2003 정보 및 제어 학술회의 (2003.11.21-22)

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

자기 회귀 웨이블릿 신경 회로망을 이용한 혼돈 시스템의 일반형 예측 제어

Sung Jin You*, Yoon Ho Choi**, Jin Bae Park*

* 연세대학교 전기전자공학과
** 경기대학교 전자공학부

Abstract : This paper proposes the generalized predictive control(GPC) method of chaotic systems using a self-recurrent wavelet neural network(SRWNN). The poposed SRWNN, a modified model of a wavelet neural network(WNN), has the attractive ability such as dynamic attractor, information storage for later use. Unlike a WNN, since the SRWNN has the mother wavelet layer which is composed of self-feedback neurons, mother wavelet nodes of the SRWNN can store the past information of the network. Thus the SRWNN can be used as a good tool for predicting the dynamic property of nonlinear dynamic systems. In our method, the gradient-descent(GD) method is used to train the SRWNN structure. Finally, the effectiveness and feasibility of the SRWNN based GPC is demonstrated with applications to a chaotic system.

Keywords : Self-recurrent wavelet neural network, Generalized predictive control, Gradient descent, Chaotic system, Chaos control

Ⅰ. Introduction

Recently the neural network(NN) had been shown to be of benefit for use in control of chaotic systems[1]. And a wavelet neural network (WNN), which absorbs the advantage of high resolution of wavelets and the structure of learning and feedforward of NN, is proposed to guarantee the fast convergence[2-3]. But both NN and WNN are a static mapping, and does not represent a dynamic system mapping without the aid of tapped delays. To solve this problem, we propose self-recurrent wavelet neural network (SRWNN), which combines the advantage of attractor dynamics and information storage for later use of recurrent neural network and the advantage of fast convergence of WNN. And we design the generalized predictive controller using the proposed SRWNN. In the learning method, We use the gradient-descent(GD) method to train the SRWNN structure. Finally, the effectiveness and feasibility of the SRWNN based GPC is demonstrated with applications to a chaotic system.

Ⅱ. SRWNN Identification for chaotic systems

1. SRWNN structure

This section discusses the structure of the SRWNN. A schematic diagram of the proposed SRWNN structure is shown in Fig. 1, which has N_i inputs, one output, and N_w wavelet nodes. The proposed SRWNN consists of four layers, which are input layer, mother wavelet layer, product layer and output layer. Each node of mother wavelet layer has a mother wavelet and a self-feedback loop. In this paper, we choose the first derivative of a Gaussian function, $\varphi(x)=-x\exp(-1/2x^2)$ as a mother wavelet. A wavelet φ_{jk} of each node is derived from its mother wavelet φ by

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

where, m_{jk} and d_{jk} are the translation factor and the dilation factor of the wavelets, respectively. The subscript jk indicates the kth input term of the jth wavelet. In addition, the inputs of this layer for discrete time n can be denoted by

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

where, θ_{jk} denotes the weight of the self- feedback loop. The input of this layer contains the memory terms $\varphi_{jk}(n-1)$, which store the past information of the network. This key aspect is the apparent different between a WNN and a SRWNN. Also, the structure of the SRWNN is the same that of WNN when $\theta_{jk}=0$. Thus we can see that SRWNN is a generalization system of the WNN.

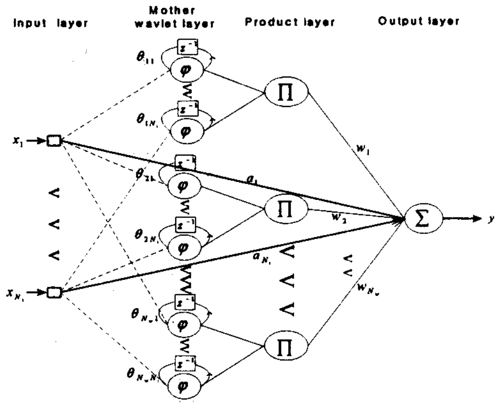

Fig. 1 Structure of the proposed SRWNN

The nodes in product layer are given by the product of the mother wavelet as follows:

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

Finally, the output of a SRWNN is composed by each self-recurrent wavelet and parameters as follows:

$$y(n)=\sum_{j=1}^{N_w} w_j \Phi_j(X)+\sum_{k=1}^{N_i} a_k x_k \quad (4)$$

where, w_j is connection weight between product nodes and output nodes, and a_j is connection weight between input nodes and output nodes. W is weighting vector of the SRWNN:

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

Here, the initial value of θ_{jk} is given 0. That is, there are no feedback units initially.

