Frontiers in Signal Processing, Vol. 4, No. 1, January 2020
https://dx.doi.org/10.22606/fsp.2020.41004

Oven Controlled Crystal Oscillator Control Based on BP Neural Network Tuning PID

Wanqiang Wu, Liangfu Peng*, Gui Gan

College of Electrical & Information Engineering, Southwest Minzu University, Chengdu 610041, China
Email: 605544232@qq.com

Abstract. In order to improve the problem that the traditional PID control oven controlled crystal oscillator cannot be adjusted in real time in the process of clock taming, a BP neural network tuning PID control algorithm is proposed. BP neural network tuning PID control algorithm can learn the rule of PID control online, and can adjust the parameters of PID control in real time. The results of simulation in MATLAB show that there is no obvious overshoot and oscillation in step response controlled by BP neural network tuning PID, and the system is stable faster than the traditional PID control. Therefore, BP neural network tuning PID control has better control effect than traditional PID control in oven controlled crystal oscillator control, and has a strong self-adaptability in parameter adjustment.

Keywords: Clock taming, oven controlled crystal oscillator (OCXO), incremental PID control, BP neural network (BPNN), adaptive adjustment

1 Introduction

The research and development of 5G mobile communication network will promote the rapid development of the Internet of Things era, and in the upcoming 5G era, everything will be interconnected. Each networked device and node requires precise time scales to support the operation of the system [1]. In the white paper released by the International Mobile Telecommunication (IMT)-2020 (5G) promotion group, for some collaborative services with 300ns requirements, the recommended indicator assignments are: ±30 ns at the source, ±200ns at the load, and ±50ns at the access. Compared with the ±1.5μs index allocation of 4G basic services, each part of the index allocation of 300 ns scale has a significant improvement. This requires us to conduct research on key technologies such as synchronization source, high-precision transmission, and synchronous access [2].

The most fundamental problem in improving synchronization performance in 5G services is to improve the performance of the synchronous clock source. Global Navigation Satellite System (GNSS) signals can be used as an accurate, reliable, and stable time source. But once the GNSS signal is not available for a short period of time, a synchronous clock is required to provide a temporary clock signal, so it is necessary to tame the local clock with GNSS. The GNSS second pulse has high long-term stability and low short-term accuracy, while the local crystal oscillator has low long-term stability and high short-term precision. The local crystal after tame by GNSS second pulse has high long-term stability and short-term accuracy. Therefore, the crystal oscillator domesticated by GNSS second pulse can be used as a high precision clock source.

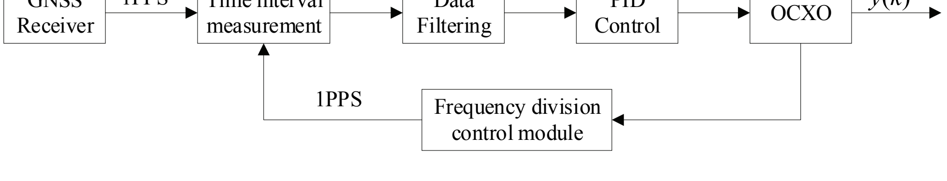

Figure 1. Schematic diagram of GNSS signal taming oven controlled crystal oscillator

Clock tame is to compare the 1PPS signal outputted by the GNSS receiver with the 1PPS signal of the crystal oscillator frequency division output, measure the frequency difference between the two, filter the obtained frequency difference data, pass it to the PID controller, and then the PID controller. Control the OCXO to adjust the output frequency to achieve synchronization with the GNSS signal [3]. The block diagram of the GNSS signal taming OCXO is shown in Fig. 1. It can be seen from Fig. 1 that the control of the OCXO is a very important part of the clock tame process during the whole clock tame process. Therefore, the high-precision time synchronization requirement for 5G requires more in-depth research on OCXO control to support the improvement of time synchronization accuracy.

The OCXO is a commonly used clock source with high short-term stability. However, the output frequency and phase of the OCXO will drift as the working time increases, where aging and temperature are the most important factors influencing the drift [4]. Therefore, it is necessary to adjust the output parameters of the crystal oscillator in time to ensure a reliable frequency of the crystal oscillator output. The control crystal oscillator is mostly controlled by traditional PID. PID control is based on the error of the system, using proportional, integral, differential to calculate the control amount for control. In general, control parameters must be determined by experience and on-site commissioning [5]. The main advantage of BP neural network is that it has strong nonlinear mapping ability. In theory, BP neural network with three or more layers can approximate a nonlinear function with arbitrary precision as long as the number of neurons in the hidden layer is sufficient [6]. Therefore, combining PID control with BP neural network makes it possible to improve the crystal control performance during clock tame.

