Model Adaptive Control Based on a Compound Orthogonal Neural Network

Bingjun Guo[1], and Jinshou Yu[2]

[1] Petrochemical College, East China University of Science and Technology, Shanghai, China, 201512, Email: bingjunguo@sina.com
[2] Information College, East China University of Science and Technology, Shanghai, China, 200237, Email: jshyu@ecust.edu.cn

Abstract

Combining the adaptive inverted control method based on a compound orthogonal neural network with generic mode control scheme, an adaptive control algorithm based on a compound orthogonal neural network has been proposed, which can embed the process model into the controller by the inverted control method with neural networks. It can guarantee the realizability of the generic model control scheme based on neural networks. The reference trajectory is a pseudo-second-order curve. As the compound orthogonal neural network is easy to implement and very fast in convergence speed, it is suitable to apply in real control system. It is also very easy to tune for the controller. The simulation results show the effectiveness of the proposed control scheme.
Keywords. Generic model control, Compound orthogonal neural network (CONN), Pseudo-second-order system, Adaptive inverted control

I. Introduction

An early method [1] was to apply the nonlinear model in control law and it needed to calculate the numerical inversion of the complicated process model. Generic model control (GMC) [2][3][4] algorithm is an effective nonlinear control one, which can directly embed the nonlinear process model into the controller. The reference trajectory is a pseudo second order curve. It is easy to tune the controller. But it is very important that it can calculate the control law according to the first order differential equation of the controlled plant in GMC. Sometime it is very difficult to calculate the control law according to the first order differential equation in GMC. In order to overcome the weakness, an effective method is to apply neural networks to get the inverse model for the controlled plant in adaptive control. Generally BP and RBF neural networks have been widely applied in the inverse control system [9][10]. But BP neural network is of some weaknesses, for example, it is a complicated algorithm, slow learning speed and easy to get a local minimum value. RBF neural network is also of complicated algorithm. They are not very suitable in real time control system. The compound orthogonal neural network is a simple network and it is of a high-speed convergence of learning process [8]. Here we propose a generic model adaptive control scheme based on neural networks with the method of inverse system for the nonlinear process to directly embed the process model into the controller. It can guarantee the realizability of GMC. This paper is arranged as follows: first simply introduce generic model control, the inverse system method of the nonlinear process model, then explain a generic model adaptive control scheme based on a compound orthogonal neural network, finally show the effectiveness of the control scheme by simulation results.

II. Generic Model Adaptive Control Based on a Compound Orthogonal Neural Network

A. Generic model control (GMC)

We consider a nonlinear system described by differential equations of the type:

$$\dot{y} = f(y, u, d, t, \theta) \tag{1}$$

where u is input vector of dimension m, d the disturbance vector of suitable dimension, y the output vector of suitable dimension and θ the parameter vector of dimension n. According to the GMC basic principle [2], we develop the control algorithm, which consists of three terms (dynamic process model, proportional action term and integral action term, respectively), from

$$f(y,u,d,t,\theta) - K_1(y_{sp} - y) - K_2\int_0^t (y_{sp} - y)dt = 0 \tag{2}$$

where y_{sp} is a set point, K_1 and K_2 are parameter diagonal matrices of dimension n×n. Then the generic model control structure is shown in Figure 1.

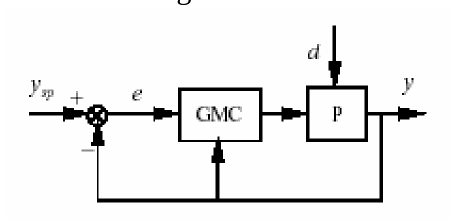

Fig.1. Generic model control structure

The control algorithm in Eq. 2 is generally implicit. Here it is solved on-line by some iterative numerical methods. If $f(y,u,d,t,\theta)$ is linear with respect to u, e.g. $f(y,u,d,t,\theta) = h(y,d,t,\theta)u$, then Eq. 2 becomes an explicit algorithm

$$u = (h(y,d,t,\theta))^{-1}(K_1(y_{sp} - y) + K_2\int_0^t (y_{sp} - y)dt) \tag{3}$$

Eq. 3 is a continuous form of GMC. In order to apply to the discrete system, the integral item must be represented by the discrete form. The discrete form of GMC is:

$$u_k = (h(y_k, d_k, t, \theta))^{-1}(K_1(y_{sp} - y_k) + K_2\sum_0^k (y_{sp} - y_k)\Delta t) \tag{4}$$