2. Identification of chaotic systems with SRWNN

This paper uses the series-parallel method for identification of a chaotic system. The identification model of a chaotic system consists of SRWNN and tapped delay lines. The current input and the most recent output of the system are fed into the SRWNN. And the error $e_f(n)$ between the actual system output and the SRWNN output is used to train the SRWNN. The current SRWNN output represents as follows:

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

where, N_s and N_e indicate the number of external inputs and input state variables, respectively. And $y_c(n)$ and $u(n)$ denote the chaotic system output and the identification input, respectively.

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

$$J_f(n)=\frac{1}{2}[y_c(n)-y(n)]^2=\frac{1}{2}e_f(n)^2 \quad (7)$$

where, $y_c(n)$ is the chaotic system output and $y(n)$ is the current output of SRWNN for the discrete time n. By using the gradient-descent method, weight values of the SRWNN are adjusted so that the error is minimized after a given number of training cycles. The gradient-descent method may be defined as:

$$\begin{aligned}W(n+1) &= W(n)+\Delta W(n) \\ &= W(n)+\eta_P\left(-\frac{\partial J_f(n)}{\partial W(n)}\right)\end{aligned} \quad (8)$$

where, η_P represents the learning rate of the SRWNN parameter.

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

$$\begin{aligned}\frac{\partial J_f(n)}{\partial W(n)} &= e_f(n)\frac{\partial e_f(n)}{\partial W(n)} \\ &= -e_f(n)\frac{\partial y(n)}{\partial W(n)}\end{aligned} \quad (9)$$

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

$$\frac{\partial y(n)}{\partial a_k(n)}=x_k \quad (10)$$

$$\frac{\partial y(n)}{\partial m_{jk}(n)}=\frac{w_j}{d_{jk}}\frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (11)$$

$$\frac{\partial y(n)}{\partial d_{jk}(n)}=\frac{w_j}{d_{jk}}z_{jk}\frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (12)$$

$$\frac{\partial y(n)}{\partial \theta_{jk}(n)}=\frac{w_j}{d_{jk}}\phi_j(n-1)\frac{\partial \Phi_j(X)}{\partial z_{jk}} \quad (13)$$

$$\frac{\partial y(n)}{\partial w_j(n)}=\Phi_j(X) \quad (14)$$

where,

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

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

III. SRWNN based generalized predictive control

In this section, we propose the generalized predictive control(GPC) system based on the SRWNN. We assume that the output data of the SRWNN are available on-line for the design and use of the controller. In our design method, an on-line system based on a SRWNN is employed and a nonlinear feedback controller with a predictive control scheme is implemented. The overall configuration of the SRWNN based GPC system is shown in Fig. 2, where the SRWNN output $y(n)$ is controlled to track the reference $r(n)$.

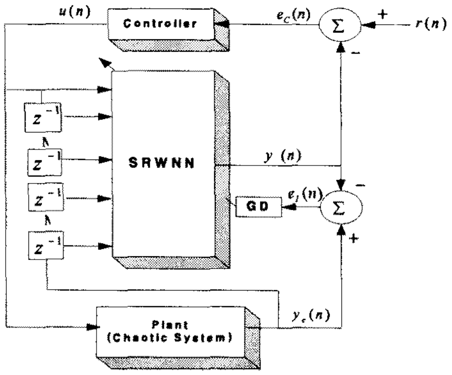

Fig. 2 Structure of GPC using SRWNN

Our propose is to find the optimal control input $u(n)$ in order to minimize the control error function as follows:

$$J_C(n+1) = \frac{1}{2}(e_C(n+1)^2 + \lambda \triangle u(n)^2) \tag{15}$$

where, $e_C(n) = r(n) - y(n)$. $\lambda \geq 0$ is the control increment weighting.

To minimize $J_C(n)$, the control input $u(n)$ is updated via the GD scheme:

$$u(n+1) = u(n) - \eta_U \frac{\partial J_C(n)}{\partial u(n)} \tag{16}$$

where η_U is the learning rate of the control input. We can see that the controller relies on the identification of the SRWNN. Thus to improve the controller performance, it is necessary that the SRWNN output well approaches the chaotic system output. In this aspect, the SRWNN, which has the information storage ability, is a suitable tool. Differentiating the result with respect to $u(n)$, it can be obtained that

$$\frac{\partial J_C}{\partial u(n)} = -e_C(n+1)\frac{\partial y(n+1)}{\partial u(n)} + \lambda \triangle u(n) \tag{17}$$

where $\partial y(n+1)/\partial u(n)$ can be analytically evaluated by SRWNN structure, Eqn. (4) as follows:

$$\begin{aligned}\frac{\partial y(n+1)}{\partial u(n)} &= \frac{\partial y(n+1)}{\partial X}\frac{\partial X}{\partial u(n)} \\ &= \left[\sum_{j=1}^{N_w} \frac{w_j}{d_{jk}}\frac{\partial \Phi_j(X)}{\partial z_{jk}} + a_k\right]_{k=N_i+1}\end{aligned} \tag{18}$$

where $\partial X/\partial u(n) = [0 \ 0 \ \ldots \ 1 \ 0 \ \ldots \ 0]^T$.

