Application of Neural Network in Optimization of PID Controller

Dan Sui

School of Software Engineering, Anyang Normal University, Aanyang455000, Henan, China
School of Information Science and Technology, Wuhan University of Technology, Hubei430070, China

Zhen Jiao

School of Software Engineering, Anyang Normal University, Aanyang455000, Henan, China

Abstract
Optimization of PID controller parameters has been a hot issue in the fields of Automatic control. In the automatic control process, the controlled object has nonlinear and uncertainty characteristics. Traditional PID parameters methods are often time-consuming and difficult to obtain control effect, causing the control accuracy not high. In order to solve the optimization problem of PID controller parameters and improve system performance, we propose optimization method of PID parameters based on neural network. This method regards PID controller as the original input of neural network, the optimal parameters as the output of neural network. PID control parameters are dynamically adjusted in the control process to optimize itself by the associative memory of neural network and self-learning. Simulation results compared with the traditional PID parameters optimization method show that, this method has strong robustness and improves the system response speed, its anti-interference ability and adapt to the changing of parameter that is superior to the conventional PID control.
Key words: NEURAL NETWORK, CONTROLLER, PARAMETER OPTIMIZATION, PSO

1. Introduction

PID controller, since appeared for decades, has become a main technical tool in the industrial process control and applied successfully in the process controlled field of industry such as machinery, metallurgy, power, and light industry [1]. For the PID controller, to get ideal control effect require to optimize its three parameters before putting it into operation, namely, proportional coefficient (K_P), derivative time (T_i), derivative time(T_d). Characteristic requirements for PID controller are different due to there are all kinds of controlled objects. The goal of parameters optimization is managing to make the characteristics of the controller and controlled objects better coordinated so as to achieve the best control effect[2].If the selection of PID controller parameter is improper, its control effect will be very poor despite the PID controller itself is very advanced. However, there are all kinds of uncertainty and nonlinear in the industrial control system. These characteristics make model parameters variety. Therefore, it is extremely difficult to establish the precise traditional math model. The traditional

PID controller parameter optimization methods which can't guarantee the normal work and hard to achieve ideal control effect require the PID controller has the function of online optimization. Therefore, it has become a hot topic [3].

Recent years have witnessed the rapid growth in neural network. It can learn by itself and simulate the system parameters without knowing about the structure of the system so as to get system rule. Currently, PID controller parameters optimization adopted by neural network are becoming a hot issue[4].Especially radial basis function neural network with its nonlinear mapping approximation capability, adaptive learning, parallel processing and strong robustness is applicable for complex nonlinear system modeling and control[5,6].However, since the RBF neural network is affected by its parameters, it is imperative to optimize the parameters of neural network itself in the process of the PID controller parameters optimization so as to ensure to achieve the optimal PID controller parameters. Based on these reasons, this paper proposes a particle swarm optimization method of RBF neural network PID controller parameters optimization. The experimental results on the MATLAB platform show that the PID parameters obtained from PID controller parameters optimization method can get satisfactory control result.

2 The Principle of PID Controller

In the process of industrial control, the typical PID controller are incremental, its structure is shown in fig. 1[7]:

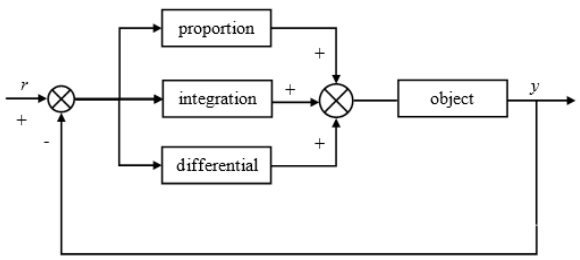


Figure 1. PID controller system structure

The conventional PID control system is composed of analog PID controller and controlled object. Its control law is:

$$u(t)=K_pe(t)+T_i\int_0^t(t)+T_d\frac{de(t)}{dt} \tag{1}$$

Among them: T_d is derivative time coefficient, T_i is integral time coefficient and K_p is proportional coefficient.

