혼돈 비선형 시스템의 퍼지 신경 회로망 기반 일반형 예측 제어

論 文
53D-2-1

Fuzzy Neural Network Based Generalized Predictive Control of Chaotic Nonlinear Systems

崔 鍾 泰* · 崔 允 浩**
(Jong Tae Choi · Yoon Ho Choi)

Abstract - This paper presents a generalized predictive control method based on a fuzzy neural network(FNN) model, which uses the on-line multi-step prediction, for the intelligent control of chaotic nonlinear systems whose mathematical models are unknown. In our design method, the parameters of both predictor and controller are tuned by a simple gradient descent scheme, and the weight parameters of FNN are determined adaptively during the operation of the system. In order to design a generalized predictive controller effectively, this paper describes computing procedure for each of the two important parameters. Also, we introduce a projection matrix to determine the control input, which decreases the control performance function very rapidly. Finally, in order to evaluate the performance of our controller, the proposed method is applied to the Duffing and Hénon systems, which are two representative continuous-time and discrete-time chaotic nonlinear systems, respectively.

Key Words : Chaos control, Generalized predictive control, Prediction, Fuzzy neural networks, Projection

1. Introduction

t has long realized that the responses of many nonlinear dynamical systems do not follow simple, regular, and predictable trajectories, but swirl around in a random-line and seemingly irregular, yet well-defined, fashion. As long as the process involved is nonlinear, even a simple strictly deterministic model may develop such complex behavior. This behavior has been understood and accepted as chaos and has led to dramatic developments in the nonlinear sciences and dynamical systems engineering.

n the last few decades, chaos has received increasing attention in various areas such as mathematics, engineering, physics, biology, economics, etc., since chaos may actually useful under certain circumstances [1-3]. Due to its unpredictability and irregularity, chaos can lead systems to troubled, unstable, or even catastrophic situations. Therefore, in many cases, chaos is considered as an undesirable phenomenon to be avoided or controlled [4].

The concept of chaos control, similar to conventional systems control, has come to mean stabilization of unstable periodic orbits of a dynamical system. In the pursuit of chaos control, many methods and techniques discussed in the last few years work remarkably well up to a point. However, there are still many problems that need to be considered. For instance, the parametric variation control method, which is proposed by Davies and Rangavajhula [5]. is out of the question when none of the system parameters is accessible; the engineering feedback control [6], open-loop entrainment-migration methods, and conventional control methods [7-11] such as optimal control, adaptive control, and robust control may not be feasible when an explicit, faithful mathematical model cannot be constructed due to the extreme complexity of the physical system to be controlled. Various difficulties encountered in handling chaotic systems have posed a real need for using some kind of intelligent mechanism that does not rely on accurate mathematical models or accessible parameters of physical systems [12-14].

Generally speaking, expert systems, neural network, fuzzy logic technology, statistical methods, Petri-nets, genetic algorithm, *etc.*, belong to the most widely recognized class of artificial intelligence technologies. Neural networks are mathematical models developed in an attempt to emulate human neural systems. Conventional neural networks have good ability of self-learning but have some limitations such as slow convergence, difficulty in reaching global minima in the parameter space, and sometimes instability as well [15]. On the other hand, fuzzy logic technology is a human-imitating logic, but is lack of ability of self-learning and self-tuning [16]. Therefore, in the research area of intelligent control, FNNs (Fuzzy Neural Networks) are devised to overcome the limitations and to combine the advantages of both neural networks and fuzzy logic [17,18]. This provides a strong motivation for using FNNs for chaos control.

This paper proposes a design method of a generalized predictive controller [19,20] for chaotic systems whose precise mathematical models are not available. In our method, FNN is used as the predictor whose parameters are tuned by the error between the actual output of chaotic system and that of its FNN model. A generalized predictive controller for FNN model is developed in such a way that its

* 正 會 員 : 延世大 工大 電氣電子工學科 博士課程
** 正 會 員 : 京畿大 電子工學部 副教授 · 工博
接受日字 : 2003年 4月 9日
最終完了 : 2004年 1月 3日

parameters are adjusted by using the gradient descent scheme, where the difference between the actual output and the reference signal is used as control input. Also, we have introduced a projection matrix to determine the control input, which decreases the control performance function very rapidly [21]. Finally, in order to evaluate the performance of our controller, we apply the proposed method to the continuous-time and discrete-time chaotic nonlinear systems.

This paper is organized as follows. Section 2 describes the structure and learning method of FNNs. In Section 3, the structure of a generalized predictive control system based on FNN model and the design method of the controller are presented. For illustrative purposes, simulation results for two representative chaotic systems (the Duffing and Hénon systems) are presented in Section 4. Finally, conclusions are drawn in Section 5.

Throughout this paper, we will use the following generic notations: lower case symbols such as x, y, α, w, refer to scalar valued objects, lower case boldface symbols such as $\mathbf{x}, \mathbf{y}$, refer to vector valued objects, and finally capital symbols will be used as matrices.

2. Fuzzy Neural Networks

Fuzzy logic has a distinguished feature that it can describe complex nonlinear systems linguistically [22,23]. However, it is very difficult to identify the fuzzy rules and tune the membership functions of the fuzzy reasoning mechanism. Neural networks, on the other hand, utilize their learning capability for automatic identification and tuning, but they have the following problems [24-27]: (i) they need accurate input-output data; (ii) their learning process is time-consuming, to mention a few.

The basic idea of using FNN is to realize the process of fuzzy reasoning by the structure of a neural network and to make the parameters of fuzzy reasoning be expressed by the connection weights of a neural network. FNNs can automatically identify the fuzzy rules by modifying the connection weights of the networks using the gradient descent scheme. Among various fuzzy inference methods, FNNs use the sum-product composition. The functions that are implemented by the networks must be differentiable in order to apply the gradient descent scheme to their learning.

2.1 Structure of fuzzy neural networks

Fig. 1 shows the configuration of FNN, which has two inputs (x_1, x_2), one output (y), and three membership functions in each premise. The circles and the squares in the figure represent the units of the network. The denotations w_c, w_g, w_b, and the numbers 1, -1 between the units denote connection weights of the network.