K_1 and K_2 can be calculated as follows:

$$K_{1(i,i)} = \frac{2\xi_i}{\tau_i} \qquad K_{2(i,i)} = \frac{1}{\tau_i^2} \tag{5}$$

where ξ_i and τ_i determine the shape and speed of the desired closed loop trajectory (the reference trajectory), respectively. The reference trajectory for a step change in the set point has a pseudo-second-order response. Yamuna et al [4] showed that the formulae can be used to accurately calculate the specified response for any values of ξ_i and τ_i. When the values of ξ_i and τ_i, which correspond to the desired specified response, are selected, and K_1 and K_2 can be calculated by Eq. 4. It is important to calculate the control law according to the first order differential equation in GMC. In order to overcome the weakness, we propose a generic model control scheme based on neural networks with the method of inverse system for the nonlinear process to directly embed the process model into the controller. It can guarantee the realizability of GMC and then it can expand GMC to the neural networks field.

B. Generic model control based on a neural network

The GMC scheme based on a neural network is shown in Figure 2. Here P is the controlled plant, d the internal disturbances, NN_c the neural network controller, I the integral controller, y_{sp} the set point for the system and y the system output. PI is a proportional-integral controller as shown in Figure 2.

$$PI(z) = K_1 + K_2 I \tag{6}$$

where I is an integral controller.

Obviously, the impulse transfer function of the closed loop system for SISO is:

$$\frac{y(z)}{y_{sp}(z)} = \frac{K_2 I^2 NN_c P + K_1 INN_c P}{K_2 I^2 NN_c P + K_1 INN_c P + 1} \tag{7}$$

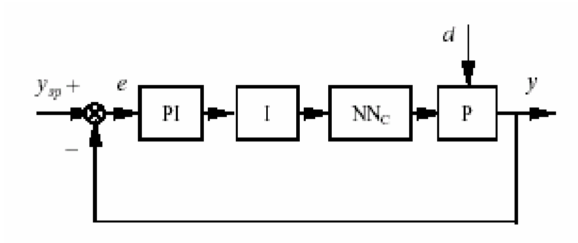

Fig.2. Generic model control based on a neural network

So long as we design the neural network controller NN_c to achieve the inverse control for the controlled plant, we can achieve the generic model control based on neural networks. Obviously, in the perfect model, the impulse transfer function of the closed loop system is:

$$\frac{y(z)}{y_{sp}(z)} = \frac{K_1 Tz + K_2 T^2 - K_1 T}{z^2 + (K_1 T - 2)z + K_2 T^2 - K_1 T + 1} \tag{8}$$

It is a discrete form for pseudo-second-order system. Therefore, It is very easy to tune for the controller and calculate the control law.

C. Inverse realization of GMC and controller design

If we can design the neural network controller NN_c to achieve the inverse control for the controlled plant, we can achieve generic model control based on neural networks. So its key is how to achieve the inverse model for the nonlinear controlled plant. Here, we apply NHARMA equation 9 to describe the n order nonlinear plant P

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

where $u \in R$ is input variable, $y \in R$ output variable, f the nonlinear reflection, and m, n is the order for the time series of input and output for $m \le n$. For the subset $A \in R^{m+n+-1}$, if there is any one element $[y(k), y(k-1), \cdots, y(k-n+1);\ u(k), u(k-1), \cdots, u(k-m+1)]$ for any two different inputs $u_1(k)$ and $u_2(k)$, there is the following expression: $f[y(k), y(k-1), \cdots, y(k-n+1); u_1(k), u(k-1), \cdots, u(k-m+1)] \neq f[y(k), y(k-1), \cdots,$ $y(k-n+1); u_2(k), u(k-1), \cdots, u(k-m+1)]$, we define the system Eq. 9 in the point $[y(k), y(k-1), \cdots, y(k-n+1);, u(k-1), \cdots, u(k-m+1)]^T$ is inversable. Here, we assume that the controlled plant is inversable.

For the inverse model of the controlled plant, we can directly identify the controlled plant by neural networks. We assume the input vector for neural network is:

$$IN = [y(k+1), y(k), \cdots, y(k-n+1); u(k-1), \cdots, u(k-m+1)] \tag{10}$$

Then the input $u(k)$ of the controlled plant is the output of the neural network controller NN_c. Because the future output $y(k+1)$ of the system can be achieved, when the neural network is trained, it becomes the input of the neural network. Usually, the output $\bar{u}(k)$ of the neural network is not the same as $u(k)$ for the training data. So the inverse neural network model describing the dynamic characteristics of the controlled plant is:

$$\bar{u}(k) = f_1[y(k+1), y(k), \cdots, y(k-n+1); u(k-1), \cdots, u(k-m+1)] \quad (11)$$

where f_1 is the reflection of the neural network controller NN_c.