2 Digital PID Control Algorithm

2.1 Positional PID Control

At present, the OCXO control mostly adopts the traditional PID control [7-9], and the PID control schematic diagram is shown in Fig. 2:

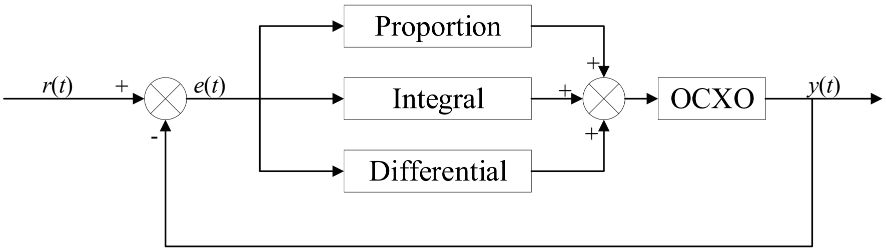

Figure 2. PID control schematic

Traditional PID control uses the difference between input and output to adjust the three parameters of proportion, integral and differential. The positional PID control formula is:

$$u(k) = K_p e(k) + K_i \sum_{i=0}^{k} e(i) + K_d[e(k) - e(k-1)] \tag{1}$$

In formula (1), K_p denotes proportional coefficient, K_i denotes integral coefficient, K_d denotes differential coefficient, K denotes sampling sequence number, $u(k)$ denotes input value of the controlled system, and $e(k)$ denotes deviation of the system at the k-th sampling.

2.2 Incremental PID Control

Through the positional PID control formula, it can be seen that each output is related to all past states. If $u(k)$ is to be calculated, not only $e(k\text{-}1)$ and $e(k\text{-}2)$ are involved, but also the previous deviations must be added together, which is prone to large cumulative error [10]. Incremental PID only needs to calculate the increment. When the calculation error or precision is insufficient, the influence on the control quantity calculation is small, so the incremental PID control is generally adopted. According to the positional PID control formula, the control amount of the k-1th sampling time is:

$$u(k-1) = K_p e(k-1) + K_i \sum_{i=0}^{k-1} e(i) + K_d [e(k-1) - e(k-2)] \tag{2}$$

By subtracting the k-1th sampling time control amount from the kth sampling time control amount, it is obtained:

$$\mathrm{u}(k) = u(k-1) + \mathrm{K}_p [e(k) - e(k-1)] + K_i e(k) + K_d [e(k) - 2e(k-1) + e(k-2)] \tag{3}$$

After determining the three parameters of K_p, K_i, and K_d, the control increment can be obtained by calculating the deviation of the three measurements before and after. The advantage of the incremental PID control algorithm is that even if the current calculation output is wrong, it will not have a big impact on the system [11].

3 BP Neural Network Tuning PID Control

3.1 PID Control System Based on BP Neural Network

The structure of the PID control system based on BP neural network is shown in Fig. 3. The whole system consists of two parts: the classic incremental PID controller and the BP neural network.

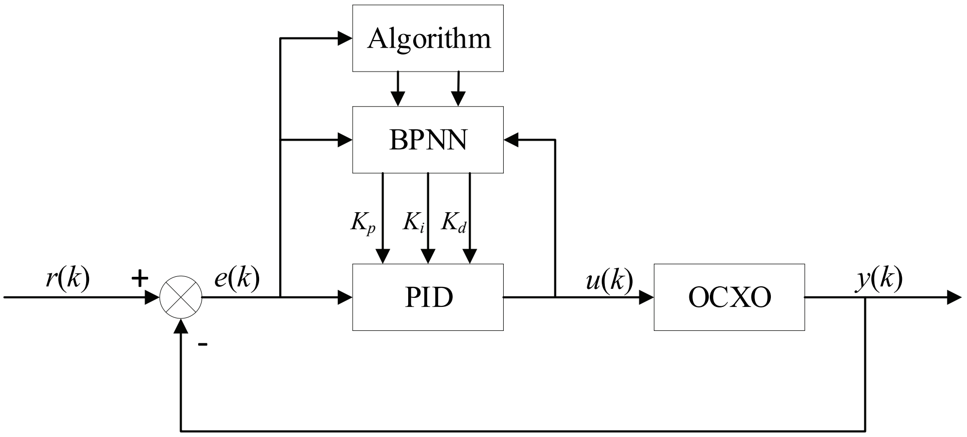

Figure 3. Based on BP neural network tuning PID control system structure

(1) Incremental PID controller: The three parameters of K_p, K_i, and K_d of the controller are online adjustment mode. The controller performs closed-loop control on the OCXO under the adjustment of BP neural network.