FNN can be divided into two parts according to the process of the fuzzy reasoning: the premise part and the consequence parts. The premise part consists of layers (A) through (E) and the consequence parts consist of layers (E) through (F). The grades of the membership functions in the premise are calculated in layers (A) through (D). The connection weights w_c and w_g are the parameters used to determine

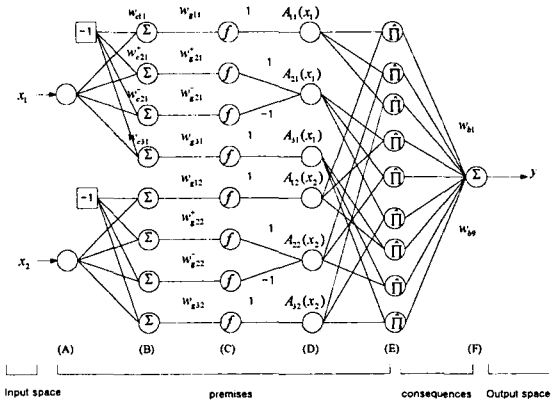

Fig. 1 Configuration of FNN

the central position and the gradient of the sigmoid function in the units of layer (C), respectively. The output of unit in layer (C) is given by:

$$f = \frac{1}{1 + exp\{-w_g(x_j - w_c)\}} \tag{1}$$

where x_j is the j-th input. Thus, by initializing the connection weights appropriately, the membership functions A_1, A_{2j} and A_{3j} can be allocated to the universe of discourse, as shown in Fig. 2.

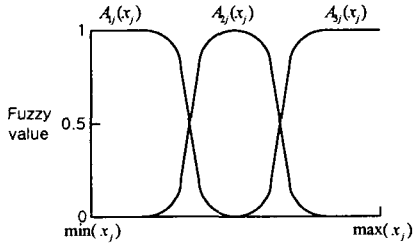

Fig. 2 Membership function in premise

The pseudo-trapezoidal membership function A_2 is composed by using two sigmoid functions, as illustrated in Fig. 3.

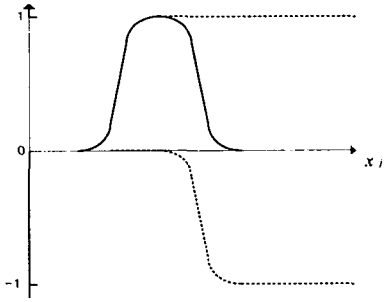

Fig. 3 Composition of membership function A_2

The truth values of the fuzzy rules are obtained as the outputs of the units in layer (E). In case of Fig. 1, the input space is divided into nine fuzzy subspaces, as shown in Fig. 4.

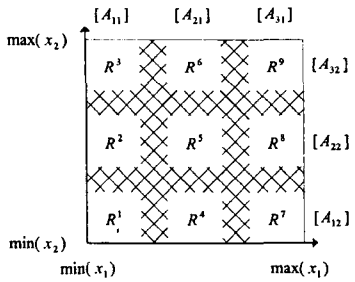

Fig. 4 Fuzzy subspace in input space

The truth value of the fuzzy rule in each subspace is given by the product of the grades of the membership functions for the units in layer (E), as follows:

$$\text{Inputs of layer (E)} : \mu_i = \prod_j A_{i,j}(x_j) \tag{2}$$

and

$$\text{Outputs of layre (E)} : \hat{\mu}_i = \frac{\mu_i}{\sum_{k=1}^{n} \mu_k} \tag{3}$$

where μ_i is the truth value of the ith fuzzy rule, $\hat{\mu}_i$ is the normalized value of μ_i and n is the number of fuzzy rules. The subscript j in Eqn. (2) varies, as shown in Fig. 4. In the subspace of R^1, for example, Eqn. (2) is written as $\mu_1 = A_{11}(x_1)A_{12}(x_2)$. FNN realizes the center of gravity defuzzification formula using $\hat{\mu}_i$ in Eqn. (3).

The consequence parts consist of layers (E) through (F), and the fuzzy reasoning is realized as:

$$R^i : \text{If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ then } y = w_{bi} \ (i = 1, 2, \cdots, n)$$

and

$$y = \frac{\sum_{i=1}^{n} \mu_i w_{bi}}{\sum_{i=1}^{n} \mu_i} = \sum_{i=1}^{n} \hat{\mu}_i w_{bi} \tag{4}$$

where R^i is the i-th fuzzy rule, A_{i1} and A_{i2} are fuzzy variables in the premise, w_{bi} is a constant, n is the number of fuzzy rules, and y is the inferred value.

2.2 Learning method of fuzzy neural networks

The weights w_b should be modified to identify fuzzy rules using the gradient descent method. In order to apply the gradient descent method, the squared error function is defined as:

$$E = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}e^2 \tag{5}$$

where y is the inferred value of FNN, and $\hat{y}$ is the desired value.

Using the gradient descent method, the weights can be modified as:

$$\begin{aligned} w_{bi}(k+1) &= w_{bi}(k) + \Delta w_{bi}(k) \\ &= w_{bi}(k) - \eta \frac{\partial E}{\partial w_{bi}(k)} \\ &= w_{bi}(k) - \eta \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_{bi}(k)} \\ &= w_{bi}(k) + \eta (y - \hat{y}) \hat{\mu}_i \\ &= w_{bi}(k) + \eta \cdot e \cdot \hat{\mu}_i \end{aligned} \tag{6}$$

where $\Delta w_{bi}(k) = w_{bi}(k) - w_{bi}(k-1)$, k means the k-th time to update the weights, and η is called the learning rate.

3. Design of a Generalized Predictive Controller Using Fuzzy Neural Networks

3.1 Structure of a generalized predictive control system

We discuss both continuous-time and discrete-time uncertain chaotic control 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 objective 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.

It is assumed that the closed-loop system output data are available on-line for the design and use of the controller. In the design, an on-line system prediction unit based on FNN is employed and a nonlinear feedback controller with a generalized predictive control scheme is implemented. Fig. 5 shows the overall configuration of the closed-loop control system, where the output $y(k)$ is to be controlled to track the reference, $y_r(k)$.

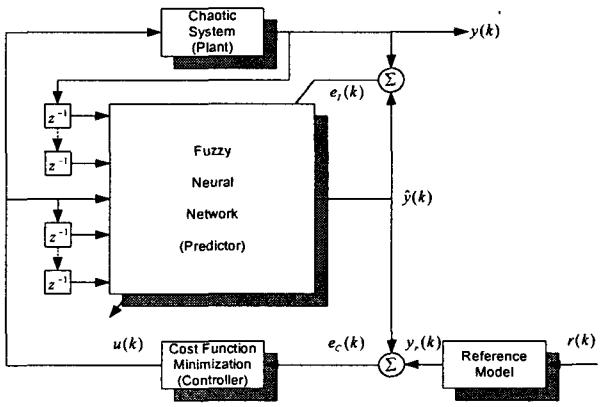

Fig. 5 Structure of a generalized predictive control system using the fuzzy nerual network

In Fig. 5, predictor parameters are tuned by the prediction error, $e_f(k)$, between the actual output of a chaotic nonlinear system and

that of a fuzzy neural network model. And the parameters of a generalized predictive controller are tuned by gradient descent method which uses control error, $e(k)$, between the prediction output of a fuzzy neural network model and the reference signal.

