퍼지 및 지능시스템 학회 2003, Vol. 13, No. 5, pp. 606-613

Radial Basis Function Network Based Predictive Control of Chaotic Nonlinear Systems

Yoon Ho Choi* and Se Min Kim**

* School of Electronic Engineering, Kyonggi University
** LG Innotek Co., Ltd., 148-1, Mabuk-Ri, Gusung-Eup

Abstract

As a technical method for controlling chaotic dynamics, this paper presents a predictive control for chaotic systems based on radial basis function networks(RBFNs). To control the chaotic systems, we employ an on-line identification unit and a nonlinear feedback controller, where the RBFN identifier is based on a suitable NARMA real-time modeling method and the controller is predictive control scheme. In our design method, the identifier and controller are most conveniently implemented using a gradient-descent procedure that represents a generalization of the least mean square(LMS) algorithm. Also, we introduce a projection matrix to determine the control input, which decreases the control performance function very rapidly. And the effectiveness and feasibility of the proposed control method is demonstrated with application to the continuous-time and discrete-time chaotic nonlinear system.

Key Words : Chaos control; Chaotic systems; Predictive control; Radial basis function network; Projection matrix

1. Introduction

The study of chaotic systems over the past few decades has led to many interesting and diverse results, and research works on control and synchronization of chaotic nonlinear systems have been increasing. A nonlinear dynamical system with a chaotic attractor produces motion on the attractor that has random-like properties. Due to its unpredictability and irregularity, chaotic phenomena lead systems to be unstable or performance-degraded situations. As a result, in many cases, chaos is regarded as an undesirable phenomenon to be canceled or controlled.

As the effort for controlling chaos, the OGY method [1] proposed by Ott. et al.,involves stabilizing one of the unstable periodic orbits embedded in the chaotic attractor using small time-dependent perturbations of a system parameter. But this method demands the requirements that system parameters are accessible or can be perturbed from the outside. Later, Chen et al. [2,3] applied some conventional linear feedback control method to the control of chaotic systems. This approach, however, can be applied when we know an exact or at least approximate formula for the desired orbit, which, in turn, requires somewhat complex calculations. Recently, intelligent control approaches using artificial neural networks and fuzzy logics are adopted for chaos control [4,5].

In this paper, we carry out the predictive control using radial basis function networks for the chaotic systems. The predictive control is a well-established control technique for linear systems that can be used to control a wide range of processes [6,7]. However, because the neural network approximating process is a nonlinear model, an analytical controller cannot be obtained without constraints on the controller outputs. Although the RBFN model is highly nonlinear, the predictive control algorithm can easily be implemented in an adaptive mode by using an on-line estimation algorithm such as recursive least mean square(LMS). Thus, the gradient projection method as a kind of LMS algorithm is used to optimize the control performance function iteratively.

As relatively well-known chaotic systems, we consider continuous-time chaotic systems (Duffing system) and discrete-time chaotic systems (Hénon system) to verify the performance of the controller designed by our method.

In Section 2, we describe the architecture of RBF networks and training algorithm for identifying the chaotic nonlinear systems. Section 3 contains predictive control design to obtain the gradient matrix so that the amplitude of cost function is minimized. In Section 4, we present the practical verification as applying to the chaotic nonlinear systems and finally we come to a brief conclusion.

2. Radial Basis Function Networks for Modeling Chaotic Systems

2.1 Radial basis function networks

2.1.1 Architecture

Fig. 1 shows a schematic diagram of an RBFN with four receptive field units; the activation level of the i^{th}

접수일자 : 2003년 3월 27일
완료일자 : 2003년 5월 23일
본 연구는 2001년도 경기대학교 해외 파견 연구비 수혜에 의해 연구되었음.

receptive field unit (or hidden unit) is

$$\phi_i(\mathbf{x}) = G(\|\mathbf{x} - \mathbf{z}_i\| / \sigma_i) \qquad i = 1,2,\cdots,H \tag{1}$$

where $\mathbf{x}$ is a multidimensional input vector, $\mathbf{z}_i$ is a center vector with the same dimension as $\mathbf{x}$, σ_i is the width as standard deviation, $\|\mathbf{x} - \mathbf{z}_i\|$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{z}_i$, H is the number of radial basis functions (or, equivalently, the hidden units), and $\phi_i(\mathbf{x})$ is the i^{th} radial basis function with a single maximum at the i^{th} center vector.

