http://www.zju.edu.cn/jzus; E-mail: jzus@zju.edu.cn
ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.5, No.1, P.62-67, Jan., 2004

Hybrid internal model control and proportional control of chaotic dynamical systems*

QI Dong-lian(齐冬莲)†, YAO Liang-bin(姚良宾)

(*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: ldq0924@china.com.cn

Received Oct.8, 2002; revision accepted Jan.10, 2003

Abstract: A new chaos control method is proposed to take advantage of chaos or avoid it. The hybrid Internal Model Control and Proportional Control learning scheme are introduced. In order to gain the desired robust performance and ensure the system's stability, Adaptive Momentum Algorithms are also developed. Through properly designing the neural network plant model and neural network controller, the chaotic dynamical systems are controlled while the parameters of the BP neural network are modified. Taking the Lorenz chaotic system as example, the results show that chaotic dynamical systems can be stabilized at the desired orbits by this control strategy.

Key words: Chaos, Neural network, Internal model control, Proportional control
Document code: A **CLC Number:** TP273

INTRODUCTION

One of the oldest researches of chaos in control fields can be seen in the paper by Kalman (1956), who first found non-synchronous oscillations in a two-dimensional sampled-data control system. In the OGY method developed by Ott *et al*. (1990), chaotic phenomena was eliminated by adjusting parameters of the systems when the chaotic orbit comes near a periodical orbit. But the OGY's method requires monitoring the system long enough to determine a linearization mode in the neighborhood of the desired unstable periodic orbit before it can be applied. Besides, the determination of small perturbation requires the eigenvalues and eigenvectors of the unstable orbits. Since then, many methods for controlling chaos have been developed (He and Chen, 2002; Tong *et al*., 2002).

Along with the development of NN (Neural Networks), the chaotic behavior, one of the characteristics of NN, has been discussed in many papers. Alsing and Garielides (1994) used a feed-forward back propagation neural network to stabilize the unstable periodic orbits embedded in a chaotic system. Their control algorithm used for training the network was based on the OGY method, and thus inherited its deficiencies mentioned above. Another approach for controlling chaos with a neural network was proposed by Otaware and Fan (1995), who used the same network structure as Alsing's. The above two neural network approaches are both supervised learning methods, in which a feedforward multiplayer neural network is trained by data pairs generated from a chaotic system to produce a time series of small perturbations necessary for control. The disadvantage is that the fixed points of the chaotic system need to be determined; and/or that the system's nonlinear dynamics must be analyzed in advance. However, since the application of neural network controllers to chaotic systems yields great benefits, we adopted a systematic approach for designing a neural network controller for controlling chaos by the hybrid Internal Model Control learning method. In this paper, we adopt proportional control method and the IMC (Internal Model Control) method of NN, which are new methods for controlling chaotic dynamical systems.

* Project supported by the National Natural Science Foundation of China (No.60374013), and Doctoral Subject Research Fund of Ministry of Education of China (No.1999033512), and Science and Technology Plan Fund of Zhejiang Province (No.991110412)

INTERNAL MODEL CONTROL THEORY

Internal Model Control (Fig.1) is a kind of supervised learning approach. It consists of controller, plant model, and robustness filter. The controller is generally trained to represent the inverse of the plant, if the inverse exists. Many researchers had proved the good performance of the Internal Model Control method. It is an important nonlinear controlling approach to system and had been discussed by many researchers (Datta and Choa, 1996).

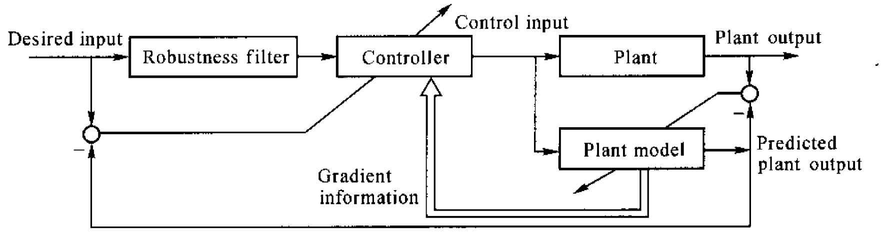

Fig.1 The structure of the Internal Model Control learning system

The plant model and the controller can be trained off-line, using data collected from plant operations. The weighted coefficients of the plant model are modified according to the errors between the outputs of the inverse model and the plant output. During the modifying process, the plant model replaces the unknown plants. If the plant model and controller both employ multiplayer feed-forward neural network, the error used to modify the parameters of the NN inverse model, is back-propagated via the neural network plant model. The function of plant model is just to back-propagate the error. This may bring a little error, but the effect of error is somewhat light. Generally speaking, it has an impact only on the controller's convergence rate, not on the final convergence precision. IMC learning method is substantially a predicting control method where the plant model and inverse model are linked by the gradient information.