So far, we described the algorithm for one-step ahead predictive control scheme. Next, let this algorithm is extended by using the technique in GPC theory. Unlike ordinary predictive controls, a GPC uses a *receding horizon control* strategy[4]. The future values of reference signal and the chaotic output are needed to formulate the control signal. The SRWNN can be used to predict future values of the chaotic system. We define to be the same the control horizon N_u as the prediction horizon N_p. Using this definition, we can denote the following vectors:

$$R = [\, r_{n+1} \ r_{n+2} \ \ldots \ r_{n+N_p}]^T \tag{19}$$
$$Y = [\, y_{n+1} \ y_{n+2} \ \ldots \ y_{n+N_p}]^T \tag{20}$$
$$E = [\, e_{n+1} \ e_{n+2} \ \ldots \ e_{n+N_p}]^T \tag{21}$$
$$U = [\, u_{n+1} \ u_{n+2} \ \ldots \ u_{n+N_p}]^T \tag{22}$$

as the future values of the reference signal, the SRWNN output, the error vector between two vectors, and the control input vector. In addition, the control error function is defined as follows:

$$J = \frac{1}{2}[E^T E + \lambda \triangle U^2] \tag{23}$$

Using the gradient projection method, the control input vector U is updated in each iteration by

$$U = U + \eta_U EG + \lambda MU \tag{24}$$

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

$$G=\begin{bmatrix} \frac{\partial y_{n+1}}{\partial u_n} & 0 & \cdots & 0 \\ \frac{\partial y_{n+2}}{\partial u_n} & \frac{\partial y_{n+2}}{\partial u_{n+1}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots \\ \frac{\partial y_{n+N_p}}{\partial u_n} & \vdots & \cdots & \frac{\partial y_{n+N_p}}{\partial u_{n+N_{p-1}}} \end{bmatrix}.$$

where G is the gradient of the control error fuction with respect to U, which can be derived from the SRWNN and be easily evaluated.

Ⅳ. Simulation result

To visualize the validity of the proposed generalized predictive control scheme, we consider the Duffing system, which are the representative continuous-time chaotic system, and we compare the result of the proposed SRWNN based GPC with those of a WNN based GPC. The state equation of the Duffing system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}=\begin{bmatrix} x_2 \\ a_1x_1-x_1^3-a_2x_2+b\cos(wt)+u \end{bmatrix} \quad (25)$$

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

In tracking Duffing systems, we define the initial system states as (0, 0) and the sampling time is chosen as 0.05. Also the prediction horizon is 3. Reference signal is defined as one periodic solution in the case of b=2.3. The simulation environment and the performance measures are represented as shown in Table 1 and Table 2. And Fig. 3 shows the SRWNN based GPC result. From the results of the Table 2, we confirm that the SRWNN based GPC shows a better performance as compared with the WNN based GPC. Also we can see that the network structure of the proposed SRWNN is simpler than that of the WNN.

Ⅴ. Conclusion

This paper presents the self-recurrent wavelet neural network based GPC method for the chaotic nonlinear systems. Since the self-feedback units act as memory elements, the SRWNN has the capability to temporarily store information. The SRWNN is used to perform the on-line multi-steps prediction. And simulation results show that the SRWNN based GPC has a better performance as compared with the WNN based GPC.

Table 1 Comparison of the simulation environments

SRWNN (Our)	Number of mother wavelet	3
	Number of past state	2
	Number of past input	1
	ID learning rate	0.01
	Control learning rate	0.005
WNN	Number of mother wavelet	15
	Number of past state	2
	Number of past input	1
	ID learning rate	0.01
	Control learning rate	0.01

Table 2 Comparison of the performance

	ID MSE		Control MSE	
	SRWNN(our)	WNN	SRWNN(our)	WNN
x_1 state	0.010	0.062	0.070	0.222
x_2 state	0.0065	0.027	0.206	0.355

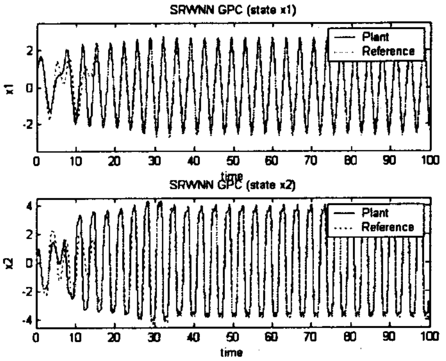

Fig. 3 The control results of Duffing system

Reference

[1] K. H. Oh, Y. H. Choi, J. B. Park and K. Y. Lim, ''Design of neural network controller for continuous-time chaotic nonlinear systems," Jour. of Institute of Electronics Engineers of Korea SC, Vol. 39, pp.51-65, 2002.

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

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

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