There are no connection weights between the input layer and the hidden layer. Typically $\phi_i(\mathbf{x})$ is a Gaussian function

$$\phi_i(\mathbf{x}) = exp\left(\frac{-\|\mathbf{x} - \mathbf{z}_i\|^2}{2\sigma_i^2}\right) \tag{2}$$

or a logistic function

$$\phi_i(\mathbf{x}) = \frac{1}{1 + exp(\|\mathbf{x} - \mathbf{z}_i\|^2 / \sigma_i^2)} \tag{3}$$

The output of an RBFN can be computed in the simple method. As shown in Fig. 1, the final output is the weighted sum of the output value associated with each hidden unit:

$$y(\mathbf{x}) = \sum_{i=1}^{H} w_i \phi(\mathbf{x}) = \sum_{i=1}^{H} w_i G(\|\mathbf{x} - \mathbf{z}_i\| / \sigma_i) = \mathbf{Gw} \tag{4}$$

where w_i is the connection weight between the i^{th} hidden unit and the output unit.

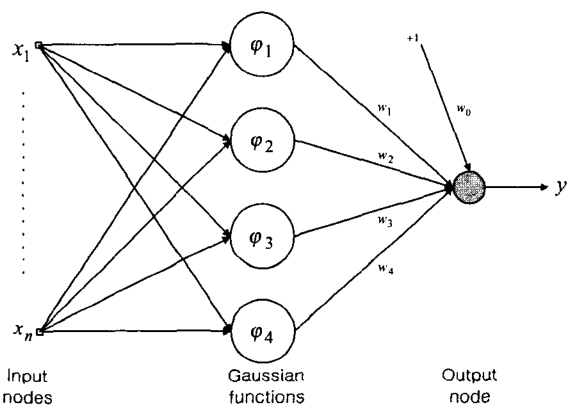

Fig. 1 Structure of RBFN

2.1.2 Recursive hybrid learning procedure

An RBFN's approximation capacity may be further improved with supervised adjustments of the center and shape of the radial basis functions. Several learning algorithms have been proposed to identify the parameters ($\mathbf{z}_i$, σ_i and w_i) of an RBFN. Besides using a supervised learning scheme alone to update all modifiable parameters, a variety of sequential training algorithms for RBFNs have been reported [8-11].

In RBF learning approach, the radial basis functions are permitted to move the locations of their centers in a self-organized fashion for the selection of initial center positions. And then, the centers of the radial basis functions, the linear weights of the output layer and all other free parameters of the network are computed using a supervised learning process. In other words, the network undergoes a hybrid learning algorithm. The self-organized component of the learning process serves to allocate network resources in a meaningful way by placing the centers of the radial basis function in only those regions of the input space where significant data are present. For the self-organized selection of the hidden units' centers, we may use the k-means clustering algorithm [12].

For the supervised learning process, the RBFN takes on its most generalized form. The mathematical basis for the learning algorithm is the optimization technique known as gradient descent.

2.2 Identification of chaotic systems with RBFNs

NARMA (Nonlinear Auto-Regressive Moving Average) is the most general model among the models used to represent a single-input single-output (SISO) nonlinear system and it can be described by the following nonlinear difference equation:

$$\begin{aligned} y(k+1) = f[&y(k), y(k-1), \cdots, y(k-p+1); \\ &u(k), u(k-1), \cdots, u(k-q+1)] \end{aligned} \tag{5}$$

where sequence $[u(k), y(k)]$ represents the input-output pair of the system at time instant k, f is an unknown nonlinear function to be estimated by a neural network, and p and q are the known structure orders of the system [13].

A RBFN is a single-hidden layer feedforward network with linear output transfer functions on the hidden-layer nodes. In other words, RBFN describes the mapping $\hat{f}$ in Eqn. (6) in terms of a weighted summation of a set of radial basis functions. And $\hat{f}$ is the estimate of f [14].

$$\begin{aligned} &\hat{f}(\mathbf{x}) = \sum_{i=1}^{H} w_i \phi_i(\mathbf{x}) + w_0 \\ &\phi_i(\mathbf{x}) = G(\|\mathbf{x} - \mathbf{z}_i\| / \sigma_i), i = 1,2,\cdots,H \end{aligned} \tag{6}$$

where the function $\phi_i(\mathbf{x})$ is called a radial basis function and w_i, w_0 are the weights matrix and bias of the neural network respectively. For the Gaussian RBF, $\phi_i(\mathbf{x})$ is defined by

$$\phi_i(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{z}_i\|^2}{2\sigma_i^2}\right) \tag{7}$$

And the standard deviation (i.e., width) of all the Gaussian RBFs is fixed at

$$\sigma = \frac{d}{\sqrt{2H}}$$

where H is the number of centers and d is the maximum distance between the chosen

Since the input to the neural network is

$$\mathbf{x} = [y(k), y(k-1), \cdots, y(k-p+1); \\ u(k), u(k-1), \cdots, u(k-q+1)] \tag{8}$$

the neural network model for the unknown system (5) can be expressed as

$$\hat{y}(k+1) = \hat{f}\,[y(k), y(k-1), \cdots, y(k-p+1); \\ u(k), u(k-1), \cdots, u(k-q+1)] \tag{9}$$

where $\hat{y}(k+1)$ is the output of the neural network.

