APPLYING NEURAL NETWORKS TO PID CONTROLLERS FOR TIME-DELAY SYSTEMS

ZHUN YU, YING-BAI XIE, YOU-YIN JING, XU-AO LU

Department of Power Engineering, North China Electric Power University, Baoding 071003, China
E-MAIL: zhuny@ncepu.edu.cn, xieyb@ncepu.edu.cn, jingyouyin@126.com, lxn@163.com

Abstract:

Generalized PID neural network (GPIDNN) has recently received more attention in industry application. To investigate the control of long time-delay systems with GPIDNN control system, both the structure and the algorithm were presented in this paper, the real-time simulation to a main steam temperature control system was also carried out. The results show that GPIDNN is less sensitive to variation in the time-delay in comparison of conventional PID control system, it has short transition time and little over-adjust as well as ideal control quality. The results obtained during the present study indicate that GPIDNN has favorable control ability with self learning and self adapting; it is suitable substitute for conventional PID controllers for long time-delay systems.

Keywords:

PID controller; neural networks; time-delay system; simulation

1. Introduction

In recent years, PID controllers are widely used in various fields of the industry [1-10]. Unfortunately, although PID controllers have many advantages, it is difficult for them to control long time-delay systems, in which the P, I, and D parameters are not easy to choose. As is well known long time-delay systems are high-grade and complicated process control systems, their stability and control quality are descended obviously because of pure time-delay especially when models are disturbed. Conventional PID controllers with fixed parameters cannot achieve satisfying results in both stability and control quality even though a group of ideal parameters have been set. Furthermore, satisfying control quality cannot be obtained while object model changes. However, main steam temperature control system of large units has the characteristics mentioned above: it has a rather large pure time-delay and its model is not certain, moreover, pure time-delay of regulating channel will change as work condition of units varies. Therefore, to such a controlled device, a new control strategy must be discussed [11].

In the following sections, a new control system based on generalized PID neural network (GPIDNN) is presented to control long time-delay systems. Simulation of various work condition will be also performed to analyze the performance of this control system.

2. Decoupling control principle based on self learning PID neural network

The network structure and the control system are shown in Figure 1.

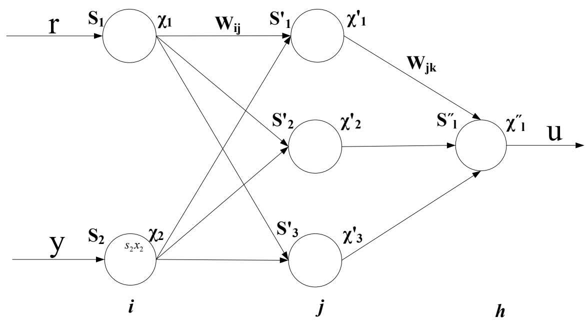

Figure 1. Input and output of network

where w_{ij} is network weight from input layer to hidden layer while w'_{jh} is network weight from hidden layer to output layer. $s_i(i=1,2)$, $s'_j(j=1,2,3)$ and s''_i are the neurons inputs of input-layer, hidden-layer and output-layer, respectively. At the same time, $\chi_i(i=1,2)$, $\chi'_j(j=1,2,3)$ and χ''_1 are the neurons outputs of input-layer, hidden-layer and output-layer, respectively.

The network, consists of a 2-3-1 structure, has three layers: input-layer, hidden-layer and output-layer. The input-layer has two neurons, one receives system setting input (r) and the other connects system output (y). The hidden-layer has three different neurons, the first is P neuron, the second is I neuron and the third is D neuron. The output-layer integrates PID control law of neurons while P, I, D parameters are embodied by net weight [12].

3. Control algorithm of GPIDNN

3.1. Forward propagation algorithm of GPIDNN

Based on setting value of control system and output of controlled device, controller output can be obtained by forward propagation algorithm of GPIDNN according to input function and output function as well as network current weight. At a random time (k), neurons inputs and outputs of input-layer are:

$$\begin{cases} s_1(k) = r(k) \\ s_2(k) = y(k) \end{cases} \tag{1}$$

$$x_i(k) = s_i(k), i = 1,2 \tag{2}$$

Hidden-layer is the key layer of network, the neurons inputs are:

$$s'_j(k) = \sum_{i=1}^{2} w_{ij} x_i(k), j = 1,2,3 \tag{3}$$

Active (output) functions of the neurons in hidden-layer are different, they can be formed as:

$$x_1'(k) = \begin{cases} 1 & x_1' > 1, \\ s_1' & -1 \le x_1' \le 1, \\ -1 & x_1' < -1, \end{cases} \tag{4}$$

$$x_2'(k) = \begin{cases} 1 & x_2' > 1, \\ x_2'(k-1) + s_2'(k), & -1 \le x_2' \le 1, \\ -1 & x_2' < -1; \end{cases} \tag{5}$$

$$x_3'(k) = \begin{cases} 1 & x_3' > 1 \\ s_3'(k) + s_3'(k-1), & -1 \le x_3' \le 1, \\ -1 & x_3' < -1; \end{cases} \tag{6}$$

