmatrix}^T,$$
$$eig(\alpha I) = \begin{bmatrix}\alpha & \alpha & \cdots & \alpha\end{bmatrix}^T.$$

Theorem 5: Let η_C be the learning rate for control input. Then, the asymptotic stability is guaranteed if η_C are chosen to satisfy

$$0 < \eta_C < \frac{2}{\lambda_{max}},$$

where λ_{max} denotes the maximum eigenvalue of the matrix $\mathbf{G}^T\mathbf{G}$.

Proof: From (36), $\mathbf{V}(n) > 0$. And, from (39), the change in the Lyapunov function is

$$\begin{aligned}\Delta\mathbf{V}(n) &= \mathbf{V}(n+1) - \mathbf{V}(n)\\ &= \frac{1}{2}\Big[\mathbf{E_C}^T(n+2)\mathbf{E_C}(n+2)\\ &\quad - \mathbf{E_C}^T(n+1)\mathbf{E_C}(n+1)\Big]\\ &= \Delta\mathbf{E_C}^T(n+1)\left[\mathbf{E_C}(n+1) + \frac{1}{2}\Delta\mathbf{E_C}(n+1)\right]\\ &= -\mathbf{E_C}^T\mathbf{G}\eta_C\mathbf{G}^T\left[\mathbf{E_C} - \frac{1}{2}\mathbf{G}\eta_C\mathbf{G}^T\mathbf{E_C}\right]\\ &= -\mathbf{E_C}^T\left[\mathbf{G}\eta_C\mathbf{G}^T\left\{I - \frac{1}{2}\mathbf{G}\eta_C\mathbf{G}^T\right\}\right]\mathbf{E_C}\\ &= -\mathbf{E_C}^T\left[\Upsilon\Omega\right]\mathbf{E_C},\end{aligned}$$

where

$$\Upsilon = \mathbf{G}\eta_C\mathbf{G}^T, \tag{40}$$

$$\Omega = I - \frac{1}{2}\mathbf{G}\eta_C\mathbf{G}^T. \tag{41}$$

If Υ and Ω are positive definite matrices, $\Delta\mathbf{V}(n) < 0$. Thus, the Lyapunov stability is guaranteed. To be $\Upsilon > 0$ and $\Omega > 0$, all eigenvalues of Υ and Ω must be larger than 0. To satisfy this condition, we obtain $\eta_C > 0$ from (40) and (41) can be represented as follows:

$$\Omega = \frac{2}{\eta_C}I - \mathbf{G}^T\mathbf{G} > 0.$$

Here, using Lemma 3, $\eta_C < 2/(\lambda_{max})$ is obtained. Therefore, we obtain $0 < \eta_C < 2/(\lambda_{max})$. This completes the proof of the theorem. □

4. SIMULATION RESULTS

In this section, we apply the proposed predictive control algorithm to two chaotic systems. Firstly, the Duffing system, which is the continuous-time chaotic system, is considered. Secondly, we consider the Hénon system, which is the discrete-time chaotic system. And also, in order to evaluate the performance of the proposed SRWNN based predictive control scheme, we compare with the WNN based predictive control scheme for each chaotic system.

4.1. The Duffing system

This Subsection considers the Duffing system, which is the representative continuous-time chaotic system. The state equation of the Duffing system is

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ a_1x_1 - x_1^3 - a_2x_2 + bcos(wt) + u\end{bmatrix}, \tag{42}$$

where $a_1 = 1.1$, $a_2 = 0.4$, $b = 2.1$ and $w = 1.8$. The tracking control objective for the Duffing system is to follow the unstable periodic solution of the Duffing system. As the value of b varies, the Duffing system may have either a chaotic or a periodic solution. The reference signal is defined as one periodic solution in the case of $b = 2.3$. In this simulation, we choose the initial state of the Duffing system as $(1, 0)$ and the learning rates for the SRWNN based identifier are chosen based on Remark 1. And also, the learning rate for the control signal is chosen as 0.006, which is based on Theorem 5.