3.2 Design of a generalized predictive controller using FNN

We are looking for an optimal control signal, $u(k)$, for minimizing the following performance function:

$$\begin{aligned}\epsilon &= \frac{1}{2}\{(\hat{y}(k+1)-y_r(k+1))^2+\lambda(u(k)-u(k-1))^2\} \\ &= \frac{1}{2}\{e_C^2(k+1)+\lambda\Delta u^2(k)\}\end{aligned} \tag{7}$$

where $\hat{y}(k+1)$ is the predicted output of FNN, $y(k+1)$ is the reference signal, the positive constant λ weights the relative importance of control signal, and $\Delta u(k)=u(k)-u(k-1)$.

Using the FNN, the predicted output is formulated as

$$\hat{y}(k+1)=\sum_{i=1}^{H} w_{bi}F_i(\mathrm{x}) \tag{8}$$

where $F_i(\mathrm{x})$ is an inferred value of a ith fuzzy rule, and H is a total number of a fuzzy rule.

To minimize ϵ, $u(k)$ is recursively calculated via the gradient descent scheme.

$$u(k+1)=u(k)-\eta\frac{\partial\epsilon}{\partial u(k)} \tag{9}$$

where the positive constant η is the learning rate. It can be seen that the controller relies on the approximation performed by FNN. Therefore, it is necessary for the estimated output $\hat{y}(k+1)$ to approach the real system output $y(k+1)$ asymptotically, which can be achieved by keeping FNN training on-line. Differentiating the Eqn. (7) with respect to $u(k)$, we have:

$$\frac{\partial\epsilon}{\partial u(k)}=e_C(k+1)\frac{\partial\hat{y}(k+1)}{\partial u(k)}+\lambda\Delta u(k) \tag{10}$$

where $\frac{\partial\hat{y}(k+1)}{\partial u(k)}$ is the gradient of FNN model with respect to $u(k)$. It can be analytically evaluated by using the known FNN structure, Eqn. (8), as follows:

$$\begin{aligned}\frac{\partial\hat{y}(k+1)}{\partial u(k)} &= \sum_{i=1}^{H} w_{bi}F_i(\mathrm{x})G_i(\mathrm{x})\frac{\partial\mathrm{x}}{\partial u(k)} \\ &= \sum_{i=1}^{H}[w_{bi}F_i(\mathrm{x})G_i(\mathrm{x})]\frac{\partial\mathrm{x}}{\partial u(k)}\end{aligned} \tag{11}$$

where $\frac{\partial\mathrm{x}}{\partial u(k)}=[0,0,\cdots,1,0,\cdots 0]^T$ and we describe the computing procedure for $G_i(\mathrm{x})$ in Appendix A.1.

Finally, Eqn. (9) becomes:

$$\begin{aligned}u(k+1) &= u(k) \\ &-\eta\left[e_C(k+1)\sum_{i=1}^{H}[w_{bi}F_i(\mathrm{x})G_i(\mathrm{x})]\frac{\partial\mathrm{x}}{\partial u(k)}+\lambda\Delta u(k)\right]\end{aligned} \tag{12}$$

So far, we have described the algorithm for a one-step ahead predictive control scheme. Then, we let the algorithm described above be extended to a multi-step ahead control scheme, which considers not only the instant value of the control signal but also its future values. The future values of the references and the system outputs are needed to formulate the control signal. Since FNN model represents the system to be controlled asymptotically, it can be used to predict future values of the system output. The control performance function for a multi-step ahead predictive control is:

$$J=\frac{1}{2}\left[\sum_{i=1}^{N}(\hat{y}(k+i)-y_r(k+i))^2+\lambda\sum_{i=1}^{N}\Delta u^2(k+i-1)\right] \tag{13}$$

where N is the prediction horizon.

And, we denote the following vectors:

$$\begin{aligned}\overline{Y}_{N|k} &= [y_r(k+1),\ y_r(k+2),\ \cdots,\ y_r(k+N)]^T \\ \hat{Y}_{N|k} &= [\hat{y}(k+1),\ \hat{y}(k+2),\ \cdots,\ \hat{y}(k+N)]^T \\ E_{N|k} &= [e(k+1),\ e(k+2),\ \cdots,\ e(k+N)]^T \\ \overline{U}_{N|k} &= [\Delta u(k),\ \Delta u(k+1),\ \cdots,\ \Delta u(k+N-1)]^T \\ U_{N|k} &= [u(k),\ u(k+1),\ \cdots,\ u(k+N-1)]^T\end{aligned} \tag{14}$$

Using the vectors of the Eqn. (14), the Eqn. (13) is rewritten as follows:

$$J=\frac{1}{2}[E_{N|k}^T E_{N|k}+\lambda\overline{U}_{N|k}^T\overline{U}_{N|k}] \tag{15}$$

Thus, the control objective is to find $U_{N|k}$ such that J is minimized. By using the gradient descent scheme, the control sequence U can be updated at each iteration as follows:

$$U_{N|k+1}=U_{N|k}-\eta D_{N|k} \tag{16}$$

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 [21]. The search direction at time k is given by

$$D_{N|k} = P_{N|k}\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}E_{N|k} + \lambda M\bar{U}_{N|k} \tag{17}$$

where $P_{N|k}$, denoted by the projection matrix, is an $N\times N$ diagonal matrix with unity initial value, $P_{N|k_0}=0$, and the M is a $N\times N$ matrix as follows:

$$M=\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{18}$$

And, the partial derivative of $\hat{Y}_{N|k}$ with respect to $U_{N|k}$ is given by:

$$\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}=\begin{bmatrix} \frac{\partial \hat{y}(k+1)}{\partial u(k)} & \frac{\partial \hat{y}(k+2)}{\partial u(k)} & \cdots & \frac{\partial \hat{y}(k+N)}{\partial u(k)} \\ 0 & \frac{\partial \hat{y}(k+2)}{\partial u(k+1)} & \cdots & \frac{\partial \hat{y}(k+N)}{\partial u(k+1)} \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{\partial \hat{y}(k+N)}{\partial u(k+N-1)} \end{bmatrix} \tag{19}$$

where $\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}$ is the gradient of the control performance function with respect to $U_{N|k}$, which can be derived from FNN model. We also describe the computing procedure for $\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}$ in Appendix A.2.