In the advance of system identification, we firstly initialize centers by the k-means algorithm to improve control performance.

The learning strategy for system identification is the selection of weights and centers using a supervised learning process. A natural candidate for such a process is error-correction learning, which is most conveniently implemented using a gradient- descent procedure that represents a generalization of the LMS algorithm.

The first step in the development of such a learning procedure is to define the instantaneous value of the modeling performance function:

$$\delta = \frac{1}{2} e_I^2(k+1) \tag{10}$$

where $e_I(k+1)$ is the error signal, defined by

$$e_I(k+1) = y(k+1) - \hat{y}(k+1) \tag{11}$$

where $y(k+1)$ is the actual output signal.

The requirement is to find the free parameters w_i, $\mathbf{z}_i$ so as to minimize δ. The results of this minimization are summarized in Eqns. (12), (13). The following two equations describe adaptation formulas for the linear weights and positions of centers for RBFNs.

1) Linear weights (output layer)

$$w_i(k+1) = w_i(k) - \eta_1 \frac{\partial\delta}{\partial w_i(k)}, \; i = 1, 2, \cdots, H$$

$$\frac{\partial\delta}{\partial w_i(k)} = -e_I(k+1) G(\ \mathbf{x} - \mathbf{z}_i\ /\sigma) \tag{12}$$

2) Positions of centers (hidden layer)

$$\mathbf{z}_i(k+1) = \mathbf{z}_i(k) - \eta_2 \frac{\partial\delta}{\partial \mathbf{z}_i(k)}, \; i = 1, 2, \cdots, H$$

$$\frac{\partial\delta}{\partial \mathbf{z}_i(k)} = -\frac{2}{\sigma^2} w_i(k) e_I(k+1) G(\ \mathbf{x} - \mathbf{z}_i\ /\sigma)(\mathbf{x} - \mathbf{z}_i) \tag{13}$$

3. Design of Predictive Controller for Chaotic Systems

In this Section, we implement multi-step predictor with the estimated output $\hat{y}$ of RBFN that is a model of chaotic systems, and then design the predictive controller for chaotic systems.

3.1 Chaotic systems

In this paper, we consider the Duffing and Hénon systems, which are two representative continuous-time and discrete-time chaotic nonlinear systems, respectively.

The solution to the Duffing equations is often used as an example of a classic chaotic system. The state equation of the Duffing system is

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} y \\ a_1 x - x^3 - a_2 y + b\cos(wt) + u \end{pmatrix} \tag{14}$$

where typically, $a_1 = 1.1$, $a_2 = 0.4$, $b = 2.1$ and $w = 1.8$.

And, we discuss another two-dimensional map with a strange attractor. Hénon chose to study mappings rather than differential equations because maps are faster to simulate and their solutions can be followed more accurately and for a longer time.

The state space of the Hénon system with x and y as state variables is expressed by

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} y_n + 1 - a x_n^2 \\ b x_n + u \end{pmatrix} \tag{15}$$

where, $a = 1.4$, $b = 0.3$.

3.2 Predictive control algorithm

Since the given systems are uncertain, we assume that the closed-loop system output data are available on-line for the controller. We take the approach that employs an on-line system identification unit and a nonlinear feedback controller, where the RBFN identifier is based on a suitable NARMA real-time modeling method and the controller is a predictive controls cheme. The overall configuration of the closed-loop control system is shown in Fig. 2, where the output $y(k)$ is to be controlled to track the reference, $r(k)$.

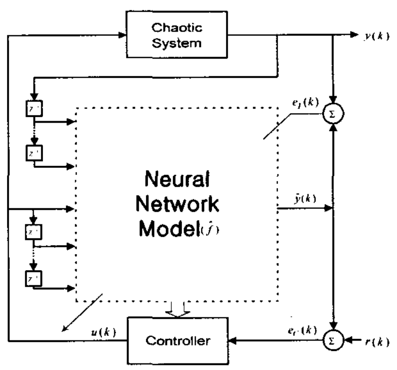

Fig. 2 Block diagram of control system

Our purpose is to select optimal control signal u in order to minimize the control performance function:

$$\epsilon = \frac{1}{2} e_C^2(k+1) \tag{16}$$

where $e_C(k+1)$ is the control error signal, defined by

$$e_C(k+1) = r(k+1) - \hat{y}(k+1) \tag{17}$$

and $r(k+1)$ is the desired output signal.

Using the neural network structure shown in Fig. 1, Eqn. (9) can be rewritten to give

$$\hat{y}(k+1) = \sum_{i=1}^{H} w_i G(\mathbf{x} - \mathbf{z}_i /\sigma) + w_0 \tag{18}$$

To minimize ϵ, $u(k)$ is recursively calculated via a simple gradient-descent method.

$$u(k+1) = u(k) - \eta \frac{\partial \epsilon}{\partial u(k)} \tag{19}$$

where $\eta > 0$ is a learning rate. It can be seen that the controller relies on the approximation performed by the neural network. Therefore, it is necessary that $\hat{y}(k+1)$ approaches the real system output $y(k+1)$ asymptotically. This can be achieved by keeping the neural network training on-line.