The simulation environments of both the SRWNN based predictive controller and the WNN based predictive controller are shown in Table 1. Fig. 4 compares the control results using the SRWNN and the WNN. The mean-squared error(MSE)s of the SRWNN based predictive control and the WNN based predictive control are tabulated in Table 2, and the control errors are compared in Fig. 5. From the results of Table 2, we can observe that the SRWNN based predictive control shows better performance compared with the WNN based predictive control. Note that the network structure of the proposed SRWNN is simpler than that of the WNN as shown in Table 1.

4.2. The Hénon system

We consider the Hénon system, which is the discrete-time chaotic system. The state equation of the Hénon system is

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} x_2(n)+1-ax_1^2(n) \\ bx_1(n)+u \end{bmatrix}, \tag{43}$$

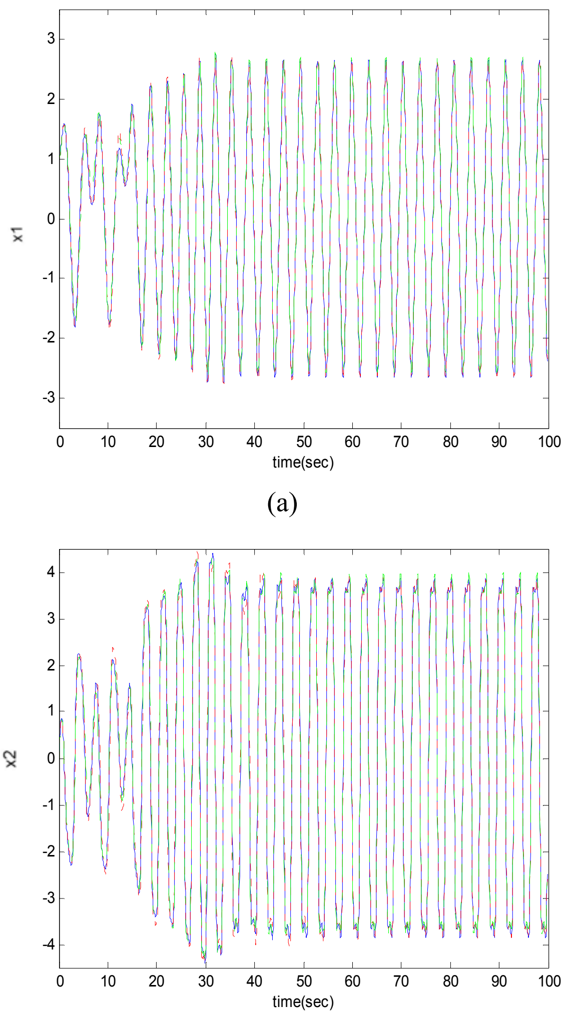
where $a=1.4$, $b=0.3$. The goal for control of the Hénon system is to regulate the chaotic orbit to the desired signal. In this simulation, we define the reference signal as $[0,-1]$ and choose the initial state of the Hénon system as $(1,0)$. The learning rates for the SRWNN based identifier are chosen based on Remark 1 and the learning rate for the control signal is chosen as 0.2, which is based on Theorem 5.

Table 1. Comparison of the simulation environments for the Duffing system.

Simulation Condition	SRWNN	WNN
No. of wavelons	12	20
No. of past inputs	2	2
No. of past states	2	2
No. of prediction steps	3	3
Sampling rate	0.05	0.05
ID learning rate	Adaptive	0.02
Control learning rate	0.006	0.02
Iteration	2000	2000

Table 2. Comparison of the simulation results for the Duffing system.

