Wavelet Neural Network Based Generalized Predictive Control of Chaotic Systems Using EKF Training Algorithm

○Kyung Ju Kim*, Jin Bae Park ** and Yoon Ho Choi ***

* Department of Electrical Electronic Engineering, Yonsei University, Seoul, Korea
(Tel : +82-2-2123-2773; E-mail: mirukkj@control.yonsei.ac.kr)
** Department of Electrical Electronic Engineering, Yonsei University, Seoul, Korea
(Tel : +82-2-2123-2773; E-mail: jbpark@control.yonsei.ac.kr)
*** School of Electronic Engineering, Kyonggi University, Suwon, Korea
(Tel : +82-31-249-9801; E-mail:yhchoi@kyonggi.ac.kr)

Abstract: In this paper, we presented a predictive control technique, which is based on wavelet neural network (WNN), for the control of chaotic systems whose precise mathematical models are not available. The WNN is motivated by both the multilayer feedforward neural network definition and wavelet decomposition. The wavelet theory improves the convergence of neural network. In order to design predictive controller effectively, the WNN is used as the predictor whose parameters are tuned by error between the output of actual plant and the output of WNN. Also the training method for the finding a good WNN model is the Extended Kalman algorithm which updates network parameters to converge to the reference signal during a few iterations. The benefit of EKF training method is that the WNN model can have better accuracy for the unknown plant. Finally, through computer simulations, we confirmed the performance of the proposed control method.

Keywords: Wavelet Neural Network, Generalized Predictive Control, Chaotic System, Extended Kalman Filter, Chaos Control

1. Introduction

The study of chaos has led to many interesting over the past few decades, and many research results are published. A nonlinear dynamical system with a chaotic feature produces motion on the attractor that has random-like properties. Due to unpredictability and irregularity, the chaotic phenomena lead systems to be unstable or performance-degraded situations.

In 1990, Ott et al. proposed the so-called OGY method, by which the chaotic phenomenon of a dynamical system can be stabilized by a small perturbation of an accessible system parameter when the chaotic orbit approaches a periodic orbit near a saddle point. Since then, a number of successful control methods and techniques for controlling chaotic systems have been developed [1-4]. Among these chaos control techniques, the conventional control techniques such as optimal controls, adaptive controls and robust controls were also introduced to control chaotic systems, and these kinds of techniques have been shown effective [5, 6]. Notice, however, that most of these techniques can be applied to control chaotic systems only when the exact (or, at least, a good approximate) mathematical model of the chaotic system is available. To overcome this shortage, predictive control methods, which may be considered as a kind of adaptive control strategy, are suggested for controlling unknown chaotic systems. For example, Park et al. presented a generalized predictive control method based on an ARMAX model for chaos control for discrete-time systems [7].

In the meanwhile, the neural network (NN) has been used to control nonlinear and chaotic systems, because no mathematical models are needed. But the NN has defects, which come from their inherent characteristics, such as slow convergence, complex calculation. Accordingly, the wavelet neural network (WNN), which has the advantages of high resolution of wavelets, has been proposed to guarantee the fast convergence [8].

Generally, the performance of control method, which uses the universal approximator, depends on the accuracy of approximator. Therefore we need superior training algorithm of network for better performance. Usually, the gradient descent (GD) method is used to train the network parameter for networks (NN and WNN, etc.), but GD method has some drawback, such as local minima problem. Another training method, genetic algorithm (GA), which is the intelligent method with needless mathematical calculation, has a long training time. To overcome this problem, we use the Extended Kalman Filter (EFK) training method [9] which updates network parameters to converge during a few iterations. The benefit of EKF training method is that the WNN model can have better accuracy for the unknown plant.

In this paper, we propose a predictive control technique, which is based on WNN, for the control of chaotic systems whose precise mathematical models are not available. In our method, the WNN is used as the universal approximator of the chaotic plant (Duffing) and the model is made by on-line and off-line identification (ID). We apply the EKF training method to making good WNN ID model

In our predictive controller using the WNN model is developed in such a way that the parameters of the predictive controller are adjusted by using the gradient descent scheme [10], where the difference between the actual output and the reference signal is used as control input.

This paper is organized as follows: In Section 2, we present some basics of the WNN and the EKF training method. Section 3 discusses the WNN based GPC strategy. Section 4 presents a simulation result. Finally, Section 5 gives some concluding remarks.