Substituting Eqn. (17) into Eqn. (16) and then differentiating the result with respect to $u(k)$, we have

$$\frac{\partial \epsilon}{\partial u(k)} = - e_C(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)} \tag{20}$$

where $\partial \hat{y}(k+1)/\partial u(k)$ that is known as the gradient of the neural network model with respect to $u(k)$ can be analytically evaluated by using the known neural network structure, Eqn. (18) as follows :

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = - \frac{2}{\sigma^2} \sum_{i=1}^{H} \left(w_i G(\mathbf{x} - \mathbf{z}_i /\sigma)(\mathbf{x} - \mathbf{z}_i) \frac{\partial \mathbf{x}}{\partial u(k)} \right) \tag{21}$$

where, $\frac{\partial \mathbf{x}}{\partial u(k)} = [\overleftarrow{0, 0, \cdots, 0,}^{p} \; \overleftarrow{1, 0, \cdots, 0}^{q}]'$,

and $'$ denotes "transpose". Finally, Eqn. (19) becomes

$$u(k+1) = u(k) - \frac{2}{\sigma^2} \eta e_C(k+1) \sum_{i=1}^{H} \left(w_i G(\mathbf{x} - \mathbf{z}_i /\sigma)(\mathbf{x} - \mathbf{z}_i) \frac{\partial \mathbf{x}}{\partial u(k)} \right) \tag{22}$$

So far, we described the algorithm for one-step ahead predictive control scheme. Then let the algorithm described above be extended to a multi-step ahead predictive control scheme, which considers not only the design of the instant value of the control signal but also its future values. In general, predictive control strategy consists of the following four parts:

1) At each sampling time, the estimated value of neural network model $\hat{y}(k+i)$ is predicted over the prediction horizon $i = 1, \cdots, N_p$. This prediction depends on the future values of the control signal $u(k+i)$ within a control horizon $i = 1, \cdots, N_u$, where $N_u \le N_p$.

2) A reference trajectory $r(k+i)$, $i = 1, \cdots, N_p$ is defined which describes the desired process trajectory over prediction horizon.

3) The vector of future controls is computed such that a cost function depending on the predicted control errors is minimized. The first element of the control vector is applied to the process.

4) The prediction error between the measured process output and the predicted output is used for the systems identification.

Steps 1) - 4) are repeated at each sampling instant; this strategy is known as a receding horizon control.

As a result, future values of setpoint and the system output are needed to formulate the control signal. Since the neural network model represents the system to be controlled asymptotically, it can be used to predict future values of the system output. For this purpose, let N be prediction horizon ($N_p = N$) and $N_u = N_p$ be simply selected in the paper. Then, we can denote the following vectors:

$$\begin{aligned} R_{N|k} &= [r_{k+1}, r_{k+2}, \cdots, r_{k+N}]' \\ \hat{Y}_{N|k} &= [\hat{y}_{k+1}, \hat{y}_{k+2}, \cdots, \hat{y}_{k+N}]' \\ E_{N|k} &= [e_{k+1}, e_{k+2}, \cdots, e_{k+N}]' \end{aligned} \tag{23}$$

as the future values of the setpoint, the predicted output of neural network model and the error vector between two vectors, respectively.

Define the control sequence as

$$U_{N|k} = [u_k, u_{k+1}, \cdots, u_{k+N-1}]' \tag{24}$$

and consider the following control performance function:

$$J = \frac{1}{2} [E'_{N|k} E_{N|k}] \tag{25}$$

Then the control purpose is to find $U_{N|k}$ such that J is minimized. Using the gradient projection method, the control sequence U is updated at each iteration as follows:

$$U_{N|k+1} = U_{N|k} - \eta D_{N|k} \tag{26}$$

where, $D_{N|k}$ denotes the search direction at the present time instant k. Also $D_{N|k}$ is determined in the sense that the negative of the gradient projected on the constraints gives the direction in which the control performance function decreases most rapidly [15]. The search direction at time k is given by

$$D_{N|k} = - E_{N|k} P_{N|k} \frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}} = - E_{N|k} P_{N|k} \Gamma_{N|k} \tag{27}$$

where $P_{N|k}$, called the projection matrix, is a $N \times N$ diagonal matrix initiated to unity $P_{N|k_0} = I$ and

$$\Gamma_{N|k} = \begin{bmatrix} \frac{\partial \hat{y}_{k+1}}{\partial u_k} & 0 & \cdots & 0 \\ \frac{\partial \hat{y}_{k+2}}{\partial u_k} & \frac{\partial \hat{y}_{k+2}}{\partial u_{k+1}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}_{k+N}}{\partial u_k} & \frac{\partial \hat{y}_{k+N}}{\partial u_{k+1}} & \cdots & \frac{\partial \hat{y}_{k+N}}{\partial u_{k+N-1}} \end{bmatrix} \tag{28}$$

$\Gamma_{N|k}$ is the gradient of the control performance function with respect to $U_{N|k}$, which can be derived from the RBFN model and can be easily evaluated. we describe the computing procedure for $\Gamma_{N|k}$ in the Appendix A.