(2) BP neural network: The neural network obtains the input parameters of the network according to the running state of the system. The system learns the control law autonomously according to the learning algorithm and adjusts the network weight coefficient, so that the neural network outputs the optimal PID control parameters under the control law [12].

3.2 BP Neural Network Structure and Algorithm

The number of input neurons in the neural network depends on the complexity of the controlled system. In this paper, four input neurons are used: $X(1)=e(k)-e(k-1)$, $X(2)=e(k)$, $X(3)=e(k)-2*e(k-1)+e(k-2)$, $X(4)=u(k-1)$. The number of layers in the hidden layer and the number of neurons in each layer can be determined by empirical formula $q = \sqrt{n+m} + f$, where n, q and f are the number of neurons in the input layer, the hidden layer and the output layer respectively. In this paper, a single layer of hidden layer is used, and the number of neurons is 8; the number of neurons in the output layer is consistent with the three parameters controlled by PID, so the number of neurons is 3 [13]. The BP neural network structure is shown in Fig. 4.

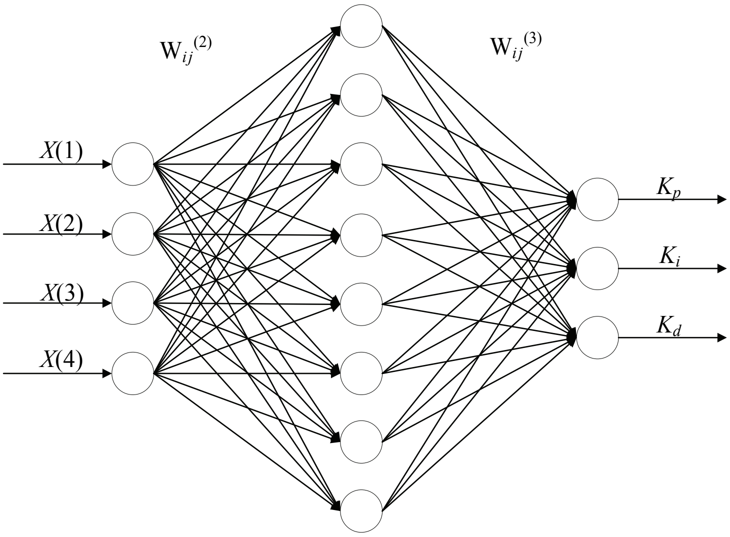

Figure 4. BP neural network structure

The input to the network input layer is:

$$I_j^{(l)} = X(j) \qquad (j = 1,2,...,M) \tag{4}$$

In the formula (4), M is 4.

The input and output of the hidden layer of the network are:

$$net_i^{(2)}(k) = \sum_{j=0}^{M} w_{ij}^{(2)} O_j^{(1)} \tag{5}$$

$$O_i^{(2)}(k) = f(net_i^{(2)}(k)) \quad (i = 1,2,3...,Q) \tag{6}$$

The $w_{ij}^{(2)}$ in the equation (5) is a weighting coefficient of the input layer to the hidden layer, and the number Q of hidden layer neurons is 8. The hidden layer activation function $f(x)$ is set to the Relu function:

$$f(x) = \begin{cases} x \ , & x > 0 \\ 0 \ , & x \le 0 \end{cases} \tag{7}$$

The input and output of the network output layer are:

$$net_i^{(3)}(k) = \sum_{j=0}^{Q} w_{ij}^{(3)} O_j^{(2)} \tag{8}$$

$$\begin{aligned} &O_l^{(3)}(k) = g(net_l^{(3)}(k)) \quad (l = 1,2,3) \\ &O_1^{(3)}(k) = K_p \\ &O_2^{(3)}(k) = K_i \\ &O_3^{(3)}(k) = K_d \end{aligned} \tag{9}$$