2. Wavelet Neural Network

2.1 WNN structure

A schematic diagram of the proposed WNN structure is shown in Fig. 1, which has Ni inputs, Nw wavelet nodes and one output. The WNN structure consists of three layers.

The layer 1 is an input node layer. This layer accepts the input variables and transmits the accepted inputs to the next layer directly. The layer 2 is a wavelet node layer. Each node of this layer has a mother wavelet and its output is the product of the mother wavelets. The layer 3 is the output layer. The node of output is a linear combination of consequences obtained from the weighted output of the layer 2 and weighted input of the layer 1.

In this paper, we select the first derivative of a Gaussian function as a mother wavelet function:

$$\varphi(z_{jk}) = -z_{jk}\exp\left(-\frac{1}{2}z_{jk}{}^2\right). \quad (1)$$

The output of each wavelet node, $\phi_j(\mathbf{X})$, is derived from its mother wavelet $\varphi(z_{jk})$ as follows:

$$\phi_j(\mathbf{X}) = \prod_{j=1}^{N_W}\varphi(z_{jk}) \text{ with } z_{jk} = \frac{x_k - m_{jk}}{d_{jk}} \quad (2)$$

where x_k denotes the input of the WNN, and m_{jk}, d_{jk} are translation and dilation parameters of the WNN, respectively. The subscript jk indicates the k th input term of the j th wavelet.

And the output of WNN is

$$y = \sum_{j=1}^{Nw} c_j\phi_j(\mathbf{X}) + \sum_{k=1}^{Ni} a_k x_k, \quad (3)$$

where a_k is the weight between input and output layers and the c_j is the weight between wavelet layer and output layer.

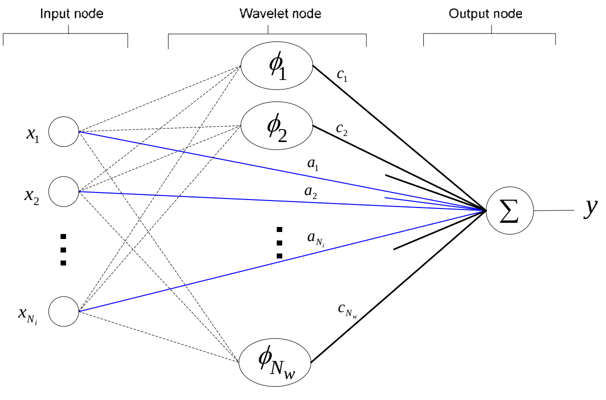

Fig. 1 The WNN structure.

2.2 EKF training method

In order to apply the EKF training algorithm, we let the weights in the network represent the state $\boldsymbol{\theta}$ of the nonlinear system as the weight vector, i.e.

$$\boldsymbol{\theta} = [a_1,\cdots,a_{Ni},c_1,\cdots,c_{Nw},m_{11},\cdots,m_{NwNi},d_{11},\cdots,d_{NwNi}]^T. \quad (4)$$

The vector θ contains the all weight of WNN, and its dimension equal to the total number of weights in the WNN.

Thus, the system model is

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \omega_n, \quad n > 2, \quad (5)$$

$$\begin{aligned}\hat{y}_{n+1} &= h(\boldsymbol{\theta}_{n+1},\mathbf{X}_{n+1}) + \upsilon_n \\ &= \sum_{j=1}^{Nw} c_{j,n+1}\phi_{j,n+1}(\mathbf{X}_{n+1}) + \sum_{k=1}^{Ni} a_{k,n+1}x_{k,n+1} + \upsilon_n,\end{aligned} \quad (6)$$

where the variable ω_n, υ_n are the degree of updating state and the degree of measurement residual, respectively. They are assumed to be independent of each other and with the following normal probability distributions:

$$p(\omega_n) \sim N(0,Q_n), \quad (7)$$

$$p(\upsilon_n) \sim N(0,R_n). \quad (8)$$

Applying the EKF to Eqs. (5) and (6), then we obtain Eqs. (9)~(11) as follows:

$$K_n = P_n H_n\left[R_n + H_n{}^T P_n H_n\right]^{-1}, \quad (9)$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + K_n\xi_n, \quad (10)$$

$$P_{n+1} = P_n - K_n H_n{}^T P_n + Q_n. \quad (11)$$

The error vector is $\xi_n = y_n - \hat{y}_n$, where y_n is the target vector and $\hat{y}_n$ is the output vector of WNN for the nth presentation of a training procedure. The H_k in Eq. (12) is a matrix of derivatives of the WNN's output with respect to all trainable weight parameters. The error covariance matrix P_n evolves recursively with the weight vector estimate.