Due to modern PID controller is digital, it is necessary to discretize formula(1) so as to obtain digital PID control law[8]:

$$\begin{aligned}u(k)&=u(k-1)+K_p[e(k)-e(k-1)]\\&+t_ie(k)+T_d[e(k)-2e(k-1)+e(k-2)]\end{aligned} \tag{2}$$

It is usually adopt system out integral absolute error as integrated time absolute error(ITAE) in the process of modern engineering control:

$$J=\int_0^t t\left|e(t)\right|dt \tag{3}$$

In order to control the emergence of overshoot, if it appears, punish it. Therefore, we put overshoot into objective functions, then the optimal control index of the system is:

$$\mathrm{J(ITAE)}=\int_0^t t\left|e(t)\right|dt+ct\left|e(t\right. \tag{4}$$

The design issue of PID controller is how to choose three parameters, namely, proportionality, integration and differential coefficient. Owing to PID control parameters of the nonlinear and time varying, it is difficult for conventional PID controller parameters optimization method to obtain ideal parameters, thus the control effect is not perfect. RBF neural network is a kind of two-level feed-forward network with the simulation of the adjustment in mid-partial in human brain and neural network. It can approach to any nonlinear function with arbitrary precision and promptly adjust the three parameters of PID controller. In this regard, this paper adopt RBF neural network to optimize PID controller parameters.

3. The PID Controller of RBF Neural Network Optimization

3.1. The PID Controller of RBF

The basic idea of the PID controller parameters optimization of RBF neural network is to optimize PID controller parameters by continuous acquisition of PID controller parameters, using RBF neural network to automatically adjust proportional, integral, differential control parameters in accordance with the state of the system and nonlinear approximation.

The principle is shown in fig. 2.The controller is composed of two parts: Digital controller and BP neural network. According to the system running state, RBF neural network dynamically regulate the parameters of PID controller and finally achieve the optimal PID controller performance.

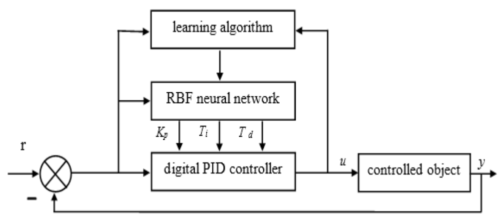


Figure 2. The Principle of RBF Neural Network PID Controller

3.2. RBF Neural Network

RBF neural network, or the radial basis function neural network, is a kind of network of efficient local approximation of feed-forward type. It distinguishes itself from other neural network with the best approximation performance and the global optimal characteristics. Thanks to its simple structure and fast training speed, it has abroad applications in the fields of time prediction, mode recognition and nonlinear function approximation. Typical structure as shown in figure 3.RBF neural network generally includes input layer, hidden layer and output layer. Hidden layer usually adopts the basis function as excitation function. This paper select Gaussian function as the hidden layer of basic functions.

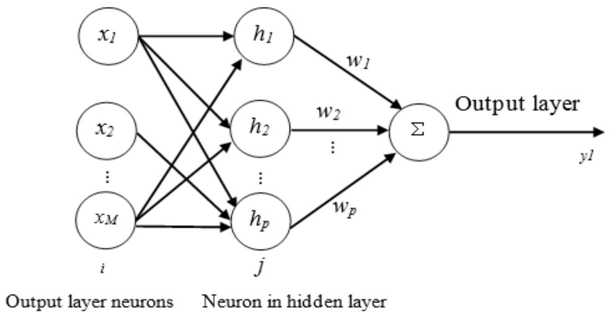


Figure 2. The Principle of RBF Neural Network PID Controller

Set network's input layer contains N neurons, input vector is $X=(x_1,x_2,\cdots,x_n)^T, X\in R^n$ The output layer contains M neurons and output variable is $Y=(y_1,y_2,\cdots,y_n)^T, Y\in R^n$.The hidden layer contains L neurons and output variable is $Z=(y_1,y_2,\cdots,y_l)^T, Z\in R^l$ Hidden nodes output to calculate through the formula:

$$Z_{ip}=\varphi(\|x_p-c_i\|=\exp[-\sum_{j=1}^{n}\frac{(x_j-c_{ij})^2}{2\sigma^2} \qquad (5)$$

Among them, c_{ii} and σ_i denote the center and width of the hidden

layer unit basis function respectively. Z_{ip} denotes the output of hidden nodes i when input the p sample.

