Based on Neural Network PID Controller Design and Simulation

Hao Xu
School of Automation
Wuhan University of Technology
Wuhan, 430070, China
E-mail:xuhaogc@163.com

Jingang Lai,Zhenhong Yu,Jiaoyu Liu
School of Automation
Wuhan University of Technology
Wuhan, 430070, China

***Abstract*—The technologic of PID control is very conventional. There is an extensive application in many fields at present. The PID controller is simple in structure, strong in robustness, and can be understood easily. Then neural networks have great capability in solving complex mathematical problems since they have been proven to approximate any continuous function as accurately as possible. Hence, it has received considerable attention in the field of process control. Due to the complication of modern industrial process and the increase of nonlinearity, time-varying and uncertainty of the practical production processes, the conventional PID controller can no longer meet our requirement. This paper introduces the theoretical foundation of the BP neural network and studying algorithm of the neural network briefly, and designs the PID temperature control system and simulation model based on BP neural network.**

Keywords-PID controller; BP neural network; temperature control system

I. INTRODUCTION

Artificial neural network was proposed by the psychologist McCulloch and mathematicians Pitts in 1943. They put forward the first neural network of MP .They proposed formalized mathematical description and network structure method of neurons through the MP model. To prove that a single neuron can perform logic functions, thus creating the era of artificial neural networks [1]. In 1949, psychologist Hebb proposed the Hebb learning rule of changing neuron connection strength, which is "Highlight the correction hypothesis." In 1957, Rosenblatt proposed Perceptron model. It is a neural network that has a single processing unit, providing an important direction for the study of the neural network model. In 1960, Widrow and Hoff proposed adaptive linear element (Adaline) model and neural network PID controller and simulation Widrow — Hoff learning rule. Thus, in the 1960s, setting off a first wave of neural network research. The neural network control for automation and control technology as the twenty-first century technology, theory and practice at home and abroad fully proved that it play an important role in complex industrial process control. And Industrial needs advanced control methods, an urgent need for practical engineering neural network control method. Therefore, studying neural networks in control applications to improve the level of automation and economic efficiency of enterprises is of great importance [2].

In this paper, we conducted in-depth discussions about the BP neural network PID control. Using BP network can not only speed up the learning rate and reduce the shock, but also can realize the nature of binding of the neural network and PID control law. Based on the operating status of the system, BP neural network continuously adjust the three parameters of PID controller to reach some sort of performance indicators to optimize.

II. BP NEURAL NETWORK

A. BP network model

The first neural network model -MP model is proposed by MeCulloch and Pitts in 1943, as figure 1 show [3]

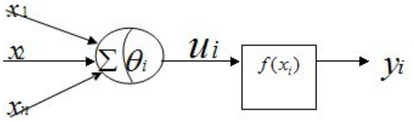

Figure 1. MP neuron model structure

y_i: The output of the neuron i;
x_i : The input of the neuron;
w_{ij} : Neuron connection weights;
θ_i : The threshold of the neuron;
$f(u_i)$: Nonlinear interaction function of neurons

The output of the neuron y_i can be describe as

$$y_i = f(\sum_{j=1}^{n} w_{ij}y_j - \theta_i)(i \neq j) \tag{1}$$

If $u_i = \sum_{j=1}^{n} w_{ij} - \theta_j$, thus $y_i = f(u_i)$.

B. Typical multi-layer forward Neural Networks —BP neural network structure and algorithm.

Multilayer forward network contains an output layer and an input layer, one or more hidden layer. In many forward network, the most typical is the error back propagation BP neural network. Structure shown as Figure 2 [4]

Published by Atlantis Press, Paris, France.

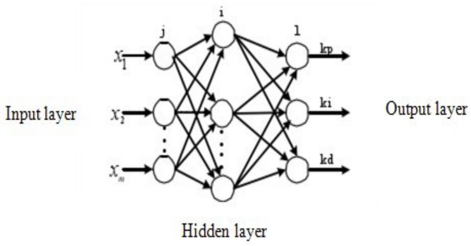

Figure 2. BP network topology diagram

The neural network input layer is made by normal nerve cell. The hidden layer and the output layer are made by applied smart nerve cell. The connecting weight value among layer is studied by error back propagation algorithm. The smart nerve cell of the hidden layer and the output layer can use LMS algorithm to fulfill inner adjustable parameters studying. If a BP network has L layer and n node and the character of node is Sigmiod function [5]. In order to make it simply, the network has just one output y. The sample number is N,(x_k , y_k) (k=1,2,⋯,N),the node i has output o_i ,if the input is x_k then the network output is y_k and the mode i has output o_{ik} .Now, concerning to layer l node j. When sample k input, the input of node is $\mathrm{net}^l_{jk} = \sum_j w^{l-1}_{jk}$