Each individual element of the control sequence is updated by clipping the results obtained from Eqn. (16) according to:

$$\underline{\Delta u} \le \Delta u(k+i-1) \le \overline{\Delta u} \tag{20}$$

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 heuristically chosen to be some very small values.

The projection matrix $P_{N|k}$ is then updated according to $U_{N|k}$ at each iteration, by

$$P_{N|k}(i,i) = \begin{cases} 0 & \text{if } \Delta u \le \Delta u(k+i-1) \le \overline{\Delta u} \\ P_{N|k-1}(i,i) & otherwise \end{cases}$$
$$i=1,2,\cdots,N \tag{21}$$

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

4. Simulation Results

This paper performed simulation under PC with Intel Pentium Microprocessor using MATLAB 6.x program package. In this section, we present some simulation results to validate the control performance of the proposed controller for both continuous-time and discrete-time chaotic systems. Also, in order to evaluate performance of proposed controller, we compare the results of a FNN based generalized predictive controller with those of a neural network(NN) based generalized predictive controller.

4.1 Objects of simulation

We consider the Duffing system and Hénon system as representative continuous-time and discrete-time chaotic systems. The state equations of these systems are as follows:

Duffing:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\end{bmatrix}=\begin{bmatrix} y(t) \\ a_1x(t)-x^3(t)-a_2y(t)+bcos(wt)+u(t)\end{bmatrix} \tag{22}$$

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

Hénon: $$\begin{bmatrix}x_{n+1}\\ y_{n+1}\end{bmatrix}=\begin{bmatrix} y_n+1-ax_n^2 \\ bx_n+u_n\end{bmatrix} \tag{23}$$

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

Figs. 6 and 7 are the strange attractors of Duffing and Hénon systems, respectively.

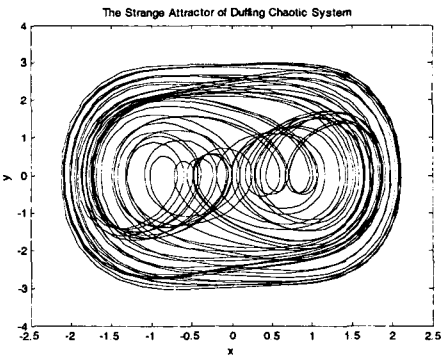

Fig. 6 Strange attractor of the Duffing system

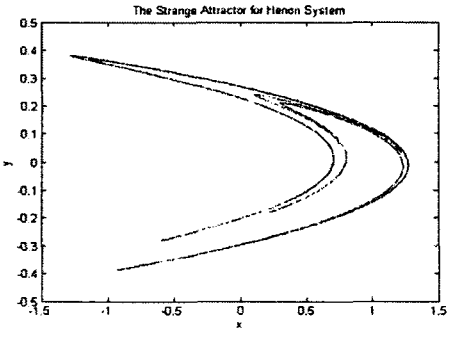

Fig. 7 Strange attractor of Hénon system

4.2 Controlling the Duffing system

The control objective for the Duffing system is to follow the unstable periodic solution of the Duffing equation. In other words, as the value of b in Eqn. (22) varies, the Duffing equation may have either a chaotic or a periodic solution. In the tracking control for Duffing system, both a FNN and NN predictors have two past outputs of the plant, one current output and one past output of controller as the inputs. A FNN predictor uses twelve membership functions in each input and the hidden layer of the NN predictor has five nodes. Also, the learning rates of a FNN predictor, a NN predictor, and a generalized predictive controller are 0.1, 0.1, and 0.01, respectively. And, each sampling period is 0.05 second. We define the initial system state as (1, 0) and the reference signal as one periodic solution in the case of b= 2.3. Finally, the weighting factor, λ, of a generalized predictive controller is 0.1.

The membership functions of the FNN used for this simulation are shown in Fig. 8. And, Fig. 9 shows the on-line prediction result for a FNN predictor, and Fig. 10 shows the tracking control result for a FNN based generalized predictive controller. Finally, Fig. 11 shows the control signal using the FNN for Duffing system.

Also, Fig. 12 shows the on-line prediction result for a NN predictor, and Fig. 13 shows the tracking control result for a NN based generalized predictive controller. And, Fig. 14 shows the control signal using the NN for Duffing system.

In this paper, we uses the mean-squared errors (MSEs) as the performance index. The MSEs of system prediction and control performance are given in Table 1.

From the results obtained above, we can see that although prediction errors (state x: 0.0317, state y: 0.1231) of a FNN model are more than that (state x: 0.0108, state y: 0.0113) of a NN model, a FNN based generalized predictive controller shows better control performance, and it is faster and more effective, as compared with a NN based generalized predictive control. In other words, the MSEs (state x : 0.1044, state y : 0.5554) of a FNN based generalized predictive control are better than the MSEs (state x3.7029, state y: 11.4418) of a NN based generalized predictive control. Also, unlike control signal shown Fig. 14, the control signal in Fig. 11, which is the output of a FNN based generalized predictive controller, is more stable. Especially control signal in Fig. 11 converges at about 1000 steps. On the other hand, control signal in Fig. 14 not converge.

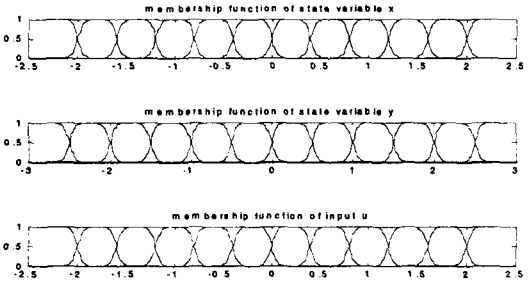

Fig. 8 Membership functions of FNN (Duffing system)

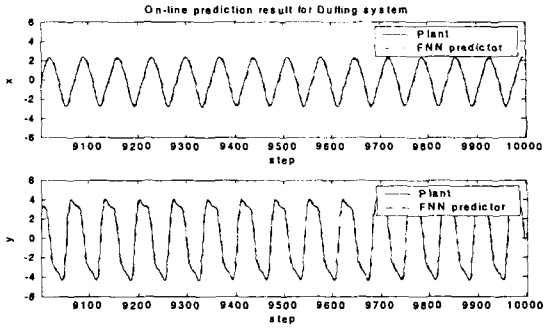

Fig. 9 On-line prediction results using the FNN predictor for Duffing system

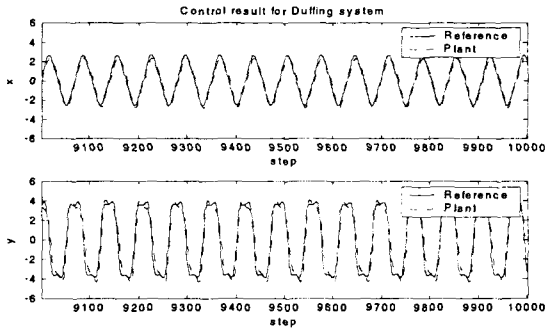

Fig. 10 Tracking control results for Duffing system (FNN)