Performance	SRWNN	WNN
ID MSE (x1)	0.002	0.069
IDMSE (x2)	0.005	0.030
Control MSE (x1)	0.008	0.089
Control MSE (x2)	0.038	0.058



(a)
(b)

Fig. 4. The predictive control results for the Duffing system. (solid line: reference signal, dotted line: SRWNN result, and dash-dotted line: WNN result) (a) state x_1 (b) state x_2.

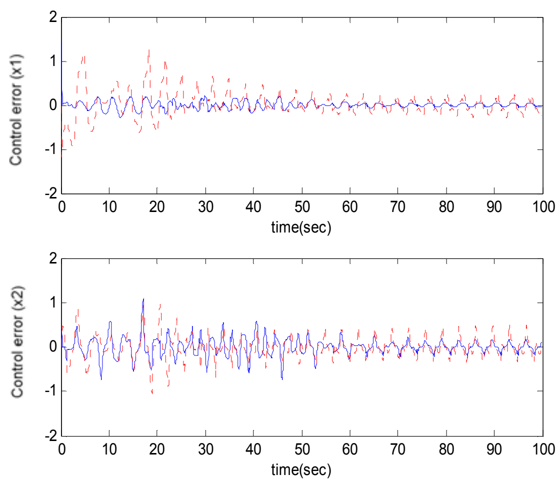


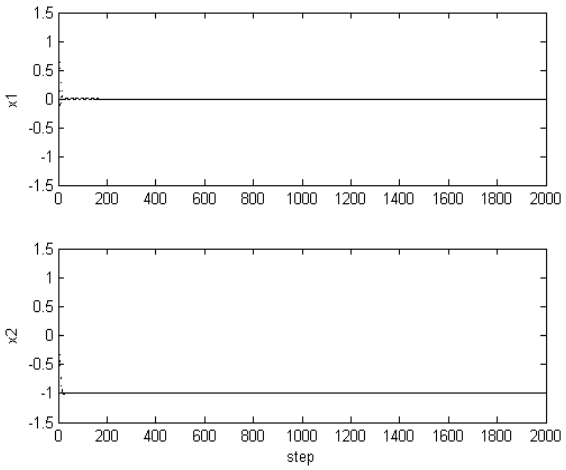
Fig. 5. The control errors of the SRWNN(solid line) and WNN(dotted line).

Table 3. Comparison of the simulation environments for the Hénon system.

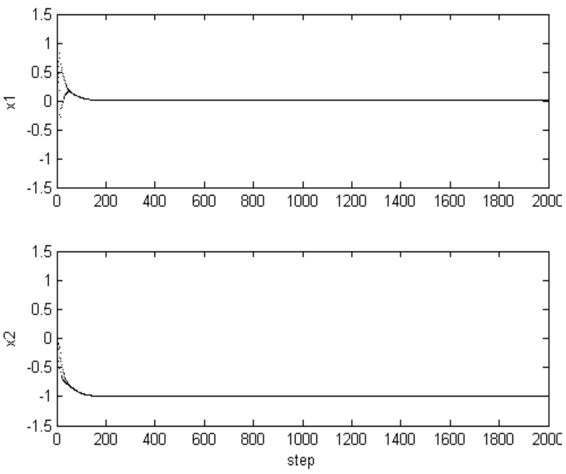
Simulation Condition	SRWNN	WNN
No. of wavelons	9	20
No. of past inputs	1	2
No. of past states	2	2
No. of prediction steps	3	3
Sampling rate	0.05	0.05
ID learning rate	Adaptive	0.1
Control learning rate	0.02	0.2
Iteration	2000	2000

Table 4. Comparison of the simulation results for the Hénon system.

Performance	SRWNN	WNN
ID MSE (x1)	0.0015	0.0024
IDMSE (x2)	0.0005	0.0005
Control MSE (x1)	0.308e-32	0.519e-19
Control MSE (x2)	0.000e-32	0.501e-19



(a)



(b)

Fig. 6. The predictive control results for the Hénon system.