The individual element of the control sequence is updated by clipping the results obtained from Eqn. (26) according to

$$\underline{\Delta u} \le \Delta u(k+i-1) \le \overline{\Delta u} \tag{29}$$

where $\Delta u(k+i-1) = u(k+i-1) - u(k+i-2)$ and each $\underline{\Delta u}$ and $\overline{\Delta u}$ can be arbitrarily chosen as very small value.

The projection matrix P is then updated according to $U_{N|k}$ at each iteration.

$$P_{N|k}(i,i) = \begin{cases} 0 & \text{if } \underline{\Delta u} \le \Delta u(k+i-1) \le \overline{\Delta u} \\ P_{N|k-1}(i,i) & \text{otherwise} \end{cases}$$

$$i = 1, 2, \cdots, N \tag{30}$$

After finding the new control sequence, the first element of $U_{N|k+1}$ is applied to the system as the control signal.

4. Simulation Results

In this Section, we present some simulation results to validate the proposed predictive control scheme for both continuous-time and discrete-time chaotic systems. And, in order to evaluate the performance of the proposed controller, we compares the results of a RBFN based predictive controller with those of a NN based predictive controller.

4.1 Controlling the Duffing system

In tracking Duffing system, we define the initial system state as (0, 0) and the learning rate is chosen as 0.001. Reference signal is solution in case that parameter b, of Duffing equation is 2.3. Also, in on-line learning procedure, the number of hidden nodes is 10 and to avoid complex model structure order, we make each of feedback input/output signal have only one. The prediction horizon is selected as 3 and the sampling time is defined as 0.05. And, NN model has two past outputs of the plant, one current output and one past output of controller as the inputs. The hidden layer of the NN model has five nodes.

Note that we can obtain the system identification error and control performance error from simulation results (see Figs. 3-6). The mean-squared error (MSE) for system identification and control performance indicates in Table 1.

From the results obtained above, we can see that although identification error of RBFN model are more than those of NN model, RBFN based predictive controller shows a better final control performance, and it is faster and more effective, as compared with the NN based predictive control.

Table 1 Mean square errors of Duffing system

	RBFN		NN	
	ID Error (MSE)	Control Error (MSE)	ID Error (MSE)	Control Error (MSE)
State x	0.0328	0.0158	0.0108	3.7029
State y	0.1138	0.0712	0.0113	11.4418

4.2 Controlling the Hénon system

In this subsection, simulation results of the proposed predictive control scheme for the discrete-time chaotic systems are presented. In this simulation, the control objective is to regulate the chaotic orbit to the desired point. As like continuous-time chaotic systems, the number of hidden nodes is 10 and learning rate is chosen as 0.01 and to avoid complex model structure order, we make each of feedback input/output signal have only one. The prediction horizon is selected as 3 and the system initial state starts at (0, 0). And, NN model for Hénon system has four inputs; two past outputs of Hénon system, one current and one past control signal of the controller. Also, the node number of the hidden layer for the NN is 10.

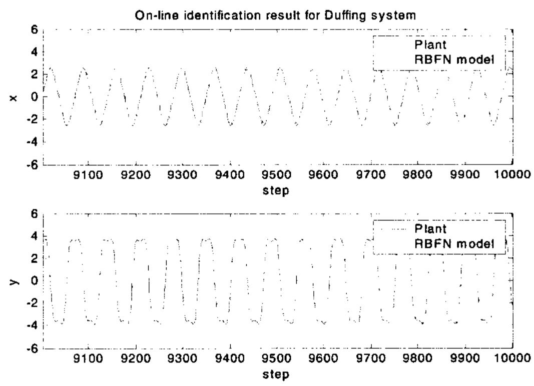

Fig. 3 Estimated output of RBFN model vs. output of system

Figs. 7 and 8 show the output of the system and the output of RBFN model approximating Hénon system, respectively. Also, Figs. 9 and 10 show the regulation control results for Hénon system and the output of NN model, respectively. Table 2 describes the error between the desired output and the system output and the error between the system output and the estimated model output, where the reference signal $\mathbf{r}$ is [0, -1].