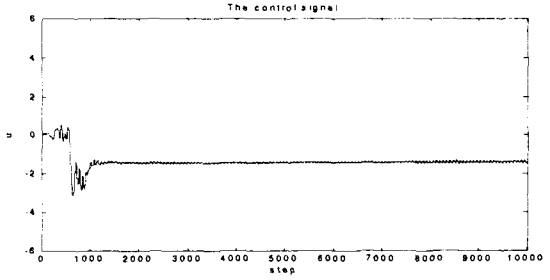

Fig. 11 Control signal for Duffing system (FNN)

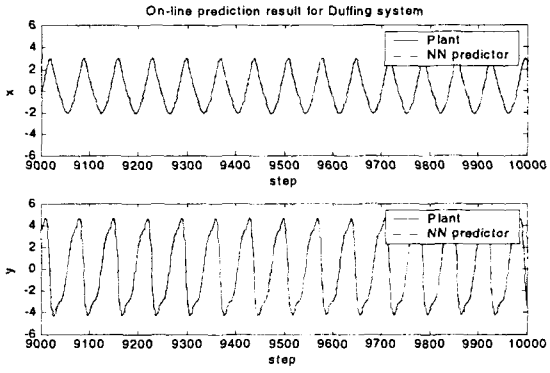

Fig. 12 On-line prediction results using the NN predictor for Duffing system

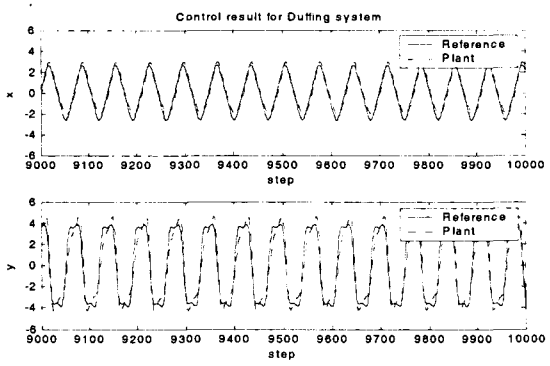

Fig. 13 Tracking control results for Duffing system (NN)

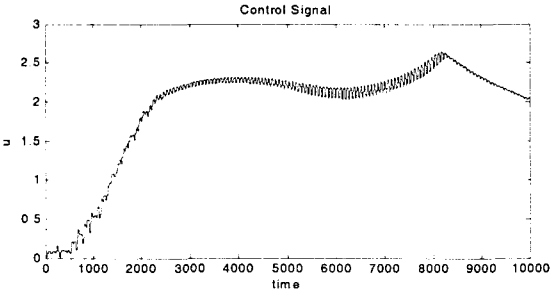

Fig. 14 Control signal for Duffing system (NN)

Table 1 Prediction and control errors for Duffing system

		State x	State y
Fuzzy Neural Network (FNN)	Prediction (MSE)	0.0317	0.1231
	Control (MSE)	0.1044	0.5554
Neural Network (NN)	Prediction (MSE)	0.0108	0.0113
	Control (MSE)	3.7029	11.4418

4.3 Controlling the Hénon system

n this subsection, the simulation results of the proposed generalized predictive control scheme for the discrete-time chaotic systems are presented. Also, in order to evaluate the control performance of our method, we compare the simulation results of the proposed controller with those of a NN based generalized predictive controller. In this simulation, the control objective for the Hénon system is to regulate the chaotic orbit to a desired point. Both FNN and NN predictors for Hénon system have four inputs; two past outputs of Hénon system, one current control signal of the controller, and one past control signal of the controller. FNN has twelve membership functions in each input and the node number of the hidden layer for the NN is 10. The learning rate for a FNN predictor, a NN predictor and a generalized predictive controller are 0.1, 0.1, and 0.01, respectively. For the Hénon system, the system initial state starts from(0, 0) and the reference signal is (0, -1). Finally, the weighting factor of control signal for a generalized predictive controller is 0.1.

Fig. 15 shows the membership functions of a FNN for Hénon system, Figs. 16 and 17 show the regulation control results and the on-line prediction result for a FNN based generalized predictive controller, respectively. Fig. 18 shows the control signal output for a FNN based generalized predictive controller.

Also, Fig. 19 shows the regulation control result for a NN based generalized predictive controller, and Fig. 20 shows the on-line prediction result for a NN predictor. Fig. 21 shows control signal using a NN for Hénon system.

The MSEs of system prediction and control errors for both controllers are given in Table 2.

From Figs. 16 and 19, we can see that the chaotic signal controlled by a FNN based generalized predictive controller converges to the desired point at about 230 steps, but the chaotic signal controlled by a NN based generalized predictive controller converges to the desired point at about 750 steps. From the results obtained above, we can see that a FNN based generalized predictive controller shows the better control performance, as compared with a NN based generalized predictive controller, and that the chaotic signal controlled by a FNN based generalized predictive controller converges to the desired point rapidly. In other words, the prediction (state x: 0.0087, state y: 0.0149) and control errors (state x 0.0101, state y : 0.0192) of a FNN based generalized predictive control are better than the prediction (state x: 0.0916, state y : 0.0164) and control errors (state x: 0.1464, state y : 0.3990) of a NN based generalized predictive control. And, control signal of a FNN based generalized predictive controller is faster than control signal of a NN based generalized predictive controller.