$$\begin{aligned}H &= \frac{\partial\hat{y}}{\partial\boldsymbol{\theta}} \\ &= \left[\frac{\partial\hat{y}}{\partial a_1} \cdots \frac{\partial\hat{y}}{\partial a_{Ni}} \; \frac{\partial\hat{y}}{\partial c_1} \cdots \frac{\partial\hat{y}}{\partial c_{Nw}} \right. \\ &\qquad \left. \frac{\partial\hat{y}}{\partial m_{11}} \cdots \frac{\partial\hat{y}}{\partial m_{NwNi}} \; \frac{\partial\hat{y}}{\partial d_{11}} \cdots \frac{\partial\hat{y}}{\partial c_{NwNi}}\right]^T.\end{aligned} \quad (12)$$

Each elements of H are as follows:

$$\frac{\partial\hat{y}}{\partial a_k} = x_k, \quad (13)$$

$$\frac{\partial\hat{y}}{\partial c_j} = \phi_j(\mathbf{X}), \quad (14)$$

$$\begin{aligned}\frac{\partial\hat{y}}{\partial m_{jk}} &= \frac{\partial\hat{y}}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial m_{jk}} \\ &= -\frac{c_j}{d_{jk}}\frac{\partial\phi_j(\mathbf{X})}{\partial z_{jk}},\end{aligned} \quad (15)$$

$$\begin{aligned}\frac{\partial\hat{y}}{\partial d_{jk}} &= \frac{\partial\hat{y}}{\partial z_{jk}}\frac{\partial z_{jk}}{\partial d_{jk}} \\ &= -z_{jk}\frac{c_j}{d_{jk}}\frac{\partial\phi_j(\mathbf{X})}{\partial z_{jk}},\end{aligned} \quad (16)$$

$$\frac{\partial \phi_j(\mathbf{X})}{\partial z_{jk}} = \varphi(z_{j1})\varphi(z_{j2})\cdots\varphi'(z_{jk})\cdots\varphi(z_{jNi}), \qquad (17)$$

where $k = 1,2,\cdots,Ni$, $j = 1,2,\cdots,Nw$, $\varphi'(z_{jk})$ means the derivative of the mother wavelet with respect to z_{jk}, that is

$$\varphi'(z_{jk}) = (z_{jk}^{\ 2} - 1)\exp\left(-\frac{1}{2} z_{jk}^{\ 2}\right). \qquad (18)$$

The time index n of Eqs. (12)-(17) is omitted for simplicity.

The EKF training algorithm repeats Eqs. (9)-(11) for updating the weight of the WNN. Before the update process, the error covariance matrix P_n must be initialized at the beginning of training.

3. Generalized Predictive Control

In design of GPC, the WNN are applied for the control of chaotic systems. In most applications of model predictive techniques, a linear model is used to predict the process behavior over the prediction horizon [11]. And some works were done to extend predictive control techniques to incorporate nonlinear models [12]. The most expensive part of the realization of a nonlinear predictive control scheme is the derivation of the mathematical model. But in many cases, it is even impossible. So in our method, the derivation may be derived from WNN ID model of the plant.

The block diagram of control system is shown in Fig. 2, where the WNN is used as the ID model.

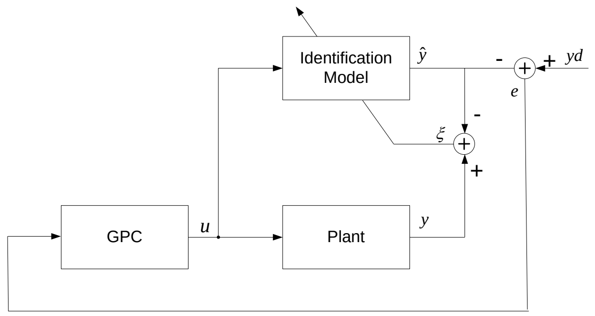

Fig. 2 Block diagram of control system.