The output of RBF network can be calculated through the formula:

$$y_{ip}=\sum_{j=1}^{l}w_{ij}\varphi(\|x_p-c_i\|) \qquad (6)$$

Among them, y_{ip} denotes the output of the i when the input of RBF neural network is the p sample . w_{ij} denotes the weight that the j radial basis function of RBF neural network connects to the i output nodes.

The performance of the RBF neural network model has direct relations with its parameters value: output weight(w_i),hidden units center(c_i)and width (σ_i).Therefore, only choose optimal w_i , c_i and σ_i can the RBF neural network predictable performance be optimum. Particle swarm optimization algorithm is a heuristic warm intelligence algorithms, which is suitable for the RBF neural network parameter optimization. Therefore this paper uses particle swarm optimization to optimize parameters of the RBF neural network.

3.3. Particle Swarm Optimization

Particle swarm optimization algorithm (PSO), the swarm intelligence algorithm put forward by Eberhard et al in 1995, is an evolutionary computation technique, which is originated from the simulation of a simplified social model. In the PSO and the D dimension solution space, each individual is seen as a particle and each particle has a speed and position information. Its speed gain a dynamic regulation in accordance with its body and peer's flying experience. Each particle has a fitness value decided by the objective function. Particles are following the current optimum particles searching in the solution space and through constant iteration to find the optimal solution. In each iteration, by tracking individual optimal value (P_{best}) and the global optimal value (g_{best}), particles can update their current location. In search of the two optimal value, each particle update their speed and position through the formula:

$$v_{id}(i+1)=\omega\times v_{id}(i)+c_1\times rand()\times(P_{best}-x_{id}(i)+c_2\times rand()\times(g_{best} \qquad (7)$$

$$x_{id}(i+1)=x_{id}(i)+v_{id}(i+1) \qquad (8)$$

$$\omega=\frac{\omega_{\max}-(\omega_{\max}-\omega_{\min})\times N}{N_{\max}} \qquad (9)$$

Among them, $v_{id}(i+1)$ denotes the current particle velocity, $v_{id}(i+1)$ denotes the updated particle velocity, $x_{id}(i)$ denotes the current location of the particle. $x_{id}(i+1)$ denotes the updated current location of the particle. $rand()$ denotes the random number of(0,1). N denotes the current evolution algebra. $N_{\max}$ denotes the maximum generation. w denotes the inertia weight. c_1 , c_2 denotes learning factors.

3.4. Particle Swarm Optimization for RBF Neural Network Optimization Process

The specific procedure for optimizing parameters w_i , c_i , σ_i as follows:

3.4.1 Initializing parameters of the particle swarm. Set the search area of the particles and initialize Pbest and gbest simultaneously.

3.4.2 Create initial particle swarm and coding the parameters of the RBF neural network into a particle.

3.4.3 Anti-spoofing each individuals into RBF neural network parameters and establish RBF neural network structure. Input training sample to the RBF neural network to train and calculate the mean square error (MSE).

3.4.4 According to its fitness value and personal best and global best value, each particle runs. If it is better than personal best and global best value, the particle will replace personal best and global best value.

3.4.5 Update each particle's search position and speed in line with the formula(7)(8)(9).

3.4.6 Judge algorithm termination criteria. If particle's evolutionary algebra become maximum iterations or MSE become the initial value, then the particle's search has done. Besides, we should regard this particle as the optimal parameter of RBF neural network Otherwise, run procedure 3.4.3 and repeat iteration constantly.

3.5. Specific Steps for PID Control Algorithm in the BP Neural Network

The specific procedure for PID controller parameter optimization as follows:

3.5.1 The initial values are w_i, c_i, σ_i.

3.5.2 Set the number of input nodes, the number of input and the number of hidden layer of RBF neural network,

3.5.3 Obtain $rin(k)$ and $yout(k)$ by sampling and calculate the error of time $k: e(k) = rin(k) - yout(k)$

3.5.4 According to the formula, we can calculate each layer's neurons' input and output of RBF neural network. Finally, network outputs $yout(k+1)$.