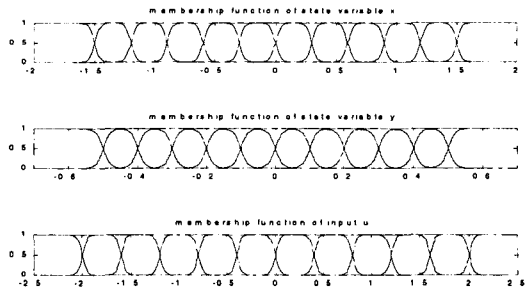

Fig. 15 The membership functions of FNN (Hénon system)

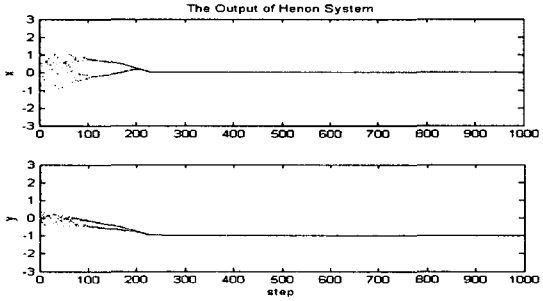

Fig 16 Regulation control results for Hénon system (FNN)

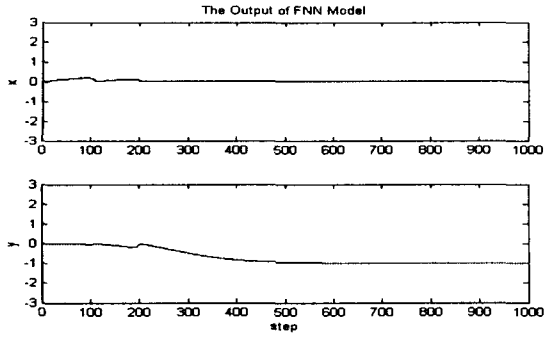

Fig. 17 On-line prediction result using the FNN predictor for Hénon system

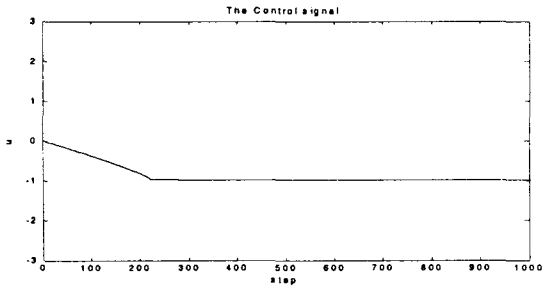

Fig. 18 Control signal for Héno system (FNN)

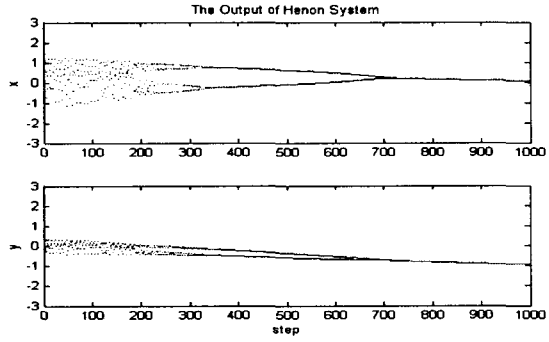

Fig. 19 Regulation control results for Hénon system (NN)

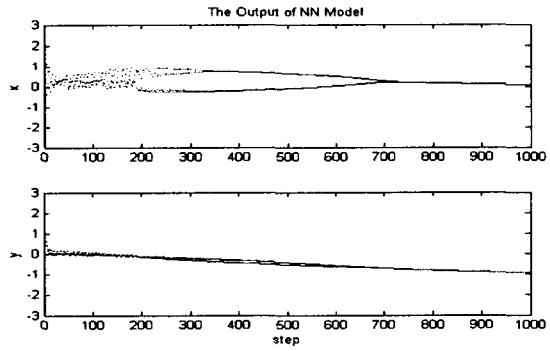

Fig. 20 On-line prediction results using the NN predictor for Hénon system

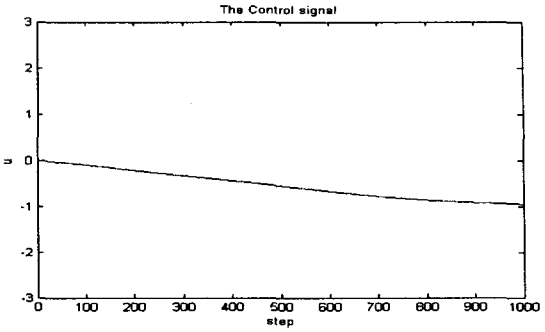

Fig. 21 Control signal for Héno system (NN)

Table 2 Prediction and control errors for Hénon system

		State x	State y
Fuzzy Neural Network (FNN)	Prediction (MSE)	0.0087	0.0149
	Control (MSE)	0.0101	0.0192
Neural Network (NN)	Prediction (MSE)	0.0916	0.0164
	Control (MSE)	0.1464	0.3990

5. Conclusions

In this paper, we have presented the design method of a generalized predictive controller based on a fuzzy neural networ model, which was used to perform the multi-step prediction on-line for the intelligent control of chaotic systems whose mathematica models are unknown. In our design method, the parameters of bot predictor and controller were tuned by a simple gradient descer scheme, and the weight parameters of FNN were determine adaptively during the operation of the system. In order to design FNN based generalized predictive controller effectively, this pap has described a computing procedure for each of the two importar parameters mentioned above. Also, we have introduced a projectio matrix to determine the control input, which decreased the contr performance function very rapidly. Finally, in order to evaluate th performance of our controller, we have applied the proposed metho to the Duffing and Hénon systems, which are two representative continuous-time and discrete-time chaotic systems, respectively. Th simulation results have shown that a FNN based generalize predictive control scheme has the faster convergence property an more accurate control performance than those obtained by som conventional NN based control schemes. Through simulations, w have also verified that the proposed generalized predictive controll scheme works well for various chaotic systems.

Acknowledgements

The work was supported by the Korea Research Foundation Grant (KRF-2000-EA0048)

References

[1] J. J. Dorning and J. H. Kim, "Bridging the Gap Between the Science of Chaos and Its Technological Application", In Applied Chaos, J. H. Kim and J. Stringer (Eds.), Wiley, New York, pp. 3-30, 1993.

[2] R. S. Mackay, "Some Thoughts on Chaos in Engineering", In Towards the Harnessing of Chaos, M. Yamaguti (Ed.) Elsevier Science, New York, pp. 73-82, 1994.

[3] Steven. H and Strogatz, *Nonlinear Dynamics and Chaos*, Addison-Wesley Publishing, Company, 1994.

[4] G. Chen and X. Dong, "From chaos to order-perspectives and methodologies in controlling chaotic nonlinear dynamical systems", International Journal of Bifurcation and Chaos Vol. 3, No. 6, pp 1363-1409, 1993.

[5] H. G. Davies and K. Rangavajhula, "Nonstationary Response of a Foldedband Attractor", Proceedings of International Mechanical Engineering Congress and Exposition, Chicago, IL, pp. 115-117, 1993.