In the real-time control of a nonlinear system, the computational load is a very important factor that affects the stability and dynamic performances of the closed-loop system. Current neural networks usually use BP or RBF network in the adaptive control system. But BP or RBF network's algorithm is complex. Using the compound orthogonal neural network to approximate the nonlinear behavior of a system can reduce the learning process of the neural network effectively. We apply the compound orthogonal neural network (CONN) to construct the inverse model of the controlled plant, then to form the generic model adaptive control based on CONN.

D. CONN approximation for the inverse model of the controlled plant

CONN consists of three layers: an input layer, a hidden layer and an output layer as shown in Figure 3. There is only relationship between neighborhood two layers and the connection weight between the input layer and the hidden layer is 1. The connection weight between the hidden layer and the output layer is w_j ($j=1,2,\cdots,g$) and g is the number of the neuron of the hidden layer. The input vector of CONN is $R_k = [x_1, x_2, \cdots, x_n]^T$. The output of the controlled plant is y and the set point is r. Then the output of the neural network is:

$$u_k = \sum_{j=1}^{g} w_j p_j \quad (12)$$

The p_j neuron of the hidden layer is Chebyshev compound orthogonal polynomial, that is, $p_1 = 1$, $p_2 = X$, $p_j = 2Xp_{j-1} - p_{j-2}$, $j = 3,4,\cdots,g$. Here we assume X as a unipolar sigmoid function:

$$X = \frac{1}{1+e^{-\sigma.net}} \quad (13)$$

where $net = \sum_{i=1}^{n} x_i$. We can shift the input from the region $[-\infty, +\infty]$ to $[0,1]$ with the unipolar sigmoid function and change the gradient of the unipolar sigmoid function through changing the parameter σ. Then we can change the learning adaptive performance and convergence of CONN. According to the reference trajectory or the learning signals we can get a learning algorithm for the parameter σ. The output error of CONN is:

$$E_k = r(k-1) - y(k) \quad (14)$$

The performance index of the network is:

$$J = E_k^2 \quad (15)$$

The parameter learning algorithm for the connection weight is:

$$W_{k+1} = W_k + \eta_1 E_k P \quad (16)$$

where $W_k = \lfloor w_1, w_2, \cdots, w_g \rfloor$, $P = \lfloor p_1, p_2, \cdots, p_g \rfloor$, η_1 is a learning ratio for the connection weight and $0 < \eta_1 < 1$.

The parameter learning algorithm for the sigmoid function is:

$$\sigma_{k+1} = \sigma_k + \eta_2 E_k p_2 \quad (17)$$

where η_2 is a learning ratio for the parameter σ and $0 < \eta_2 < 1$.

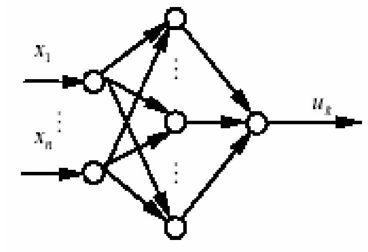

Fig.3. The neural network structure

E. The generic model adaptive control based on CONN

According to Figure 2, we can achieve the generic model adaptive control scheme based on CONN (simply, called as GMAC) as shown in Figure 4. Here P is the controlled plant, d the internal disturbances, NN_c the neural network controller, I the integral controller, y_{sp} the set point for the system and y the system output. e is the error of the controlled system, NN_{PI} is a proportional-integral controller or a single neuron PI controller as shown in Figure 4. TDL in Figure 4 is the multi-component time delay system as shown in Figure 5. The outputs consist of the time delay signals for the input as follows:

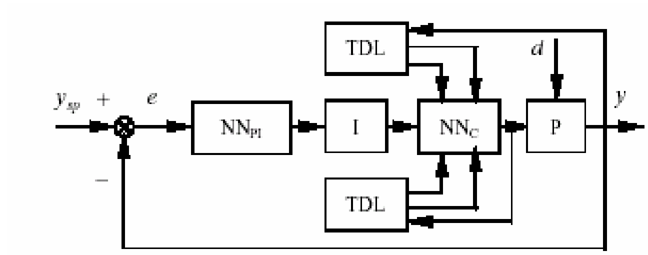

Fig.4. The generic model adaptive control scheme based on CONN

$$z(t) = [x(t-1), \cdots, x(t-n)]^T \tag{18}$$

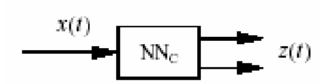

Fig.5. Multi-point time delay system

The design steps for the generic model adaptive control scheme based on CONN are as follows:

Step1: according to control performance, select the parameters K_1 and K_2 of GMC;

Step2: initialize the parameter σ and the weight W with small random numbers;

Step3: input a set of data and calculate the neuron output according to the old weight W ;

Step4; calculate the output and the output error according to the formulas 12~15;

Step5: calculate W_{k+1} and σ_{k+1} according to the updating formulas of the weight W_{k+1} and the parameter σ_{k+1};

Step6: again input a set of data and go to Step 2.