o^{l-1}_{jk} Means the output of node j when the layer is l-1 and the sample is k. Then we can have formula (2)

$$\mathrm{o}^l_{jk} = f\left(net^l_{jk}\right) \tag{2}$$

The error function is show as formula (3)

$$E_{\mathrm{k}} = \frac{1}{2}\sum_j (y_{jk} - \bar{y}_{jk})^2 \tag{3}$$

$\overline{\mathrm{y}_{jk}}$ Means the actual output of element j. Then the total error is

$$E = \frac{1}{2N} \sum_{\mathrm{k}=1}^{N} E_k \tag{4}$$

If $\delta^l_{\mathrm{jk}} = \frac{\partial E_k}{\partial_{net^l_{jk}}}$, then

$$\frac{\partial E_k}{\partial w^l_{ij}} = \frac{\partial E_k}{\partial net^l_{jk}} \frac{\partial net^l_{jk}}{\partial w^l_{ij}} = \frac{\partial E_k}{\partial net^l_{jk}} O^l_{jk} = \delta^l_{jk} O^l_{jk} \tag{5}$$

If j is a output cell, then

$$O^l_{\mathrm{jk}} = \overline{y_{jk}} \text{ ,}$$

$$\delta^{\mathrm{l}}_{\mathrm{jk}} = \frac{\partial E_k}{\partial net^l_{jk}} = \frac{\partial E_k}{\partial \overline{y_{jk}}} \frac{\partial \overline{y_{jk}}}{\partial net^l_{jk}} = \text{-}\left(\mathrm{y}_{\mathrm{k}} - \overline{y_k}\right) f'(net^l_{jk}) \tag{6}$$

If j is not an output cell, then

$$\delta^l_{jk} = \frac{\partial E_k}{\partial net^l_{jk}} = \frac{\partial E_k}{\partial O^l_{jk}} \frac{\partial O^l_{jk}}{\partial net^l_{jk}} = \frac{\partial E_k}{\partial O^l_{jk}} f'(net^l_{jk}) \tag{7}$$

O^l_{jk} is the input of next layer (l+1). In order to calculate $\frac{\partial E_k}{\partial O^l_{jk}}$,we must do from layer (l+1).As layer (l+1) cell，

$$\frac{\partial E_k}{\partial O^l_{jk}} = \sum_m \frac{\partial E_k}{\partial net^{l+1}_{mk}} \frac{\partial net^{l+1}_{mk}}{\partial O^l_{jk}} = \sum_m \frac{\partial E_k}{\partial net^{l+1}_{mk}} \omega^{l+1}_{mj} = \sum_m \delta^{l+1}_{mk} \omega^{l+1}_{mj} \tag{8}$$

Combine formula (7) with formula (8), then

$$\delta^l_{jk} = \sum_m \delta^{l+1}_{mk} \omega^{l+1}_{mj} f'(net^l_{jk}) \tag{9}$$

So：

$$\begin{cases} \delta^l_{jk} = \sum_m \delta^{l+1}_{mk} \omega^{l+1}_{mj} f'(net^l_{jk}) \\ \frac{\partial E_{\mathrm{k}}}{\partial \omega_{ij}} = \delta^l_{jk} O^{l-1}_{jk} \end{cases} \tag{10}$$

According to the analysis above, the back propagation algorithm can be shown as follow: [6]

1) Choose initial value of weight;

2) Calculate output of all nodes of hidden layer and output layer;

3) Reverse calculate gradient error;

4) Adjusting weight;

5) Repeat step 2), 3), 4) until the error is low enough.