The output layer only has one neuron. The input and output of the neuron are:

$$s_h''(k) = \sum_{j=1}^{3} w_{jh}' x_j'(k) \tag{7}$$

$$x_h''(k) = s_h''(k) \tag{8}$$

3.2. Back-propagation algorithm of GPIDNN

Back-propagation algorithm of GPIDNN updates network weight, namely it completes the function of study and memory. This algorithm is similar to the back-propagation algorithm of common multi-layer forward networks.

If the number of sampling points completed in time T is assumed to be m, it can lead to the training objective and rule as follows:

$$J = \frac{1}{m}\sum_{k=1}^{m}[r(k+1) - y(k+1)]^2 \tag{9}$$

After n_0 training and studying steps, the weight from hidden-layer to output-layer is:

$$w'_{jh}(n_0+1) = w'_{jh}(n_0) - \eta \frac{\partial J}{\partial w'_{jh}} \tag{10}$$

where η is step, and the weight from input-layer to hidden-layer can be expressed as:

$$w_{ij}(n_0+1) = w_{ij}(n_0) - \eta \frac{\partial J}{\partial w_{ij}} \tag{11}$$

and

$$\frac{\partial J}{\partial w'_{jh}} = -\frac{2}{m}\sum_{k=1}^{m}[r(k) - y(k)] \bullet \frac{y(k+1) - y(k)}{x''_h(k) - x''_h(k-1)} \bullet x'_j(k) \tag{12}$$
$$= -\frac{1}{m}\sum_{k=1}^{m}\delta'_{jh}(k) x'_j(k)$$

$$\frac{\partial J}{\partial w_{jh}} = -\frac{1}{m}\sum_{k=1}^{m}\delta'_{jh}(k) w'_{jh} \frac{x'_j(k+1) - x'_j(k)}{s'_j(k+1) - s'_j(k)} \bullet x_j(k) \tag{13}$$
$$= -\frac{1}{m}\sum_{k=1}^{m}\delta_{ij}(k) x_i(k)$$

4. Simulation examples of main steam temperature control system based on GPIDNN

Simulation in various operating modes is performed to test the performance of GPIDNN control system.

4.1. Introduction to object characteristic

In the present study, main steam temperature of 300MW unit is the controlled device, in the cascading system, object model is shown as Figure 2.

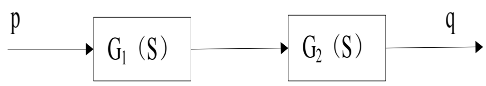

Figure 2. Object model

$$G_1(s) = \frac{7}{1+93s} e^{-61s}$$

$$G_2(s) = \frac{1.33}{1+35s} e^{-93s}$$

In the inertial region, τ=61s, T=93s, τ/T≈0.66>0.5. Thus the controlled device is a typical long time-delay system.

4.2. System simulation

Conventional PID control system and GPIDNN control system were designed for the object mentioned above. These two systems both used cascading structure. The first controller adopted conventional PID control system, its controlling parameters were taken as K_p=4.7, k_i=0.002 and k_d=28. The second controller adopted GPIDNN control system, its learning step η=0.01and the network was trained for 100 steps.

The two control systems were simulated in various work condition. In the following figures, curve 1 is response curve of GPIDNN control system while curve 2 is response curve of conventional PID control system.

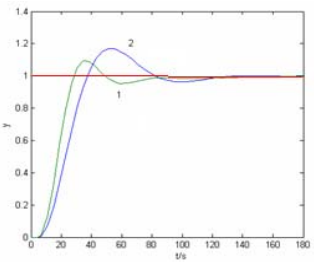

Figure 3. The system response with prescribed object

Figure.3 shows the unit step response curves of the two systems with the given objects.

As seen the transition time of curve 1 is short and over-adjust is little, it indicates that GPIDNN controller can solve the problem between stability and quick response well. So, it is applicable to long time-delay control.