The output of the network corresponds to the three parameters of the PID control, and the output layer activation function $g(x)$ is still the Relu function:

$$g(x) = \begin{cases} x \ , & x > 0 \\ 0 \ , & x \le 0 \end{cases} \tag{10}$$

The cost function set is:

$$E(k) = \frac{1}{2}(r(k) - y(k))^2 \tag{11}$$

In the neural network, the gradient descent method is used to update the weight coefficient. The local minimum point is searched for the negative gradient direction of the weight coefficient according to $E(k)$, and a global minimum inertia term that makes the search fast convergence is added:

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha \Delta w_{li}^{(3)}(k-1) \tag{12}$$

In the formula (12), η is the learning rate, and α is the inertia coefficient. From the chain rule of derivation, $\frac{\partial E(k)}{\partial w_{li}^{(3)}}$ can be further expressed as equations (13) and (14):

$$\frac{\partial E(k)}{\partial w_{li}^{(3)}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial \Delta u(k)} \cdot \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial net_l^{(3)}(k)} \cdot \frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}(k)} \tag{13}$$

$$\frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}(k)} = O_l^{(2)}(k) \tag{14}$$

Since $\frac{\partial y(k)}{\partial \Delta u(k)}$ is unknown, $\operatorname{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right)$ can be used instead, and the resulting uncertain effect can be compensated by adjusting the learning rate η.

The control increment $u(k)$ obtained by equation (3) is:

$$\Delta u(k) = K_p(e(k) - e(k-1)) + K_i e(k) + K_d(e(k) - 2e(k-1) + e(k-2)) \tag{15}$$

The first three inputs of the network can be obtained by equations (9) and (15):

$$\frac{\partial \Delta u(k)}{\partial O_1^{(3)}} = e(k) - e(k-1) \tag{16}$$

$$\frac{\partial \Delta u(k)}{\partial O_2^{(3)}} = e(k) \tag{17}$$

$$\frac{\partial \Delta u(k)}{\partial O_3^{(3)}} = e(k) - 2e(k-1) + e(k-2) \tag{18}$$

Through the above analysis, the learning algorithm for the network output layer weight coefficient is:

$$\Delta w_{li}^{(3)}(k) = \alpha \Delta w_{li}^{(3)}(k-1) + \eta \delta_i^{(3)} O_i^{(2)}(k) \tag{19}$$

$$\delta_i^{(3)} = e(k)\operatorname{sgn}\left(\frac{\partial y(k)}{\partial u(k)}\right) \cdot \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \cdot g'(net_l^{(3)}(k)) \quad (l = 1,2,3) \tag{20}$$

Similarly, the learning algorithm for the hidden layer weight coefficient is:

$$\Delta w_{li}^{(2)}(k) = \alpha \Delta w_{li}^{(2)}(k-1) + \eta \delta_i^{(2)} O_i^{(1)}(k) \tag{21}$$

$$\delta_i^{(2)} = f'(net_i^{(2)}(k)) \cdot \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k) \quad (i = 1,2,\dots,Q) \tag{22}$$

The BP neural network tuning PID control algorithm flow is as follows:

(1) Determine the number of nodes in each layer of the BP neural network, that is, determine the values of M and Q, and give the initial values of the network weight coefficients of the hidden layer and the output layer and the initial values of the learning rate and the inertia coefficient;

(2) Sampling to obtain $r(k)$ and $y(k)$, and calculate $e(k)$;

(3) Calculate the input and output of each layer of the neural network, the output of the output layer is the three parameters of the PID control, and further obtain the controller output $u(k)$ according to the control parameters;

(4) Neural network online learning, the back propagation of error, and adjust the weight coefficient of the hidden layer and the output layer [14].

4 Simulation and Analysis

In order to verify the validity and feasibility of the BP neural network based PID tuning OCXO algorithm proposed in this paper, the traditional PID control algorithm and BP neural network tuning PID control algorithm were simulated in MATLAB. Assuming that the controlled object is a nonlinear system, its mathematical model is:

$$y(k)=\frac{0.8y(k-1)}{1+y^2(k-1)}+u(k-1) \tag{23}$$

For the traditional PID control, the three control parameters of the PID are set as: K_p=0.8, K_i=0.1, K_d=0.06. Before the BP neural network tuning PID control simulation, some values need to be initialized. In this paper, the inertia coefficient and learning rate of the neural network are α=0.04, η=0.28, and the initial weight coefficients of hidden layer and output layer are random numbers between [-0.5, 0.5] generated by Rands function of MATLAB. In order to observe the simulation results as accurately as possible and obtain the same contrast scale, the sampling period and simulation time of the two control algorithms are set to the same value, respectively, the sampling period is ts=0.001s, and the simulation time is 0.5s [15].