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

$$J = \frac{1}{2}\left[E_{q,n}^{\ T} E_{q,n}\right], \qquad (19)$$

where $E_{q,n}$ is the predicted control error defined by

$$E_{q,n} = \left[e_{n+1} \quad e_{n+2} \quad \cdots \quad e_{n+q}\right]^T, \qquad (20)$$

$$e_{n+1} = yd_{n+1} - \hat{y}_{n+1}. \qquad (21)$$

Here, yd_{n+1} and $\hat{y}_{n+1}$ are the desired output signal and the WNN output signal, respectively. It can be seen that the controller is effected by the approximation performance of WNN. Therefore, it is necessary that $\hat{y}_{n+1}$ converges to the real system output y_{n+1} asymptotically. This convergence can be guaranteed via the on-line training of the WNN.

We define the following vectors:

$$R_{q,n} = \left[yd_{n+1} \quad yd_{n+2} \quad \cdots \quad yd_{n+q}\right]^T, \qquad (22)$$

$$\hat{Y}_{q,n} = \left[\hat{y}_{n+1} \quad \hat{y}_{n+2} \quad \cdots \quad \hat{y}_{n+q}\right]^T, \qquad (23)$$

$$U_{q,n} = \left[u_n \quad u_{n+1} \quad \cdots \quad u_{n+q-1}\right]^T. \qquad (24)$$

where q is the prediction horizon.

The purpose of the control is to find $U_{q,n}$ such that J is minimized. Using the gradient projection method, the control input U is updated by each iteration [13]:

$$\begin{aligned} U_{q,n+1} &= U_{q,n} + \eta \bar{P}_{q,n} \Gamma_{q,n} E_{q,n} \\ &= U_{q,n} + \eta \bar{P}_{q,n} \frac{\partial \hat{Y}_{q,n}}{\partial U_{q,n}} E_{q,n}, \end{aligned} \qquad (25)$$

where, $\bar{P}_{q,n}$ denoted by the projection matrix, is an $q \times q$ diagonal matrix with unity initial value $\bar{P}_{q,n_0} = 1$, and $\Gamma_{q,n}$ is the gradient of the output of the WNN with respect to $U_{q,n}$, which can be derived from the WNN model and can be easily evaluated.

If the predictive horizon is 3, $\Gamma_{q,n}$ is as follows:

$$\Gamma_{q,n} = \begin{bmatrix} \frac{\partial \hat{y}_{n+1}}{\partial u_n} & \frac{\partial \hat{y}_{n+2}}{\partial u_n} & \frac{\partial \hat{y}_{n+3}}{\partial u_n} \\ \frac{\partial \hat{y}_{n+1}}{\partial u_{n+1}} & \frac{\partial \hat{y}_{n+2}}{\partial u_{n+1}} & \frac{\partial \hat{y}_{n+3}}{\partial u_{n+1}} \\ \frac{\partial \hat{y}_{n+1}}{\partial u_{n+2}} & \frac{\partial \hat{y}_{n+2}}{\partial u_{n+2}} & \frac{\partial \hat{y}_{n+3}}{\partial u_{n+2}} \end{bmatrix}. \qquad (26)$$

Each individual element of the control sequence is updated by clipping the results obtained from Eq. (25) according to

$$\underline{\Delta u} \le \Delta u(n+i-1) \le \overline{\Delta u},$$

where $\Delta u(n+i-1) = u(n+i-1) - u(n+i-2)$ and each $\underline{\Delta u}$ and $\overline{\Delta u}$ can be heuristically chosen to be some very small values.

The projection matrix $\bar{P}_{q,n}$ is then updated according to $U_{q,n}$ at each iteration, by

$$\bar{P}_{q,n} = \begin{cases} 0 & if\ \underline{\Delta u} \le \Delta u(n+i-1) \le \overline{\Delta u} \\ \bar{P}_{q,n-1}(i,i) & otherwise \end{cases}, \ i = 1,2,\cdots,q. \qquad (24)$$

Finally, the first element of $U_{q,n}$, which is the new control sequence is applied to the system as the control signal.