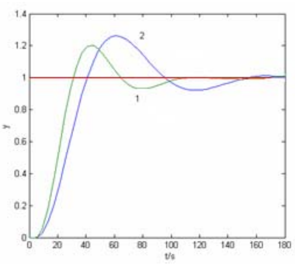

Figure 4. The system response with different inner-loop characteristic

Figure 4 shows the system response curves while the time constant of transfer function in leading region was changed from 35s to 90s. Comparing curve 1 to curve 2, it is found that the control quality of GPIDNN control system is better than conventional PID control system. Also, comparing Figures 3 and 4 shows that when time constant in leading region had a wide-range change, GPIDNN control system was not sensitive, nevertheless, the control quality of conventional PID control system deteriorated obviously and the transient time was prolonged.

Figure 5 shows the response curves of the two systems while the pure time-delay in inertial region was changed from 61s to 80s, and τ/T≈0.9. As seen, the quality of GPID NN control system is better than that of conventional PID control system.

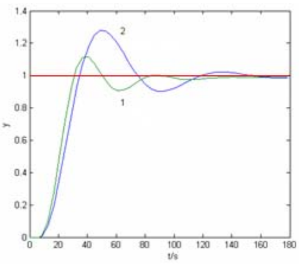

Figure 5. The system response with differentouter-loop characteristic

These simulation results have shown that GPIDNN controller can control long time-delay objects such as main steam temperature effectively. In the course of control, back-propagation algorithm also performs online setting to network weight of controller, in other words, it achieves both control and study in the same time. Since hidden-layer neurons of GPIDNN decoupling controller adopt multistep dynamic forward learning algorithm with self-feedback, in addition, instead of being mutually independent, input-layers and hidden-layers of all subnets are integrated with cross connection, hence network learning has short convergence time and quick learning speed, and it does not run into local minimum during its study.

5. Conclusions

In this paper, we have applied GPIDNN control system to a main steam temperature system. The simulation results have shown that this control system is a multilayer forward neural network within both dynamic and static characteristic, it can implement system control through self-training and self-learning while system object parameters are unknown, meanwhile, it can eliminate coupling among variables basically. Furthermore, compared with conventional PID control system, it has a number of benefits for applications in control engineering such as quick dynamic response, little over-adjust, strong self adapting, and so on.

Acknowledgements

This paper is supported by North China Electric Power University.

References

[1] A. Abbassi, L. Bahar, Application of neuron network for the modeling and control of evaporative condenser cooling load, Applied Thermal Engineering, 25, pp. 3176-3186, Jun. 2005.

[2] Junghui Chen, Tien-Chih Huang, Applying neuron networks to on-line updated PID controllers for nonlinear process control, Journal of Process Control, 14, pp. 211-230, Jun. 2003.

[3] Kapil Varshney, P.K. Panigrahi, Artificial neural network control of a heat exchanger in a closed flow air circuit, Applied Soft Computing, 5, pp. 441-465, May. 2005.

[4] P.K.Saha, Mohammed Shoib, J.Kamruzzaman, Development of a neural network based integrated control system of 120 ton/h capacity boiler, Computers &Electrical Engineering, 24, pp. 423-440, Mar. 1998.

[5] C.F.Wong, J.Shippen, B.Jones, Neural network control strategies for low specification servo actuators, Machine Tools & Manufacture, 38, pp. 1109-1124, June. 1998.

[6] A.P.Loh, K.O.Looi and K.F.Fong, Neural network modeling and control strategies for a pH process, Journal of Process Control, Vol 5 No. 6, pp. 355-362, Oct. 1995.

[7] Yusuf Oysal, Yasar Becerikli, A. Ferit Konar, Generalized modeling principles of a nonlinear system with a dynamic fuzzy network, Computers & Chemical Engineering, 27, pp. 1657-1664, Apr. 2003.

[8] A.Andrasik, A.Meszaros, S.F.de.Azevedo, On-line tuning of a neural PID controller based plant hybrid modeling, Computers & Chemical Engineering, 28, pp. 1499-1509, May. 2003.

[9] Huailin Shu, Youguo Pi, PID neural networks for time-delay systems, Computers & Chemical Engineering, 24, pp. 859-862, Dec. 2000

[10] G.W.Ng, P.A.Cook, Real-time control of systems with unknown and varying time-delays, using neural networks, Artificial Intelligence, 11, pp. 401-409, Nov. 1998.

[11] Yonghua Tao, Yixin Yin, Lusheng Ge, Application of new style PID control, Publisher of mechanical industry, Beijing, 1998.

[12] Tianyou Chai, Application of multivariable self-adapting decoupling control, Publisher of science, Beijing, 2001.