3.5.5 Constantly modify w_i, c_i, σ_i trough particle swarm.

3.5.6 Calculate $u(k)$ of PID controller.

3.5.7 Set $k = k+1$ and return to step 3.5.2.

It can be seen that parameter optimization process as shown in Figure 4.

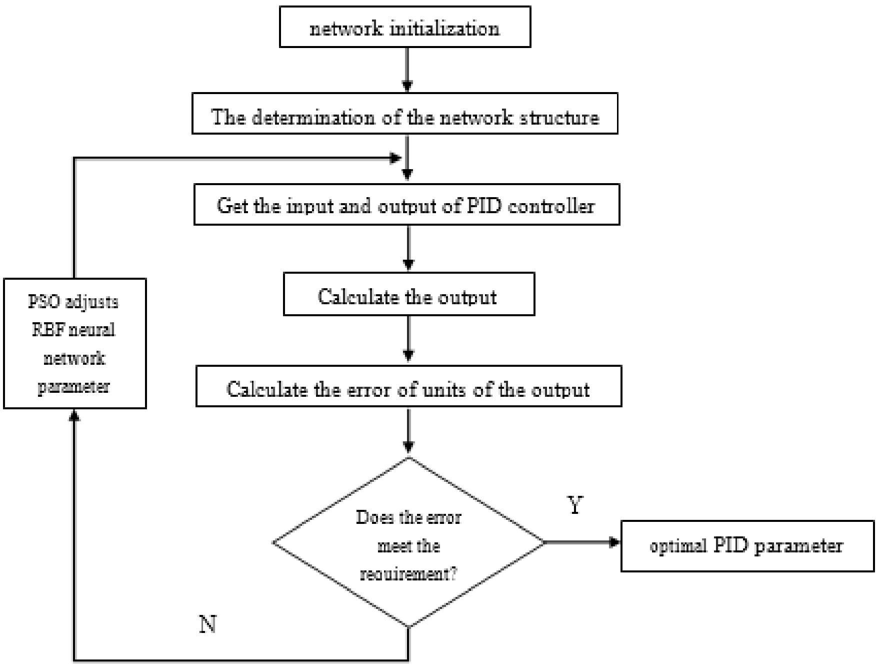


Figure 4 The Flow Chart of PID Parameter Optimization

4. Simulation Example

4.1. Simulation Object

To study the control effect of the BP neural network, we can use simulation examples for verification. Set simulation object is a nonlinear system. Its discrete model is:

$$y(k) = \frac{1.5(1-0.8e^{-0.1k})(k)y(k-1)}{1+y^2(k-1)} + 2u(k-1) \quad (10)$$

It is a slowly-time-varying system .The sampling time is 0.001s.

4.2. The Control Effect of Conventional PID Controller

In the MATLAB platform, we can use critical ratio method to determine the three parameters of PID controller,namely, $k_p = 0.2$, $k_i = 0.15$, $k_d = 0.25$.Through simulation, we can output its step response results, which we can obtain the step response curve of conventional PID controller as shown in Figure 5. It can be seen that the conventional PID controller's overshoot reaches five percent. Adjustment time is about 0.08s.

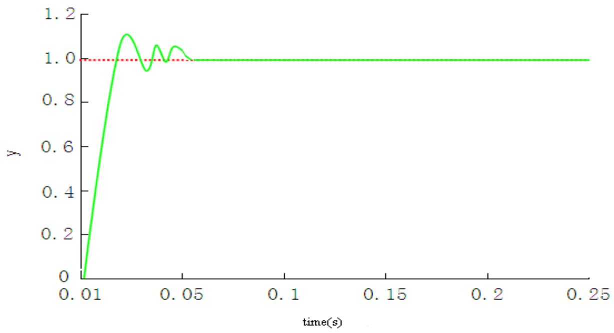


Figure 5. Response Curve of the Conventional PID Controller

4.3. The Control Effect of RBF Neural Network PID Controller

In the MATLAB platform, we can obtain the optimal structure of RBF neural network that is 2-8-1 and the learning speed is 0.45 by using PSO to optimize RBF neural network. The step response curve of RBF neural network PID controller is shown in Figure 6.It can be seen that RBF neural network PID controller's overshoot is nearly zero and adjustment time is about 0.05s.Figure 7 is the adjustment curve of PID's three parameters .Figure 8 is the change curve of the tracking error. We found that adjusting speed of the three parameters of PID is very quickly. The tracking error converges to zero. And the control effect is perfect.