In this paper, the input of the WNN in Eq. (6) is as follows:

$$\mathbf{X}_{n+1} = [u_n].$$

Therefore, Eq. (26) can be rewritten by the following equation:

$$\Gamma_{q,n} = \begin{bmatrix} \frac{\partial \hat{y}_{n+1}}{\partial u_n} & 0 & 0 \\ 0 & \frac{\partial \hat{y}_{n+2}}{\partial u_{n+1}} & 0 \\ 0 & 0 & \frac{\partial \hat{y}_{n+3}}{\partial u_{n+2}} \end{bmatrix}. \quad (27)$$

Here,

$$\frac{\partial \hat{y}_{n+1}}{\partial u_n} = \frac{\partial \hat{y}_{n+1}}{\partial \mathbf{X}_{n+1}} \frac{\partial \mathbf{X}_{n+1}}{\partial u_n} = \sum_{j=1}^{Nw} \frac{c_{j,n+1}}{d_{jk,n+1}} \frac{\partial \phi_j(\mathbf{X}_{n+1})}{\partial z_{jk,n+1}} \frac{\partial \mathbf{X}_{n+1}}{\partial u_n} + a_{k,n+1} \Big|_{k=1}, \quad (28)$$

$$\frac{\partial \hat{y}_{n+2}}{\partial u_{n+1}} = \frac{\partial \hat{y}_{n+2}}{\partial \mathbf{X}_{n+2}} \frac{\partial \mathbf{X}_{n+2}}{\partial u_{n+1}} = \sum_{j=1}^{Nw} \frac{c_{j,n+2}}{d_{jk,n+2}} \frac{\partial \phi_j(\mathbf{X}_{n+2})}{\partial z_{jk,n+2}} \frac{\partial \mathbf{X}_{n+2}}{\partial u_{n+1}} + a_{k,n+2} \Big|_{k=1}, \quad (29)$$

$$\frac{\partial \hat{y}_{n+3}}{\partial u_{n+2}} = \frac{\partial \hat{y}_{n+3}}{\partial \mathbf{X}_{n+3}} \frac{\partial \mathbf{X}_{n+3}}{\partial u_{n+2}} = \sum_{j=1}^{Nw} \frac{c_{j,n+3}}{d_{jk,n+3}} \frac{\partial \phi_j(\mathbf{X}_{n+3})}{\partial z_{jk,n+3}} \frac{\partial \mathbf{X}_{n+3}}{\partial u_{n+2}} + a_{k,n+3} \Big|_{k=1}. \quad (30)$$

4. Simulation and Result

4.1 The Duffing system

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

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} y \\ p_1 x - x^3 - p_2 y + b\cos(wt) + u \end{pmatrix}, \quad (31)$$

where typically, $p_1 = 1.1$, $p_2 = 0.4$, $b = 1.8$ and $w = 1.8$.

Duffing equation has either the periodic solution or the aperiodic (chaotic) solution depending on the value of b.

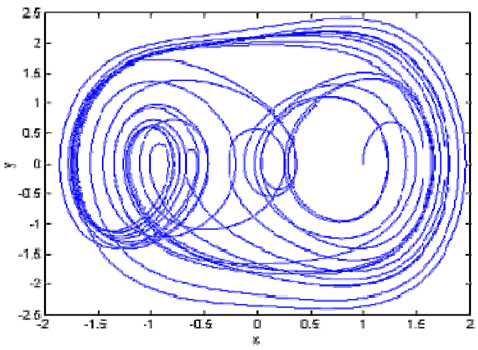

Fig. 3 The strange attractor of Duffing system.

4.2 The control of Duffing system

The desired signal is solution in case that the parameter b of Duffing equation is chosen as 2.3. The desired signal is as follows:

$$\begin{bmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \end{bmatrix} = \begin{bmatrix} y_r(t) \\ 1.1x_r(t) - x_r^{3}(t) - 0.4y_r(t) + 2.3\cos(1.8t) \end{bmatrix}. \quad (32)$$

Note that we can obtain the control result for Duffing system and the control input signal from simulation result (see Figs. 5 and 6). Fig. 4 represents the on-line ID result of the Duffing system. The mean square error (MSE) for system ID and control performance is indicated in Table 1.

The control input signals shown in Fig. 6 is not constant, but has a wave shape. This appearance is due to the cosine forcing term involved in the Duffing equation.