HIMC LEARING CONTROL SYSTEM

Structure of the HIMC learning system

Fig.2 is the proposed HIMC scheme. The structure of HIMC includes the following parts: learning controller made up of NN controller and Proportional controller, NN model used to approximate the chaotic dynamical system, and the chaotic system.

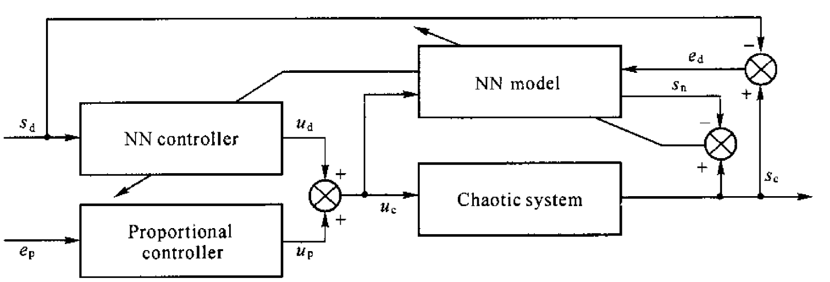

Fig.2 The proposed learning system for controlling chaos

In order to have the desired robust performance and ensure the system's stability, the proportional control is used in the HIMC learning system instead of the robustness filter. During the control processing, the error signal between the neural network plant model and the plant passes by the proportional controller first, and then the NN controller processes it.

BP network is used in the system to form neural network controller and plant model. The input of the controller is s_d where u_d is the output. u_c is the input of the neural network plant model where S_n is the output.

For BP network, the transfer function of the hidden layer is a sigmoid activating function and the function of the output layer is a linear function.

In Fig. 2, s_d is the desired output state of the next step; s_c is the current output state of the chaotic system; u_c is the input of the chaotic system and the neural network plant model where $u_c = u_d + u_p$; The input of the Proportional Controller is $e_p = y_d(k) - y(k)$; The difference between the next-step desired output and the current output state is $e_d = y_d(k+1) - y(k)$; The output of Proportional Controller is u_p; The output of the neural network controller is u_d.

Neural network plant model of the chaotic systems

Consider the chaotic dynamical systems

$$y(t) = f(y(t-1), \cdots, y(t-n), u(t-1), \cdots, u(t-m)) + \delta(t) \quad (1)$$

Its discrete time equation of the system can be expressed as:

$$y(k+1) = f(\phi(k), u(k)) \quad (2)$$

where $u(\cdot)$ are the inputs, $y(\cdot)$ are the outputs and $\delta(\cdot)$ are the noise of the system, $f(\cdot)$ is a nonlinear function describing the system, and $\phi(k)$ is a nonlinear function made up of $\{y(k-i), i = 1, 2, \cdots, n\}$ and $\{u(k-i), i = 1, 2, \cdots, m\}$.

We have known that a three-layer BP neural network with sigmoid function in the hidden layer and linear transfer functions in the output layer, can approximate virtually any function of interest to any degree of accuracy if we had a sufficient neuron number in the hidden layer. Therefore, the neural network plant model *NN* is trained on-line to approximate the nonlinear dynamic input-output behavior of the chaotic system.

Suppose that the three-layer neural network plant model is described by

$$\hat{y}(k+1) = \hat{f}(\hat{\phi}(k), u(k)) \quad (3)$$

where $u(\cdot)$ are the inputs, $\hat{y}(\cdot)$ are the outputs and $\hat{f}(\cdot)$ is a nonlinear function of the neural network plant model and $\hat{\phi}(\cdot)$ is a nonlinear function.