The simulation environments of both the SRWNN based predictive control and the WNN based predictive control are shown in Table 3, and the mean-squared error(MSE)s of the SRWNN based predictive control and the WNN based predictive control are tabulated in Table 4. In this simulation, in order to maximize the performance for the identification and control of the Hénon system, we select experimentally the number of the past input and the past state for the SRWNN and the WNN respectively. In Table 3, the number of the past input of the SRWNN is less than that of the WNN. This means that even if the network structure of the proposed SRWNN is simpler than that of the WNN, the performance of identification and control of the SRWNN is better than that of the WNN (see Table 4). In Table 4, the ID MSEs denotes the MSEs between the output of the Hénon system and the output of SRWNN after the total samples and the control MSEs denotes the MSEs between the output of the Hénon system and the reference signal after 200 samples. Fig. 6 compares the control results using the SRWNN and the WNN. From the results of Fig. 6, we can observe that the SRWNN based controller converges faster than the WNN based controller.

5. CONCLUSIONS

The SRWNN, which is a new network structure, based predictive control method has been proposed for chaotic systems. Since the SRWNN has a good ability to store the past information of the wavelon, it can be used as the model predictor of the predictive control scheme. Using the Lyapunov approach, the convergence theorems for both SRWNN predictor and controller are proven, and the condition of optimal learning rates was also established. Finally, the proposed predictive control scheme was applied to chaotic systems, the Duffing system and the Hénon system. Simulation results have shown that the SRWNN has three advantages. Firstly, the SRWNN has the simpler network structure than the WNN. Secondly, the SRWNN predictor can predict accurately complex chaotic systems. Thirdly, the proposed controller has an on-line adapting ability for controlling complex chaotic systems.

APPENDIX

In order to compute simply the jacobian matrix (34), we define $\mathbf{G}$ as follows:

$$\mathbf{G} = \begin{bmatrix} g_{11} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N1} & g_{N2} & g_{N3} & \cdots & g_{NN} \end{bmatrix}_{N\times N}.$$

Each elements of the Jacobian matrix $\mathbf{G}$ is computed by the following procedure.

A. First row

$$g_{11} = \frac{\partial y_I(n+1)}{\partial u(n)} = \frac{\partial y_I(n+1)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n)}$$

$$g_{12} = \frac{\partial y_I(n+1)}{\partial u(n+1)} = 0$$

$$\vdots$$

$$g_{1N} = \frac{\partial y_I(n+1)}{\partial u(n+N-1)} = 0,$$

where $\mathbf{x} = [y_I(n) \ \cdots \ y_I(n-N_s+1) \ u(n) \cdots \ u(n-N_e)]$ is the inputs of the SRWNN. $N_i = N_s + N_e + 1$ denotes the total number of the inputs. And also, N_s and N_e indicate the number of the past SRWNN outputs and the past control input state variables, respectively.

B. Second row

$$g_{21} = \frac{\partial y_I(n+2)}{\partial u(n)} = \frac{\partial y_I(n+2)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n)}$$

$$g_{22} = \frac{\partial y_I(n+2)}{\partial u(n+1)} = \frac{\partial y_I(n+2)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n+1)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n+1)}$$

$$g_{23} = \frac{\partial y_I(n+2)}{\partial u(n+2)} = 0$$

$$\vdots$$

$$g_{2N} = \frac{\partial y_I(n+2)}{\partial u(n+N-1)} = 0,$$

where $\mathbf{x} = [y_I(n+1) \ \cdots \ y_I(n-N_s+2) \ \ u(n+1) \ \ u(n) \ \ \cdots \ \ u(n-N_e+1)]$.