While the controlled chaotic signal based on the NN converges to the desired point at about 800 steps, the controlled chaotic signal based on RBFN converges to the desired point at about 200 steps. From the results obtained

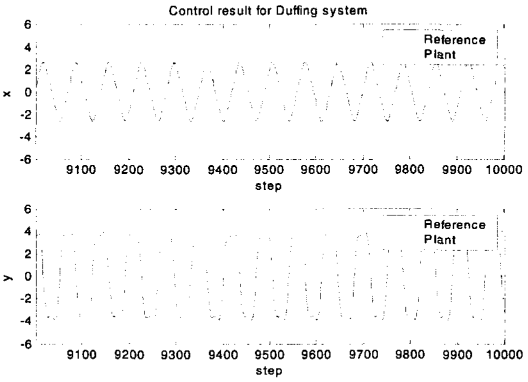

Fig. 4 State output of system vs. reference signal (RBFN)

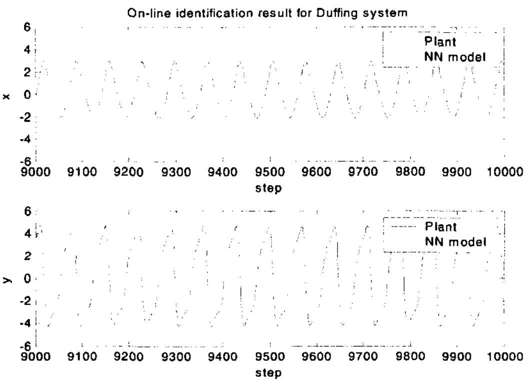

Fig. 5 Estimated output of NN model vs. output of system

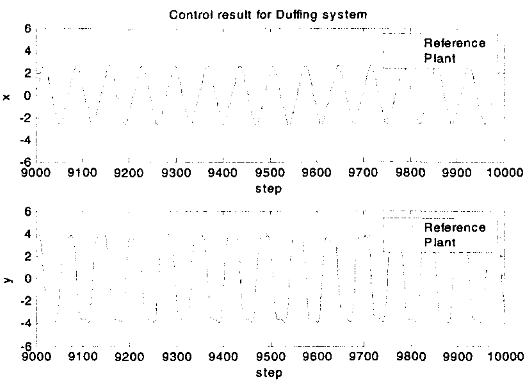

Fig. 6 State output of system vs. reference signal (NN)

above, we can see that a RBFN based predictive controller shows a better control performance as compared with a NN based predictive controller.

Table 2 Regulation errors of Hénon system

	RBFN		NN	
	System ID Error	Control Error	System ID Error	Control Error
State x	-0.0002594	-0.0009252	-0.0458	-0.0734
State y	-0.0004414	-0.0009233	-0.0082	-0.1997

5. Conclusions

In this paper, we have presented the predictive control of the chaotic systems with RBFN model, which was used to perform the multi-step prediction on-line. To control the chaotic systems, we employed an on-line identification unit and a nonlinear feedback controller, where the RBFN identifier was based on a suitable NARMA real-time modeling method and the controller was based on predictive control scheme. In our design method, the identifier and controller were implemented using a gradient-descent procedure that represents a generalization of the least mean square(LMS) algorithm. Also, we have introduced a projection matrix to determine the control input, which decreased the control performance function very rapidly. And the effectiveness and feasibility of the proposed control method have been demonstrated with application to the Duffing system and Hénon system, which are two representative continuous-time and discrete-time chaotic systems. From the simulation results, we have shown that the RBFN based predictive controller was faster convergence property and more accurate control performance than the conventional NN based controller. Also, we have verified that the proposed predictive control scheme worked well for various chaotic systems.

Appendix A

We know that the predictive control scheme is applied to RBF networks for improvement of system identification. Then, the work for finding the important parameter Γ_{Mk} is needed in control design. We can find the Jacobian matrix Γ_{Mk} by differentiating $\hat{Y}_{Mk}$ with respect to U_{Mk}. The following illustrates procedure for computing the elements in the Jacobian matrix for $N=3$.

$$(1,1)\quad g_{11}=\frac{\partial \hat{y}_{k+1}}{\partial u_k}=-\frac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M$$

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_k}=\begin{matrix}\hat{y}_k-z_{1i} \\ \hat{y}_{k-1}-z_{2i} \\ u_k-z_{3i} \\ u_{k-1}-z_{4i}\end{matrix}\quad\begin{matrix}0\\0\\1\\0\end{matrix}$$

(1,2) $g_{12}=\dfrac{\partial \hat{y}_{k+1}}{\partial u_{k+1}}=0$

(1,3) $g_{13}=\dfrac{\partial \hat{y}_{k+1}}{\partial u_{k+2}}=0$

where, $\mathbf{x}=[\hat{y}_k,\ \hat{y}_{k-1},\ u_k,\ u_{k-1}]$.