In order to accelerate the learning rate, avoid falling into the local extreme minimum values and improve system performances, the parameters (weights) of neural network plant model are adapted by additional momentum algorithm to minimize the instantaneous cost function.

$$J_1 = \frac{1}{2}[\hat{y}(k+1) - y(k+1)]^2 \quad (4)$$

Additional momentum method is used to modify the networks weights taking into consideration of the error variance influence. Without the additional momentum, the networks may fall into local extreme minimum values; which, however, can be avoided by the use of additional momentum.

The instantaneous correction applied to a synaptic weight is

$$\delta(k) = [y(k) - \hat{y}(k)]g'[net_{3i}(k)] \quad (5)$$

$$\Delta w_{3il}(k+1) = (1 - mc)\eta\delta(k)o_{2l}(k) + mc \cdot \Delta w_{3il}(k) \quad l = 1, 2, \cdots, m_1 \quad (6)$$

$$\Delta w_{2lj}(k+1) = (1 - mc) \cdot \eta g'[net_{2l}(k)]\delta(k) \cdot w_{3il}(k) + mc \cdot \Delta w_{2lj}(k) \quad l = 1, 2, \cdots, m_2 \quad (7)$$

where w_{2lj} (or w_{3il}) are the weights connecting input (or neuron) j to hidden neuron (or output) i; g is the sigmoid function; o_{ij} is the output of hidden neurons or output neurons; net_{2l} (or net_{3i}) is the input of the hidden neurons or output neurons; $g' = \frac{\partial g(x)}{\partial x}$; η is the learning rate; mc is the additional momentum coefficient usually having a value of about 0.95.

Based on the design principle of the momentum method, when the updated weights lead to too much increment, the new weights should not be employed. The value of the momentum coefficient is

$$mc(k) = \begin{cases} 0 & SSE(k) > 1.04SSE(k-1) \\ 0.95 & SSE(k) < SSE(k-1) \\ mc(k-1) & \text{others} \end{cases} \quad (8)$$

where $SSE(k) = \frac{1}{2}\sum_{k=1}^{m_2}(\hat{y}_k - y_k)^2$; m_2 is the number of neurons in the output layer.

The formula for the adaptive learning rate during the process of system identification and learning is as follows

$$\eta(k)=\begin{cases}1.05\eta(k-1) & SSE(k)<SSE(k-1)\\ 0.7\eta(k-1) & SSE(k)>1.04SSE(k-1)\\ \eta(k-1) & \text{others}\end{cases} \quad (9)$$

Chaotic dynamical system is a kind of special nonlinear system. For chaotic dynamical system, modifying the parameters of any three-layer BP neural network can approximate its input and output function.

Neural controller of the chaotic systems

The neural network controller, like the neural network plant model, is also a three-layer BP network, whose general expression is

$$u(k)=f[y_d(k+1), y_d(k+2), \cdots, y_d(k+p), u(k-1), \cdots, u(k-q), e(k)] \quad (10)$$

where y_d is the desired output of the chaotic system.

Controller weights are adapted to minimize the cost function

$$J_2=\frac{1}{2}[y_d(k+1)-y(k+1)]^2 \quad (11)$$

The instantaneous correction applied to a synaptic weight is

$$\delta(k)=[y_d(k)-y(k)]g'[net_{3i}(k)] \quad (12)$$

$$\Delta w_{3il}(k+1)=(1-mc)\cdot\eta\delta(k+1)o_{2l}(k)\cdot\frac{\partial y(k+1)}{\partial u(k)}+mc\cdot\Delta w_{3il}(k) \quad (13)$$

$$\Delta w_{2lj}(k+1)=(1-mc)\cdot\eta g'[net_{2l}(k)]\delta(k)\cdot w_{3il}(k)o_{ij}(k)\frac{\partial y(k+1)}{\partial u(k)}+mc\cdot\Delta w_{2lj}(k)$$
$$l=1,2,\cdots,n_2;\ j=1,2,\cdots,n_1 \quad (14)$$

where w_{2lj} (or w_{3il}) are the weights connecting input (or neuron) j to hidden neuron (or output) i. g is the sigmoid function; η is the learning rate; o_{ij} is the output of hidden neurons or output neurons; net_{2l} (or net_{3i}) is the input of the hidden neurons or output neurons; $g'=\frac{\partial g(x)}{\partial x}$; n_1 is the number of input neurons while n_2 is the number of hidden neurons. The momentum coefficient (mc) and learning rate η can be obtained from Eq.(8) and Eq.(9) where $SSE(k)=\frac{1}{2}\sum_{k=1}^{m_2}(y_{dk}-y_k)^2$.