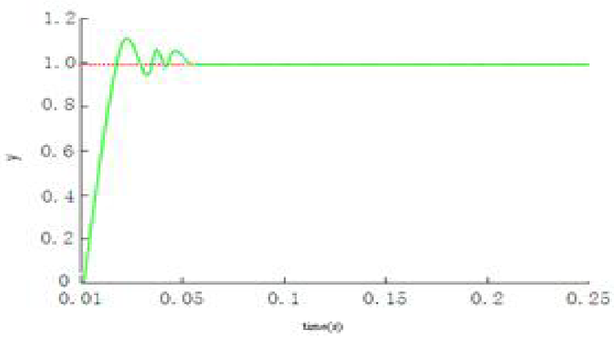
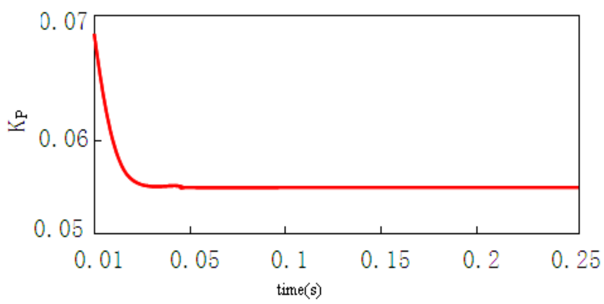
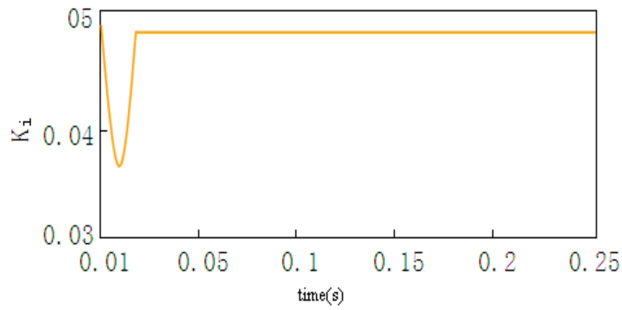


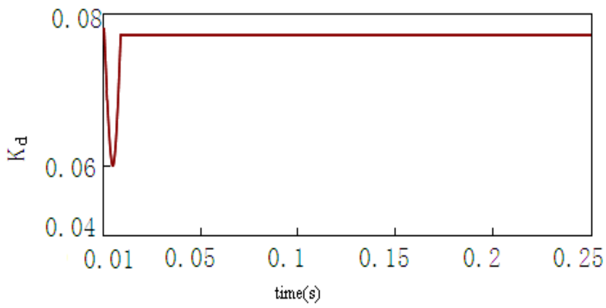
Figure6. The Step Response Curve Of RBF Neural Network PID Controller



(a)



(b)



(c)

Figure7. The Parameter Adjustment Curve Of This Paper

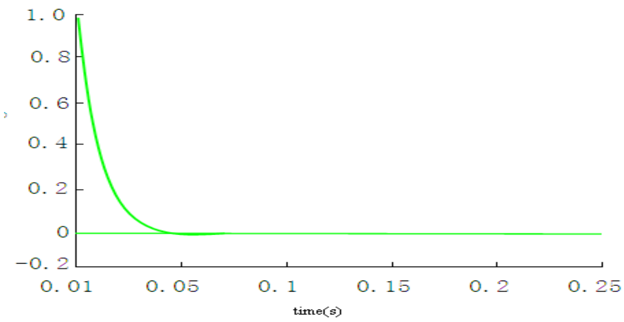


Figure 8. the Tracking Error Curve of This Paper

4.4 Anti-interference Effect of RBF Neural Network PID Controller

In order to verify the RBF neural network controller anti-interference performance, we can add distractions $v(k)=0.05$ in the 100th sampling time $t=0.1s$ .Influenced by the interference, the response curve of RBF neural network and conventional PID control are shown in Figure 9 and 10 respectively.