The unit step response curve of the system using BP neural network tuning PID control and traditional PID control is shown in Fig. 5. It can be seen from Fig. 5 that the BP neural network tuning PID control has a faster response speed and a smaller overshoot than the conventional PID control.

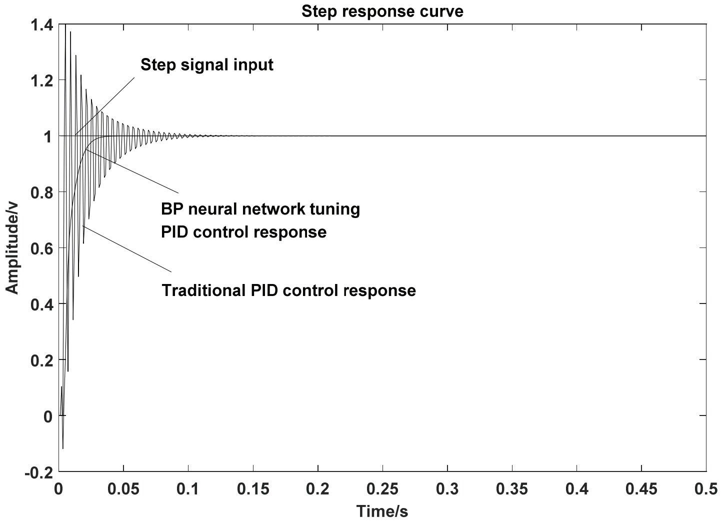

Figure 5. Step response curve

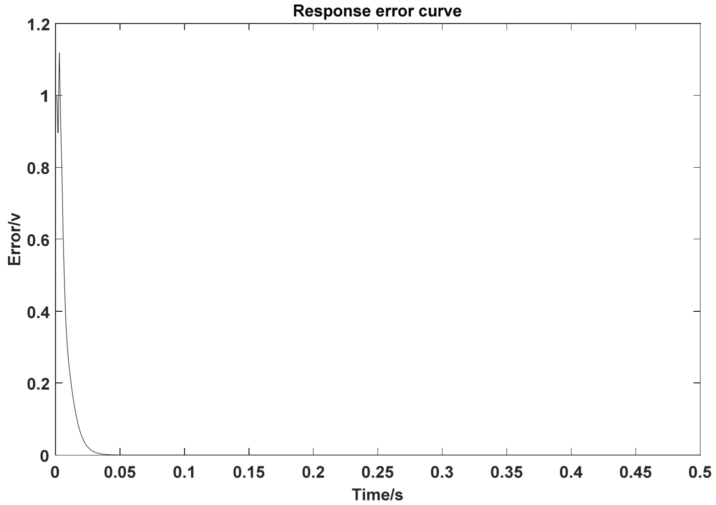

Figure 6. Response error curve

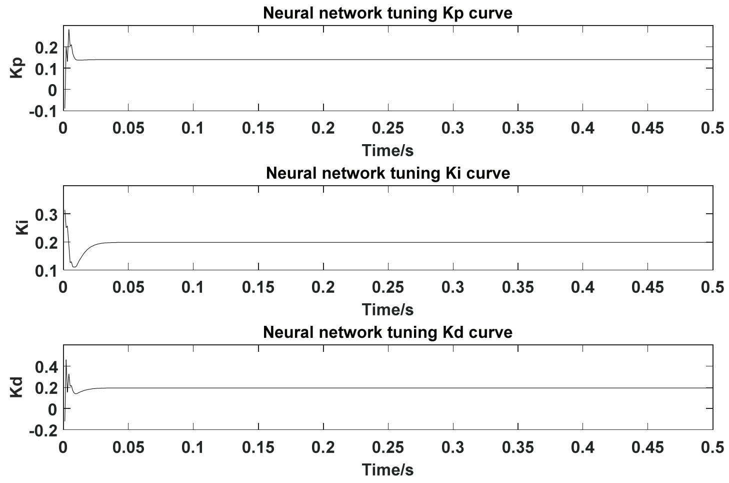

Figure 7. BP neural network tuning PID parameter curve

The error value between the input and output of the system during the BP neural network tuning process is shown in Fig. 6. The curves of three control parameters K_p, K_i, K_d of BP neural network tuning PID control in the learning process is shown in Fig. 7. It can be observed from Fig. 6 and Fig. 7 that when the system deviation exists, the BP neural network adjusts the three parameters of the PID in real time until the system output and input deviation are zero. The control parameters K_p, K_i, K_d of the traditional PID control are obtained by the trial and error method, and the parameters of the BP neural network tuning PID control are automatically adjusted by the online learning of the network, and have strong adaptability.

5 Conclusion

In this paper, the comparison of the simulation results of traditional PID control and BP neural network PID control by MATLAB shows that BP neural network tuning PID control is better than traditional PID control in the control of OCXO. The step response generated by traditional PID control has obvious overshoot and oscillation phenomenon after adjusting the three control parameters K_p, K_i, K_d according to the empirical value. The system takes a long time to stabilize. BP neural network tuning PID control adaptively adjusts the three control parameters K_p, K_i, K_d in the online learning process of BP neural network, and obtains the step response without obvious overshoot and oscillation, and the response speed is faster. The BP neural network has strong real-time performance and can be controlled in time when a deviation is detected. In summary, BP neural network tuning PID control has better control effect in OCXO control. Therefore, this control algorithm has certain reference value and application prospect for improving the performance of synchronous clock source in the aspect of 5G clock source.