In Eq.(13) and Eq.(14), $\frac{\partial y(k+1)}{\partial u(k)}$ is unknown because the plant is unconcern. Assuming a good estimation of y, we have

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)}\approx\frac{\partial y(k+1)}{\partial u(k)} \quad (15)$$

According to Liu and Asada (1996), $\hat{y}$ can approximate virtually to y after several learning steps of the IMC learning system. Therefore

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)}=g'[net_{3i}(k)].$$
$$\sum_{l=1}^{m_2} w_{3il}g'[net_{2l}(k)]w_{2l,n+1} \quad (16)$$

From Eq.(12) to Eq.(16), we can get the parameters of the neural network controller.

The closed-loop system output equation of the NN self-correction Internal Model Controller can be expressed as follows

$$y(k)=\frac{NNC\cdot G}{1+NNC[G-NN]}y_d(k) \quad (17)$$

where G is the plant, NN is neural network plant model and NNC is neural network controller.

The error equation of the closed-loop system is

$$E(k)=\frac{1-NNC\cdot NN}{1+NNC[G-NN]}y_d(k) \quad (18)$$

If $NNC(1)=NN^{-1}(1)$, for the step input, steady state error $E(\infty)$ is zero; that is, the disturbance can be eliminated, and the system can track the input signal.

Proportional controller

In order to minimize the error of the inverse model and improve the controlling performance, the Proportional Controller is used in the learning system.

The input of the Proportional Controller is $e_p=y_d(k)-y(k)$ and the output is u_p, that is $u_p=k_p\cdot e_p$.

The IMC method, based on an internal model and plant inversion, can only work for the limited class of stable, stably invertible plants. But the NN inverse model is not precise because chaotic system is not a kind of single-value system. There will be an error if neural network controller is used in the system only. Proportional Controller u_p, as a part of controller, controls the learning system with neural network controller, and at the same time, minimizes the error of the inverse model of chaotic system.

SIMULATIONS AND RESULTS

For convenience to analyze and design the control system (Lorenz, 1963), suppose that the Lorenz system is the plant. The differential equation of the chaotic model can be described by

$$\dot{x}_1 = -\sigma(x_1 - x_2)$$
$$\dot{x}_2 = \rho x_1 - x_2 - x_1 x_3$$
$$\dot{x}_3 = x_1 x_2 - b x_3$$

where x_1, x_2, $x_3 \in R$ and $\sigma, \rho, b \in R$, $\{x_1, x_2, x_3\}$ are the state variables and $\{\sigma, \rho, b\}$ are control parameters of the Lorenz system. When control parameters $\sigma = 10$, $\rho = 28$ and $b = \frac{8}{3}$, the Lorenz system is chaos.

Add controller u into the first equation of the Lorenz system and let state variable x_1 be the system output. The operation of the control system is made up of the response of the BP network, the control process and the training of the BP network, and it repeatedly alternate in the order "the response of the BP network, control determining, the response and training of the system".

When the system starts operating, all the parameters are set randomly. At this point, Proportional Controller controls the system. During the continuous learning of the network, the BP network continuously modifies its weights and the network output u_d approximates the control vector u_c, i.e. $u(k) = u_c \cong u_d$. Then the NN controller can be regarded as approximating the inverse model of the chaotic dynamical system.

Suppose the desired output of the system is a square wave. Fig.3 is the output result. In Fig. 3, the broken line indexes the ideal output (maximum $x_{1dmax} = 1$, minimum $x_{1dmin} = 0$) and the solid line indexes the actual output of the system. Fig.4 is the curve of the control signal u_c.