Table 1 Control environment and result.

Duffing system control	
Prediction horizon	3
Number of wavelet node	3
Sampling time	0.02
On-line ID error: MSE	0.0464
Control error: MSE	0.0915

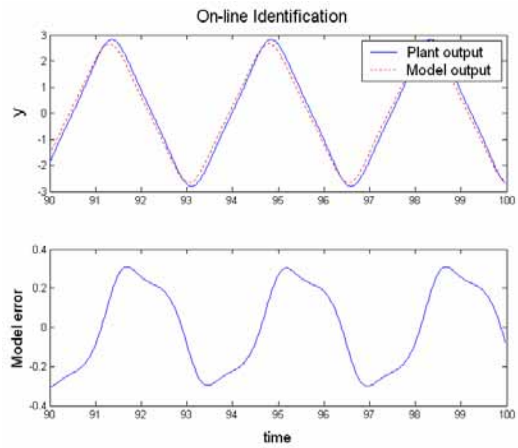

Fig. 4 The on-line ID of The Duffing system.

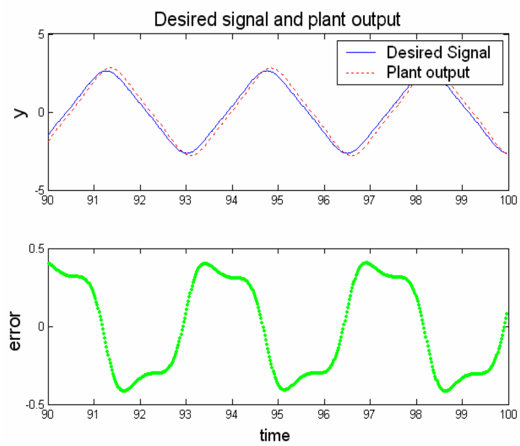

Fig. 5 The control result for Duffing system.

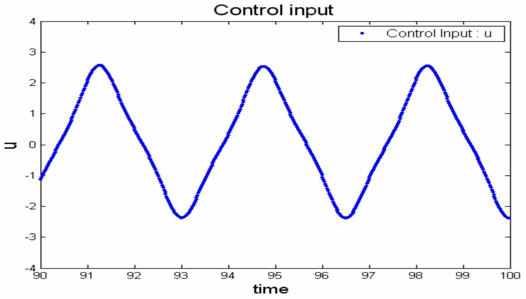

Fig. 6 The control input.

5. Conclusion

In this paper, we have presented the design method of a predictive controller based on a WNN, which was used to perform the multi-step prediction on-line, for the intelligent control of chaotic systems whose mathematical models are unknown.

Also we used the EKF training method for the training of WNN parameter. Since the WNN have the advantages of high resolution of wavelets, it have the fast convergence. Also the EKF training algorithm improved the performance of WNN. Finally, in order to evaluate the performance of our controller, the proposed method was applied to the Duffing system. The simulation results have shown that a WNN based predictive control scheme has the fast convergence property and accurate control performance.

References

[1] E. Ott, C. Grebogi, and J.A. Yorke, "Controlling Chaos," Phys. Rev. Lett. Vol. 64, No. 11, pp. 1196-1199, 1990.

[2] G. Chen and X. Dong, "On Feedback Control of Chaotic Nonlinear Dynamic Systems," Int. Jour. of Bifurcation and Chaos Vol. 2, No. 2, pp. 407-411, 1992.

[3] 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, 1993

[4] G. Chen and X. Dong, "From Chaos to Order-Perspectives and Methodologies in Controlling Chaotic Nonlinear Dynamical Systems," Int. Jour. of Bifurcation and Chaos, Vol. 3, No. 6, pp. 1363-1409, 1993.

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

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

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

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

[9] S. Singhal and L. Wu, "Training Feed-forward Networks with the Extended Kalman Algorithm," Proc. of ICASSP-89., Vol. 2, pp. 1187-1190, 1989.

[10] S. Haykin, *Neural networks – A comprehensive foundation*, Prentice-Hall, New York, 1994.

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

[12] D. D. Brengel and W. D. Seider, "Multistep Nonlinear Predictive Control," Ind. Chem. Eng. Res. 28, pp. 1812-1822, 1989.

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