[6] E. Ott, C. Grebogi, and J. A. Yorke, "Controlling chaos", Physical Review Letter, Vol. 64, No. 11, pp. 1196-1199, 1990.

[7] G. Chen and X. Dong, "On feedback control of chaotic nonlinear dynamic systems", International Journal of Bifurcation and Chaos, Vol. 2, No. 2, pp. 407-411, 1992.

[8] G. Chen and X. Dong, "On feedback control of chaotic continuous-time systems", IEEE Transaction on Circuits and Systems, Vol. 40, No. 9, pp. 591-601, 1993.

[9] T. T. Hartley and F. Mossaybei, "A Classical Approach to Controlling the Lorenz Equations", International Journal of Bifurcation and Chaos, Vol. 2, No. 4, pp. 881-887, 1992.

[10] J.M. Joo and J. B. Park, "Control of the Differentially Flat Lorenz System", International Journal of Bifurcation and Chaos, Vol. 11, No. 7, pp. 1989-1996, 2001.

[11] K. S. Park, J. B. Park, Y. H. Choi, T. S. Yoon and G. Chen, "Generalized Predictive Control of Discrete-time Chaotic Systems", International Journal of Bifurcation and Chaos, Vol. 8, No. 7, pp. 1591-1597, 1998.

[12] H. Qin, H. Zhang and G. Chen, "Neural network based adaptive control of uncertain chaotic systems", Proceedings of the IEEE Symposium Circuit and Systems, Monterey, pp. III 318-321, June 1998.

[13] H. O. Wang, K. Tanaka and T. Ikeda, "Fuzzy modeling and control of chaotic systems", Proceedings of the IEEE Symposium Circuits and Systems, Atlanta, pp. 209-212, June 1996.

[14] K. B. Kim, J. B. Park, Y. H. Choi and G. Chen, "Control of chaotic dynamical systems using radial basis function network approximators", Information Sciences, Vol. 130, pp. 165-183, 2000.

[15] T. W. Frison, "Controlling chaos with a neural network", Proceedings of the International Conference on Neural Networks, Baltimore, MD, pp. 75-80, 1992.

[16] J. S. Oh, Y. H. Choi and J. B. Park, "A Study on Design of Fuzzy Controller for Chaotic Nonlinear Systems", Proceedings of the IEEK Fall Annual Conference, Vol. 20, No. 2, pp. 277-280, 1997.

[17] S. Horikawa, T. Furuhashi and Y. Uchikawa, "On identification of structures in premise of a fuzzy model using a fuzzy neural networks", Proceedings of the 2nd IEEE International Conference on Fyzzy Systems, pp. 661-666, 1993.

[18] T. Hasegawa, S. Horikawa, T. Furuhashi and Y. Uchikawa, "On design of adaptive fuzzy neural networks and description of its dynamical behavior", Fuzzy Sets and Systems, Vol. 71, No. 1, pp. 3-23, 1995.

[19] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control, Part Ⅰ, The Basic Algorithm", Automatica, Vol. 23, No. 2, pp. 137-148, 1987.

[20] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control, Part ?, Extensions and Interpretations", Automatica, Vol. 23, No. 2, pp. 149-160, 1987.

[21] J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming, Part Ⅰ, Linear Constants", SIAM Journal Applied Mathematics, Vol. 8, pp. 181-217, 1960.

[22] C. C. Lee, "Fuzzy Logic in Control System: Fuzzy Logic in Controller Part Ⅰ", IEEE Transaction on System, Man and Cybernetics, Vol. 20, No. 2, pp. 404-418, 1990.

[23] C. C. Lee, "Fuzzy Logic in Control System: Fuzzy Logic in Controller Part Ⅱ", IEEE Transaction on System, Man and Cybernetics, Vol. 20, No. 2, pp. 419-435, 1990.

[24] K. Hornik, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, Vol. 2, pp. 359-366, 1989.

[25] T. Poggio and F. Girosi, "Networks for Approximation and Learning", Proceedings of the IEEE, Vol. 78, No. 9, pp. 1481-1497, 1990.

[26] S. M. Kim, Y. H. Choi, J. B. Park and Y. H. Joo, "Direct Adaptive Control of Chaotic Nonlinear Systems Using a Feedforward Neural Network", Proceedings of the KIEE Summer Annual Conference, pp. 401-403, 1998.

[27] K. H. Oh, J. M. Joo, K. S. Park, J. B. Park and Y. H. Choi, "A Study on the Intelligent Control of Chaotic Nonlinear Systems Using Neural Networks", Proceedings of the 14th Korea Automatic Control Conference, pp. 453-456, 1996.

Appendix A

The design method of the controller used in this paper is based on the gradient descent scheme. In this appendix, we describe a computing procedure for each of the two important parameters for the predictor and the controller in the design. The mathematical expressions are all based on the structure of FNN shown in Fig. 5.

A.1 The computing procedure for $G(x)$

We first describe the computing procedure for $G(x)$ of Eqn. (13). Assume that FNN predictor has two inputs (one past value of the chaotic system and one current value of the control signal) and three membership functions (for simplicity of description here):

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{i=1}^{9} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k)} \tag{A.1}$$

where $\mathbf{x} = [x(k) \;\; u(k)]^T$. The vector $G_i(\mathbf{x})$ is computed as follows.

$$i=1 \Rightarrow G_1(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g11}\exp\{-w_{g11}(x(k)-w_{c11})\}}{1+\exp\{-w_{g11}(x(k)-w_{c11})\}} \\ \dfrac{-w_{g12}\exp\{-w_{g12}(u(k)-w_{c12})\}}{1+\exp\{-w_{g12}(u(k)-w_{c12})\}} \end{bmatrix}^T$$

$$i=2 \Rightarrow G_2(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g11}\exp\{-w_{g11}(x(k)-w_{c11})\}}{1+\exp\{-w_{g11}(x(k)-w_{c11})\}} \\ \dfrac{-w_{g22}\exp\{-w_{g22}(u(k)-w_{c22})\}}{1+\exp\{-w_{g22}(u(k)-w_{c22})\}} - \dfrac{-w_{g32}\exp\{-w_{g32}(u(k)-w_{c32})\}}{1+\exp\{-w_{g32}(u(k)-w_{c32})\}} \end{bmatrix}^T$$

$$i=3 \Rightarrow G_3(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g11}\exp\{-w_{g11}(x(k)-w_{c11})\}}{1+\exp\{-w_{g11}(x(k)-w_{c11})\}} \\ \dfrac{-w_{g42}\exp\{-w_{g42}(u(k)-w_{c42})\}}{1+\exp\{-w_{g42}(u(k)-w_{c42})\}} \end{bmatrix}^T$$