(2,1) $g_{21}=\dfrac{\partial \hat{y}_{k+2}}{\partial u_k}=-\dfrac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M$

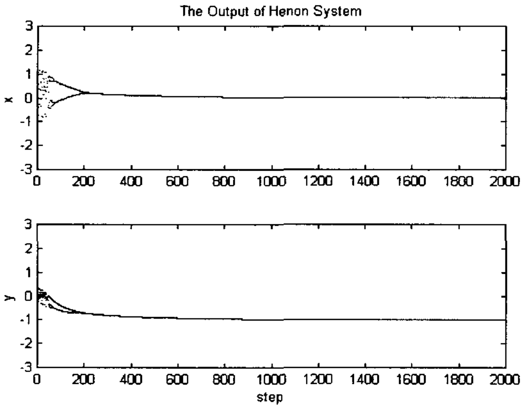

Fig. 7 State output of Hénon system (RBFN)

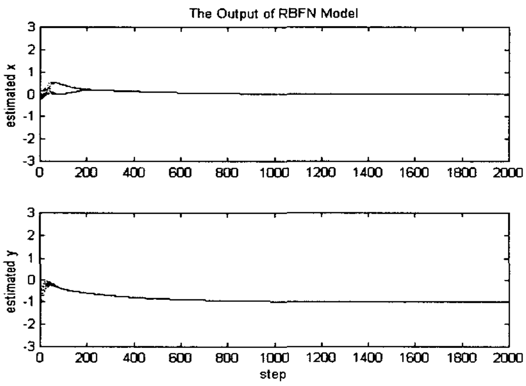

Fig. 8 Estimated output of RBFN model

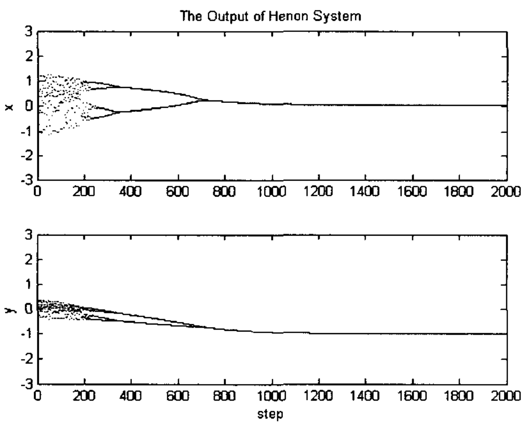

Fig. 9 State output of Hénon system (NN)

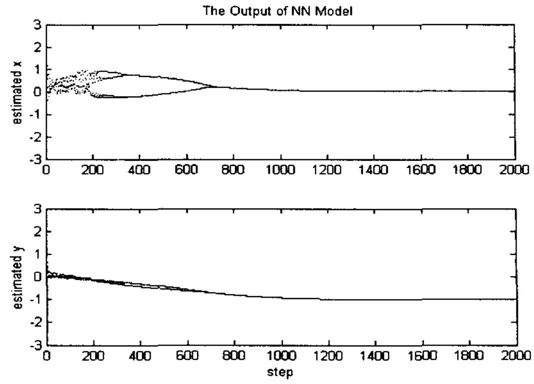

Fig. 10 Estimated output of NN model

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_k}=\begin{matrix}\hat{y}_{k+1}-z_{1i}\\ \hat{y}_k-z_{2i}\\ u_{k+1}-z_{3i}\\ u_k-z_{4i}\end{matrix}\quad\begin{matrix}g_{11}\\0\\0\\1\end{matrix}$$

(2,2) $g_{22}=\dfrac{\partial \hat{y}_{k+2}}{\partial u_{k+1}}=-\dfrac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M\neq g_{11}$

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_{k+1}}=\begin{matrix}\hat{y}_{k+1}-z_{1i}\\ \hat{y}_k-z_{2i}\\ u_{k+1}-z_{3i}\\ u_k-z_{4i}\end{matrix}\quad\begin{matrix}0\\0\\1\\0\end{matrix}$$

(2,3) $g_{23}=\dfrac{\partial \hat{y}_{k+2}}{\partial u_{k+2}}=0$

where, $\mathbf{x}=[\hat{y}_{k+1},\ \hat{y}_k,\ u_{k+1},\ u_k]$.