$$\vdots$$

C. N th row

$$g_{N1} = \frac{\partial y_I(n+N)}{\partial u(n)} = \frac{\partial y_I(n+N)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n)}$$

$$g_{N2} = \frac{\partial y_I(n+N)}{\partial u(n+1)} = \frac{\partial y_I(n+N)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n+1)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n+1)}$$

$$\vdots$$

$$g_{NN} = \frac{\partial y_I(n+N)}{\partial u(n+N-1)} = \frac{\partial y_I(n+N)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial u(n+N-1)} = \sum_{k=1}^{N_i} \left\{ \sum_{j=1}^{N_w} \frac{w_j}{d_{jk}} \frac{\partial \Phi_j(\mathbf{x})}{\partial z_{jk}} + a_k \right\} \frac{\partial x_k}{\partial u(n+N-1)},$$

where $\mathbf{x} = [y_I(n+N-1) \ \cdots \ y_I(n-N_s+N) \ \ u(n+N-1) \ \ \cdots \ \ u(n-N_e+N-1)]$.

REFERENCES

[1] G. Chen and X. Dong, "On feedback control of chaotic nonlinear dynamic systems," *Int. Jour. of Bifucation and Chaos,* vol. 2, no. 2, pp. 407-411, 1992.

[2] G. Chen and X. Dong, "On feedback control of chaotic continuous-time systems," *IEEE Trans. on Circuits ans Systems,* vol. 40, no. 9, pp. 591-601, 1993.

[3] J. M. Joo and J. B. Park, "Control of the differentially flat Lorenz system," *Int. Jour. of Bifucation and Chaos,* vol. 11, no. 7, pp. 1989-1996, 2001.

[4] K. S. Park, J. B. Park, Y. H. Chio, and G. Chen, "Generalized predictive control of discrete-time chaotic system," *Int. Jour. of Bifurcation and Chaos,* vol. 8, no. 7, pp. 137-148, 1987.

[5] A. S. Poznyak, W. Yu, and E. N. Sanchez, "Identification and control of unknown chaotic systems via dynamic neural networks," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications,* vol. 46, no. 12, pp. 1491-1495, 1999.

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

[7] G. Geng and G. M. Geary, "Application of a neural-network-based RLS algorithm in the generalized predictive control of a nonlinear air-handing plant," *IEEE Trans. on Control Systems Technology,* vol. 5, no. 4, pp. 439-445, July 1997.

[8] L. Xiaohua, W. Xiuhong, and W. Yunge, "Generalized predictive control based on error correction using the dynamic neural network," *Proc. of the 3rd World Congress on Intelligent Control and Automation,* vol. 3, no. 28, pp. 1863-1865, July 2000.

[9] 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.

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

[11] L. Cao, Y. Hong, H. Fang, and G. He, "Predicting chaotic time series with wavelets networks," *Physica D,* vol. 85 pp. 225-238, 1995.

[12] B. A. Pearlmutter, "Learning state space trajectories in recurrent neural networks," *Proc. of Int. Joint Conference on Neural Networks,* vol. 2, pp. 365-372, 1989.

[13] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamics systems control," *IEEE Trans. on Neural Networks,* vol. 6, pp. 144-156, January 1995.

[14] 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, March 1990.

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

[16] T. Yabuta and T. Yamada, "Learning control using neural networks," *Proc. of IEEE Int. Conf. Robotics and Automation,* Sacramento, CA, pp. 740-745, 1991.

Sung Jin Yoo received the B.S. degree in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 1999, respectively. Currently, he is pursuing a M.S. degree in the Dept. of Electrical and Electronic Engineering at Yonsei University, Seoul, Korea. His research interests include neural network, adaptive control, predictive control, robotic systems, nonlinear control, 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. Prof. Park is serving as the Director for the Transactions of the KIEE (1998-2004) and the Institute of Control, Automation and Systems Engineers (1999-2004).

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 an Associate Professor. From 2000 to 2002, he was with the Department of Electrical Engineering at The Ohio State University, where he was a Visiting Scholar. Professor Choi is serving as the Director for the Institute of Control, Automation and Systems Engineers (2003-2004). His research interests include intelligent control, mobile robot, web-based control system and wavelet transform.