$$i=4 \Rightarrow G_4(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g21}\exp\{-w_{g21}(x(k)-w_{c21})\}}{1+\exp\{-w_{g21}(x(k)-w_{c21})\}} - \dfrac{-w_{g31}\exp\{-w_{g31}(x(k)-w_{c31})\}}{1+\exp\{-w_{g31}(x(k)-w_{c31})\}} \\ \dfrac{-w_{g12}\exp\{-w_{g12}(u(k)-w_{c12})\}}{1+\exp\{-w_{g12}(u(k)-w_{c12})\}} \end{bmatrix}^T$$

$$i=5 \Rightarrow G_5(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g21}\exp\{-w_{g21}(x(k)-w_{c21})\}}{1+\exp\{-w_{g21}(x(k)-w_{c21})\}} - \dfrac{-w_{g31}\exp\{-w_{g31}(x(k)-w_{c31})\}}{1+\exp\{-w_{g31}(x(k)-w_{c31})\}} \\ \dfrac{-w_{g22}\exp\{-w_{g22}(u(k)-w_{c22})\}}{1+\exp\{-w_{g22}(u(k)-w_{c22})\}} - \dfrac{-w_{g32}\exp\{-w_{g32}(u(k)-w_{c32})\}}{1+\exp\{-w_{g32}(u(k)-w_{c32})\}} \end{bmatrix}^T$$

$$i=6 \Rightarrow G_6(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g21}\exp\{-w_{g21}(x(k)-w_{c21})\}}{1+\exp\{-w_{g21}(x(k)-w_{c21})\}} - \dfrac{-w_{g31}\exp\{-w_{g31}(x(k)-w_{c31})\}}{1+\exp\{-w_{g31}(x(k)-w_{c31})\}} \\ \dfrac{-w_{g42}\exp\{-w_{g42}(u(k)-w_{c42})\}}{1+\exp\{-w_{g42}(u(k)-w_{c42})\}} \end{bmatrix}^T$$

$$i=7 \Rightarrow G_7(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g41}\exp\{-w_{g41}(x(k)-w_{c41})\}}{1+\exp\{-w_{g41}(x(k)-w_{c41})\}} \\ \dfrac{-w_{g12}\exp\{-w_{g12}(u(k)-w_{c12})\}}{1+\exp\{-w_{g12}(u(k)-w_{c12})\}} \end{bmatrix}^T$$

$$i=8 \Rightarrow G_8(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g41}\exp\{-w_{g41}(x(k)-w_{c41})\}}{1+\exp\{-w_{g41}(x(k)-w_{c41})\}} \\ \dfrac{-w_{g22}\exp\{-w_{g22}(u(k)-w_{c22})\}}{1+\exp\{-w_{g22}(u(k)-w_{c22})\}} - \dfrac{-w_{g32}\exp\{-w_{g32}(u(k)-w_{c32})\}}{1+\exp\{-w_{g32}(u(k)-w_{c32})\}} \end{bmatrix}^T$$

$$i=9 \Rightarrow G_9(\mathbf{x}) = \begin{bmatrix} \dfrac{-w_{g41}\exp\{-w_{g41}(x(k)-w_{c41})\}}{1+\exp\{-w_{g41}(x(k)-w_{c41})\}} \\ \dfrac{-w_{g42}\exp\{-w_{g42}(u(k)-w_{c42})\}}{1+\exp\{-w_{g42}(u(k)-w_{c42})\}} \end{bmatrix}^T$$

A.2 The computing procedure for $\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}$

We need to find the important parameter $\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}$ in the controller design. We can find the Jacobian matrix $\frac{\partial \hat{Y}_{N|k}}{\partial U_{N|k}}$ by differentiating $\hat{Y}_{N|k}$ with respect to $U_{N|k}$. The following illustrates the procedure of computing the elements in the Jacobian matrix for $N=3$.

1st column :

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\frac{\partial \hat{y}(k+1)}{\partial u(k+1)} = 0$$

$$\frac{\partial \hat{y}(k+1)}{\partial u(k+2)} = 0$$

where $\mathbf{x} = [\, \hat{y}(k) \;\; \hat{y}(k-1) \;\; u(k) \;\; u(k-1) \,]^T$.

2nd column :

$$\frac{\partial \hat{y}(k+2)}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} \dfrac{\partial \hat{y}(k+1)}{\partial u(k)} \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{\partial \hat{y}(k+2)}{\partial u(k+1)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k+1)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\frac{\partial \hat{y}(k+2)}{\partial u(k+2)} = 0$$

where $\mathbf{x} = [\, \hat{y}(k+1) \;\; \hat{y}(k) \;\; u(k+1) \;\; u(k) \,]^T$.

3rd column :

$$\frac{\partial \hat{y}(k+3)}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} \dfrac{\partial \hat{y}(k+2)}{\partial u(k)} \\ \dfrac{\partial \hat{y}(k+1)}{\partial u(k)} \\ 0 \\ 0 \end{bmatrix}$$

$$\frac{\partial \hat{y}(k+3)}{\partial u(k+1)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k+1)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} \dfrac{\partial \hat{y}(k+2)}{\partial u(k+1)} \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{\partial \hat{y}(k+3)}{\partial u(k+2)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \frac{\partial \mathbf{x}}{\partial u(k+2)} = \sum_{i=1}^{H} [w_{bi} F_i(\mathbf{x}) G_i(\mathbf{x})] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

where $\mathbf{x} = [\, \hat{y}(k+2) \;\; \hat{y}(k+1) \;\; u(k+2) \;\; u(k+1) \,]^T$.

저 자 소 개

최 종 태 (崔 鍾 泰)

1975년 5월 31일생. 1998년 경기대 전자공학과 졸업(공학사). 2000년 동 대학원 전자공학과 졸업(공학석사). 2000년~현재 연세대학원 전기전자공학과 박사과정.

Tel : 02-2123-2773, Fax : 02-361-4539

E-mail : jtchoi@control.yonsei.ac.kr

최 윤 호 (崔 允 浩)

1957년 1월 14일생. 1980년 연세대 전기공학과 졸업(공학사). 1982년 동 대학원 전기공학과 졸업(공학석사). 1991년 동 대학원 전기공학과 졸업(공학박사). 2000년~2002년 미국 오하이오 주립대학교 방문교수. 1993년~현재 경기대 전자공학부 부교수

Tel : 031-249-9801, Fax : 031-249-9796

E-mail : yhchoi@kuic.kyonggi.ac.kr