II. Simulations

In order to verify the effectiveness of the proposed control scheme, we execute a series of simulation experiments. The nonlinear controlled plant is:

$$\dot{y} = ay^2 + bu + cuy \tag{19}$$

that has been modeled as:

$$\dot{y}_m = \hat{a}y^2 + \hat{b}u + \hat{c}uy \tag{20}$$

where $\hat{a} = -0.25$, $\hat{b} = 0.5$ and $\hat{c} = 1.75$ are perfect model parameters.
The structure of the neural network in the adaptive control system is $N_{4\times4\times1}$. The input signals of the neural network are $R_k = [v(k), y(k), y(k-1), u(k)]^T$. The learning ratio is selected with trial and error approach as $\eta_1 = 0.006$ and $\eta_2 = 0.01$. The control system will reduce the control error through updating the weight W and the parameter σ in every sampling interval according to the parameter learning algorithms.

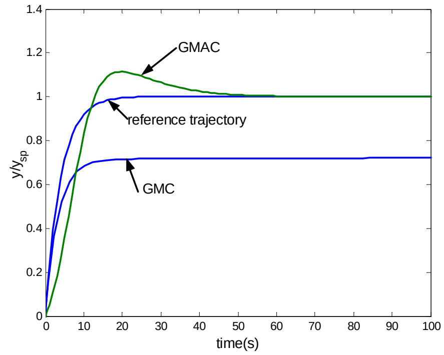

Fig. 6. Overdamped responses with parameter reduction in c

In order to compare the control performance, we respectively execute the simulation experiments according to the generic model adaptive control scheme based on CONN (GMAC) and the generic model control scheme (GMC) for the nonlinear plant. The controlled system is worked out with overdamped and underdamped reference trajectories: $K_1 = 0.25$ and $K_2 = 0.0001$ for overdamped reference trajectory and $K_1 = 0.25$ and $K_2 = 0.1$ for underdamped reference trajectory. Suppose that there is –50% parameter mismatch in c: $a = \hat{a}$, $b = \hat{b}$ and $c = 0.5\hat{c}$. The simulation results are shown in Figures 6 and 7. When there is +80% parameter mismatch in c: $a = \hat{a}$, $b = \hat{b}$ and $c = 1.8\hat{c}$. The simulation results are shown in Figures 8 and 9.

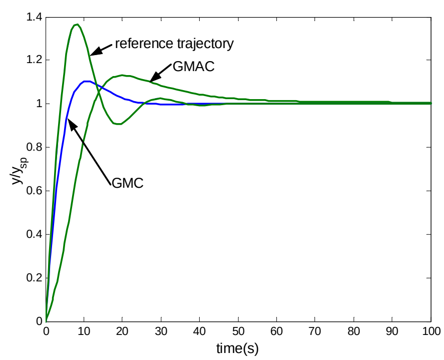

Fig. 7. Underdamped responses with parameter reduction in c

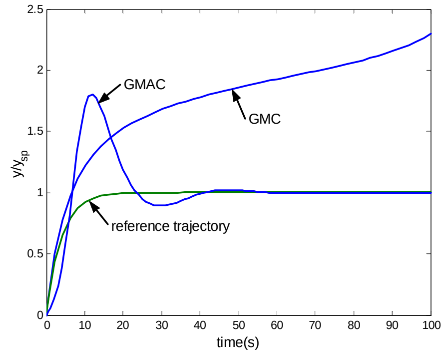

Fig. 8. Overdamped responses with parameter increase in c

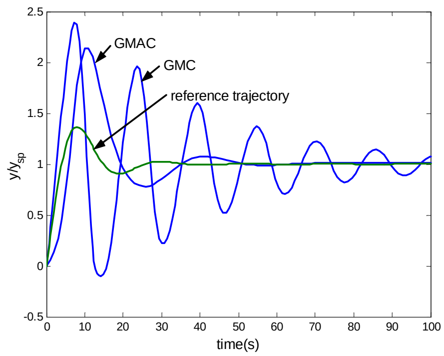

Fig. 9. Underdamped responses with parameter increase in c

The nonlinear process, Eq. 19, has been approximated as a linear model:

$$\dot{y}_m = \hat{a}y + \hat{b}u \tag{21}$$

with $\hat{a} = 0.1764$ and $\hat{b} = 0.4225$. There is a structural mismatch between the process and the model. The simulation results are shown in Figures 10 and 11.