Acknowledgements. The work of this paper is supported by the Southwest Minzu University Graduate Innovative Research Project (Master Program CX2019SZ23). A special acknowledgement should give to Southwest Minzu University for its experimental conditions and technical support.

References

1. Zhu Xuetian, Xia Xu, Qi Fei. 5G key technologies and business forecast[J]. Application of Electronic Technique, 2018, 44(9): 1-4, 8.
2. Liu Song, Lu Jingfei. Evolution of synchronous network architecture for 5G bearer [J]. Telecom Network Technology, 2017 (9): 22-26.
3. Xue Yicong, Gong Hang, LIU Zengjun, et al. Analysis of Disciplined Crystal Oscillator Method Based on GNSS. GNSS World of China, 2017, 42(4): 38-42.
4. Ma Yanqing. Research on adaptive taming and holding technology of oven controlled crystal oscillator OCXO [D].

Xi'an University of Electronic Science and Technology, 2011.

5. Zhang Yongzhen, Su Hansong, Liu Gaohua, et al. PID controller parameter adjustment based on BP neural network [J]. Journal of Nankai University (Natural Science Edition), 2018, 51 (3): 28-32.
6. Yu Geng, Fang Hongtao. Improved UKF integrated navigation algorithm based on BP neural network[J]. Application of Electronic Technique, 2019, 45(4): 29-33.
7. Sun Jiangtao. GPS-based 1PPS secondary frequency standard taming technology [D]. Xidian University, 2010.
8. Cai Xinpu. Research on time-frequency processing technology of cesium atomic frequency standard [D]. Xidian University, 2014.
9. Tian Lei. Development of oven controlled crystal oscillator synchronous control device based on adaptive PID[D]. Jilin University, 2009.
10. Cominos P, Munro N. PID controllers: recent tuning methods and design to specification[J]. IEE Proceedings-Control Theory and Applications, 2002, 149(1): 46-53.
11. Zhuang M, Atherton D P. Automatic tuning of optimum PID controllers[J]. 1993, 140(3): 216-224.
12. Shi Chunchao. Improvement of BP neural network algorithm and its application in PID control[D]. Tianjin University, 2006.
13. Zhu Haifeng, Li Wei, Zhang Lin. PID Control Based on BP Neural Network Tuning[J]. Journal of Dynamics and Control, 2005, 3(4): 93-96.
14. Jiangming K, Jinhao L. Self-Tuning PID Controller Based on Improved BP Neural Network[C]. Second International Conference on Intelligent Computation Technology & Automation. IEEE Computer Society, 2009: 95-98.
15. Liu Jinwei. Advanced PID Control and Its MATLAB Simulation [M]. Publishing House of Electronics Industry, 2003.