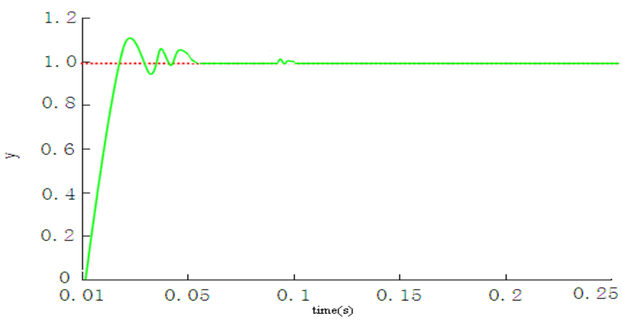


Figure 9. the Response Curve Of Conventional PID Controller Under Disturbing Forces

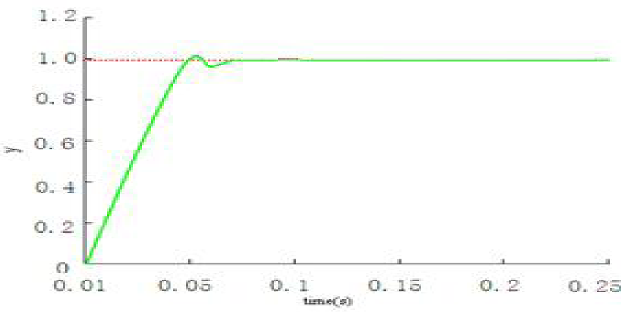


Figure 10. the Response Curve of RBF Neural Network PID Controller under Disturbing Forces

According to Figure 9 and 10,we can obviously find that under the condition that impose 5% interference on system, the influence on the response curve of RBF neural network PID controller is fairly small, which can quickly reach the stable state. And three parameters of PID controller get timely adjustments. Results indicate that the controller designed by this paper has a strong anti-jamming capability compared with conventional PID controller.

Through the simulation experiment, it can be seen that comparing to conventional PID controller, RBF neural network's overshoot is particularly small with short response time and fast response speed. At the inception of the step signal, if parameters of the PID are significant change, it can quickly adjust to the steady state. Therefore, the optimization degree of RBF neural network is fast with high control accuracy and it can be directly used in online learning optimization of PID controller. It has good robustness.

5. Conclusions

Artificial neural network theory is an emerging discipline developed in last decade. It has nonlinear mapping ability, associative memory and self-learning ability and adaptability, applying to the fields of signal analysis and speech processing. Artificial neural network is suitable for modeling and control of complex systems, especially for the uncertainty factors of system, it can demonstrate the superiority of neural network method. Given the presence of difficulty concerning the most widely used PDI controller parameters optimization in control domain and defects of conventional PID controller, this paper proposed PID parameters optimization method based on neural network. Simulation results show that, compared with the conventional PID parameters optimization method, the controller based on RBF neural network can not only improve the control effect of the conventional PID controller, but also has fast learning speed, adaptability and anti-jamming capability so as to obtain better control precision, thus suitable for modern PID controller parameters optimization.

References

1. Jinhun Liu (2003) *Advanced PID Control and MATLAB*. Electronic Industry Press: Beijing.
2. Chuanshuo Chen, Lihua Tian (1994) Tuning Method of PID Control Parameters. *Journal of Changchun Post and Telecommunications*, 12(1),p.p. 9-16.
3. Hongxin Wu, Shaoping Shen (2003) Application Of PID Control and Theory. *Control Engineering*, 10(1), p.p. 37-42.
4. Zongli Jiang (2001) *Introduction to Artificial Neural Networks*. Higher Education Press: Beijing.
5. Biliang Zhong (1999) PTD Controller Based on Neural Network Self Tuning. *Industrial Instrumentation & Automation,* 6(1), p.p. 6-8.
6. Ziwu Ren, Junshan Gao (2004) PDI Controller Backbone Neural Networks. *Control Theory & Applications*, 23 (5),p.p.16-19.
7. Ruijun Zhao, Xianlai Wang (2006) Fuzzy PID Controller in Air-conditioning Temperature Control. *Computer Simulation*, 23 (11), p.p. 311-313.
8. Yang Wang, Jiang Chu (1996) method of PID self-tuning based on recursive parameter estimation. *Information and Control*, 25(3), p.p.182-185.