The simulation results in Figures 6, 7, 8, 9, 10 and 11 show the effectiveness of the proposed control scheme.

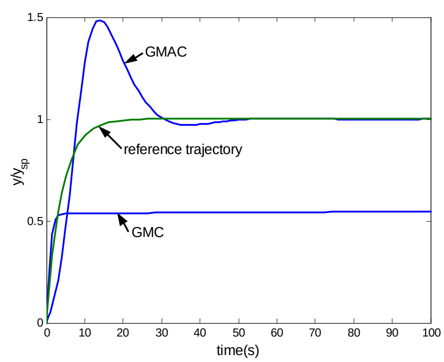

Fig. 10. Overdamped responses with a structural mismatch

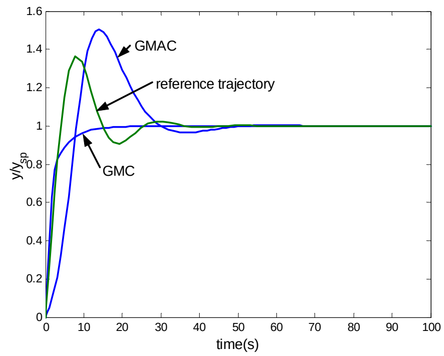

Fig. 11. Underdamped responses with a structural mismatch

IV. Conclusions

In the paper, we have proposed a generic model adaptive control scheme based on CONN, which can embed the process model into the controller by the inverted control method with CONN. It can guarantee the realizability of the generic model control scheme based on neural networks. The reference trajectory is a pseudo-second-order curve. It is very easy to tune for the controller. The proposed control scheme is of robustness and it is an effective nonlinear control scheme.

References

[1] Econmou,C.G., M.Morari and B.O.Palsson: Internal Model Control. Extension to Nonlinear Systems,Ind.Eng. Chem.Proc. Des. Dev 25, (1986) 403-411

[2] Lee,P.L. and G.R.Sullivan: Generic Model Control, Comput. Chem. Engng. 12, (1988) 573-580

[3] P.L.Lee, R.B.Newell and G.R.Sullivan. Generic Model Control-A Case Study. Canadian Journal of Chemical Engineering, 67, (1989) 478-484

[4] K.Yamuna and K.Gangiah: Adaptive generic model control: dual composition control of]distillation. AIChE Journal, 37(11), (1991) 1634-1644

[5] Z.X.Zhu and A.Jutan: Model identification and optimal stochastic control of a multivariable pressure tank system. Transaction of Institution of Chemical Engineers, 72(A), (1994) 64-76

[6] M.P.Delisa, H.J. Chae, W.A. Weigand, J.J. valdes, G. Rao, W.E. Bentley: Generic model control of induced protein expression in high cell density cultivation of escherichia coli using on-line GFP-fusion monitoring. Bioprocess and Biosystems Engineering 24, (2001) 83-91

[7] Bingjun Guo, Xiangming Hua and Arthur Jutan: Generic Model Controller Tuning for Chemical Processes with Input Saturation. Canadian Journal of Chemical Engineering, 80(2), (2002) 275-280

[8] Ye Jun and Zhang Xinhua: High speed learning algorithm for a multiplayer feedforward neural network and its application. Control and Decision, Vol. 17 Suppl. (2002) 817-819

[9] J.Fernandez de Canete, A.Garcia-Cerezo, I.Garcia-Moral: Direct control with radial basis function networks: Stability analysis and applications. Journal of Systems Architecture, 44, (1998) 583-596

[10] Martin Pottmann, H.Peter Jorgl: Radial Basis Function Networks for Internal Model Control. Applied Mathematics and Computation, 70, (1995) 283-298

Bingjun Guo was born in Auguest 1962 in Sganghai, China. He received his M.SC degree in Electrical Engineering in 1987 from Tongji University. He joined the control theory and control engineering group at East China University of Science and Technology as an Assocate professor from 1999. His research concentrates on the application of neurol networks to process control and nonlinear adaptive control.

Jinshou Yu is a professor, PhD advisor at East China University of Science and Technology. He is director of the Chemical Industrial and Engineering Society of China and vice Chairman of the Process Control Committee of the China Automation Society. Professor Yu has published 24 books on industrial automation and more than 300 papers. He has more than 41 years of experience in process control, optimaztion, control theory and applications.