Fig.5 and Fig.6 are the output results when the desired output of the chaotic system is a sine

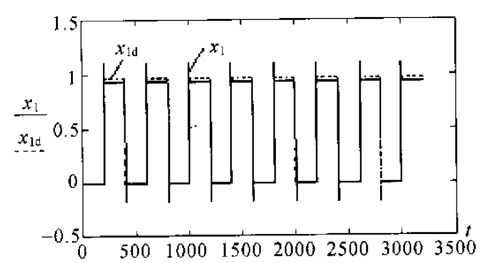

Fig.3 The actual output tracking the square wave

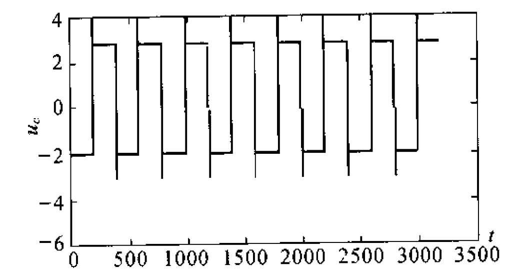

Fig.4 The control curve when the desived output is a square wave

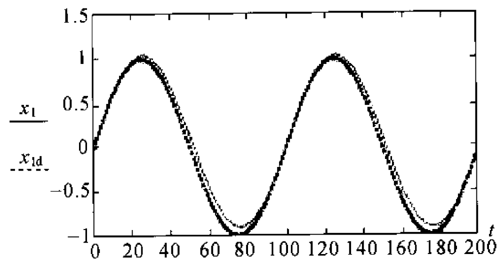

Fig.5 The real output tracking the sine wave

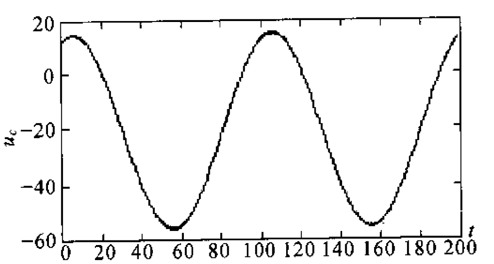

Fig.6 The output of the controller when the desived output is a sine wave

wave. In Fig.5, the solid line is the actual output and the dashed line is the desired output. Fig.6 is the control signal u_c. From Fig.3 and Fig.5, we can find that the whole control system can still run regularly, but there is a time lag between the actual output and the desired output when the desired output signal is a square wave or sine wave. In the tolerable error range, the actual output of the system can track the desired output closely.

CONCLUSIONS

This paper integrates the Proportional Control into IMC to form a new hybrid Internal Model Control learning system, called HIMC learning system. By the HIMC learning system, we can train the neural controller for the plant according to the plant output. The proposed HIMC learning method makes the design of neural controllers more feasible and practical for chaotic dynamical system, since it greatly decreases the quality and quantity expectations of the teaching signals, and shortens the long training time of the neural network. The HIMC learning system has been successfully applied to control Lorenz system. The simulation results showed that the ability of the BP network to fit the nonlinear mapping is very strong, and the learning rate is very fast. However, the difficulty is how to comprehend the error back propagation method, and determine the number of the hidden-layer neurons. Since the HIMC learning system need not know the mathematical model or the property (such as the fixed points) of the chaotic system for control, it can be applied to control a physical chaotic system in the real world directly.

References

Alsing, P.M. and Garielides, A., 1994. Using neural networks for controlling chaos. *Phys. Rev. E.*, **49**: 1225 – 1231.

Datta, A. and Choa, J.O., 1996. Adaptive internal model control design and stability analysis. *Automatics*, **32** (2): 261 – 266.

He, B. and Chen, Y., 2002. Using chaos to improve measurement precision. *Journal of Zhejiang University SCIENCE*, **3**(1): 47 – 51.

Kalman, R.E., 1956. Nonlinear aspect of sampled-data control systems. Proc. Nonlinear Circuit Analysis, p. 273 – 313.

Liu, S. and Asada, H., 1996. Teaching and learning of debarring robots using neural networks. Neural Networks Applications, IEEE Technology Update Series.

Lorenz, E.N., 1963. Deterministic non-periodic flows. *J. Atmos. Sci.*, **20**: 130 – 141.

Otawara, K. and Fan, L.T., 1995. Controlling chaos with an artificial neural network. *Proc. IEEE Int. Joint Conf. Fuzzy Sys.*, **4**: 1943 – 1948.

Ott, E., Grebogi, C. and York, J.A., 1990. Controlling chaos. *Phys. Rev. Lett.*, **64**: 1196 – 1199.

Tong, Q.Y., Fan, Y.L. and Li, Y., 2002. Application of chaotic theory to parameter estimation. *Journal of Zhejiang University SCIENCE*, **3**(1): 42 – 46.