(3,1) $g_{31}=\dfrac{\partial \hat{y}_{k+3}}{\partial u_k}=-\dfrac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M$

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_k}=\begin{matrix}\hat{y}_{k+2}-z_{1i}\\ \hat{y}_{k+1}-z_{2i}\\ u_{k+2}-z_{3i}\\ u_{k+1}-z_{4i}\end{matrix}\quad\begin{matrix}g_{21}\\g_{11}\\0\\0\end{matrix}$$

(3,2) $g_{32}=\dfrac{\partial \hat{y}_{k+3}}{\partial u_{k+1}}=-\dfrac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M\neq g_{21}$

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_{k+1}}=\begin{matrix}\hat{y}_{k+2}-z_{1i}\\ \hat{y}_{k+1}-z_{2i}\\ u_{k+2}-z_{3i}\\ u_{k+1}-z_{4i}\end{matrix}\quad\begin{matrix}g_{11}\\0\\0\\1\end{matrix}$$

(3,3) $g_{33}=\dfrac{\partial \hat{y}_{k+3}}{\partial u_{k+2}}=-\dfrac{2}{\sigma^2}\sum_{i=1}^{H} w_i G(\mathbf{x}-\mathbf{z}_i/\sigma)M\neq g_{11}$

$$M=(\mathbf{x}-\mathbf{z}_i)\frac{\partial \mathbf{x}}{\partial u_{k+2}}=\begin{matrix}\hat{y}_{k+2}-z_{1i}\\ \hat{y}_{k+1}-z_{2i}\\ u_{k+2}-z_{3i}\\ u_{k+1}-z_{4i}\end{matrix}\quad\begin{matrix}0\\0\\1\\0\end{matrix}$$

where, $\mathbf{x}=[\hat{y}_{k+2},\ \hat{y}_{k+1},\ u_{k+2},\ u_{k+1}]$.

References

[1] E. Ott, C. Grebogi and J. A. York, "Controlling Chaos," Physical Review Letters, Vol. 64, No. 11, pp. 1196-1199, 12 March 1990.

[2] G. Chen and X. Dong, "On Feedback Control of Chaotic Continuous-Time Systems," IEEE Trans. on Circuits and Systems, Vol. 40, No. 9, pp. 591-601, September 1993.
[3] G. Chen, "Optimal Control of Chaotic Systems," Int'l Jour. Bifurcation and Chaos, Vol. 4, No. 2, pp. 461-463, 1994.
[4] G. Chen, "Intelligent Identification and Control of Chaotic Dynamics," Proc. of IEEE Symp. Circuits and Systems, pp. 5-8, 1996.
[5] K. B. Kim, J. B. Park, Y. H. Choi and G. Chen, "Control of Chaotic Nonlinear Systems Using Radial Basis Function Network Approximators," Information Sciences, Vol. 130, pp. 165-183, 2000.
[6] K. S. Park, J. B. Park, Y. H. Choi, G. Chen, "Generalized Predictive Control of Discrete-Time Chaotic System," Int'l Jour. of Bifurcation and Chaos, Vol. 8, No. 7, 1998.
[7] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control, Part I and II, The Basic Algorithm," Automatica, Vol. 23, No. 2, pp. 137-148, 1987.
[8] S. Chen, C. F. N. Cowan and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," IEEE Trans. on Neural Networks, Vol. 2, No. 2, pp. 302-309, March 1991.
[9] S. M. Botros and C. G. Atkeson, "Generalization Properties of Radial Basis Functions," Advances in Neural Information Processing Systems III, pp. 707-713, Morgan Kaufmann, 1991.
[10] V. Kadirkamanathan, M. Niranjan and F. Fallside, "Sequential Adaptation of Radial Basis Function Neural Networks," Advances in Neural Information Processing Systems III, pp. 721-727, Morgan Kaufmann, 1991.
[11] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris and D. M. Hummels, "On the Training of Radial Basis Function Classifiers," Neural Networks, Vol. 5, pp. 595-603, 1992.
[12] A. S. Pandya and R. B. Macy, Pattern Recognition with Neural Networks in C++, CRC Press, 1996.
[13] K. S. Narendra and S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models," IEEE Trans. on Neural Networks, Vol. 8, pp. 475-485, 1997.
[14] K. V. Ha, "Hierarchical Radial Basis Function Networks" Proc. of IEEE International Joint Conf. on Neural Networks, Vol. 3, pp. 1893-1898, 1998.
[15] J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints," SIAM J. Applied Mathematics, Vol. 8, pp. 181-217, 1960.

저 자 소 개

Yoon Ho Choi (최윤호)
received his B.S., M.S. and Ph.D. degree in Electrical Eng. 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 2002 to 2002, he was with the Dept. of Electrical Engineering at The Ohio State University, where he was a visiting scholar. Prof. Choi is serving as a Director for the Institute of Control, Automation and System Engineers(2003-present). His research interests include intelligent control, mobile robot, web based remote control system, wavelet transform and nonlinear robust control.

Phone : +82-31-249-9801
Fax : +82-31-249-9796
E-mail : yhchoi@kuic.kyonggi.ac.kr

Se Min Kim (김세민)
received his B.S. and M.S. degree in Electrical & Electronic Engineering from Yonsei University, Seoul, Korea, in 1998 and 2000, respectively. He is currently in LG Innotek Co., Ltd., Kyonggi, Korea. His research interests include intelligent control and adaptive control.

Phone : +82-31-288-9582
Fax : +82-31-285-0199
E-mail : smkimj@lginnotek.com