C. *Improvement of BP network*

The same as Quasi-Newton algorithm, Levenberg-Marquardt can avoid calculate Heisen Matrix directly [7]. So it can reduce the computational comlexity and the memory need during training. Since the performance function of BP neural network is the mean square error of network, the Heisen Matrix can be approximately achieved from Jacobian Matrix, as formula (11) shows.

$$H = J^T J \tag{11}$$

Then the gradient can be calculated as formula (12)

$$\mathrm{g} = J^T e \tag{12}$$

J means Jacobian Matrix. The element of Jacobian Matrix is first derivative of network error to weight value and threshold value. **e** is the error vector of network. In Levenberg-Marquardt algorithm, the network weight value and threshold value can be calculated as formula (13)

$$x^{(k+1)} = x^{(k)} - \left[J^T J + \mu I\right]^{-1} J^T e \tag{13}$$

While scalar quantity u equal to 0,the algorithm is the same as Newton algorithm. As μ increases, the decrease amount of gradient will decrease. So network error versa. This can assure that the performance function of the network is decreasing all the time.

Levenber-Marquardt algorithm needs very large store memory. The advantage of it is that when the network weight number is few, the convergence rate is very high.

III. The PID of Neural Network apply in the temperature control system

The simulation model of BP neural network base on PID

Published by Atlantis Press, Paris, France.

temperature control system is shown as Figure 3

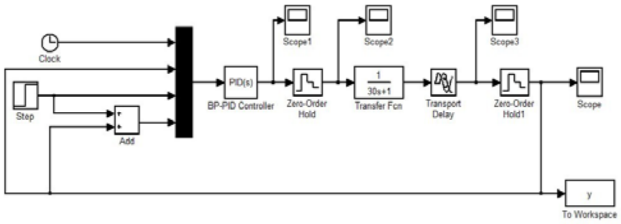

Figure 3. Simulation model

BP neural network PID control is 3-5-3 structure [9]. Mux has four input signals, Parameter settings as[u1,u2,u3,u4].Clock(u1) use for system initialization.u2,u3,u4 corresponding to a step input r_{in} ,system output y_{out} and error signal e(e= r_{in} - y_{out}). The control object parameter resistance furnace are T = 30s, τ = 10s.Given input = 200, The response curve of system neural network PID control shown as curve 3.As a comparison, curve 3 also gives a step response curve under the same conditions using the classic PID controller ,show as curve 1 and curve 2.

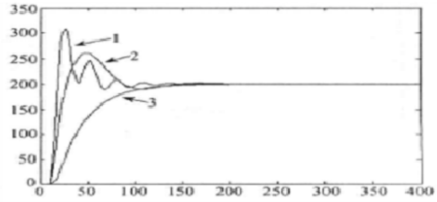

Figure 4. Step response curve of BP neural network PID control and the classic PID control

IV. CONCLUSION

Resistance furnace is widely used in metallurgical, chemical, mechanical and other types of industrial process control. Furnace temperature control has an important influence on product quality. In this paper, a improved BP network temperature control system has been established based on analyzing traditional artificial neural network. By theoretical and experimental analysis, a conclusion can be drawn that the performances of the control network is enhanced by using such improved BP network. The control precision, convergence rate and network stability have been increased observably comparing with traditional neural network. And the BP neural network PID is more adaptive, robustness, high control accuracy, a significant improvement in quality control than ordinary PID control quality. With the deepening of the study, this control method has a wide range of applications in the industrial process control prospects.

REFERENCES

[1] Gao Jun artificial neural network theory and simulation of instances [M] Beijing: Mechanical Industry Press, 2003

[2] Tao Yonghua . New PID control and its application [M]. Beijing: China Machine Press, 2002.9

[3] Liu Jinkun. Advanced PID control and MATLAB [M]. Beijing: Electronic Industry Press, 2004

[4] YI Jikai. Intelligent control technology [M] Beijing: Beijing University Press, 1999

[5] Army. Neural network PID controller and simulation [D] Hunan University Graduate Thesis, 2005

[6] Liu Di. Based on neural network PID control algorithm [D]. Hunan University Graduate Thesis, 2005

[7] Chen Hua, Cheng Yunyan. Improved BP neural network algorithm implemented in MATLAB [J]. Shaanxi University 2004, 22 (2), 45-47

[8] Chen Guiming. Using MATLAB modeling and simulation [M] Beijing: Science Press, 2001

[9] Huang Zhonglin. Control system MATLAB computing and simulation [M] Beijing: National Defense Industry Press, 2004

[10] Huang Yongan .MATLAB 7.0/Simulink 6.0 modeling and simulation development and advanced engineering applications [M], Tsinghua University Press, 2005

[11] Li Shuang, Fu Donglin. The vulcanizing machine hot plate temperature field finite element analysis [J]. Shenyang Institute of Chemical Technology, 2005, 19 (4) : 280-283.