Neural Network Based PID Control Analysis

Khizir Mahmud
Department of Electrical Engineering
Northwestern Polytechnical University
Xi'an - 710072, P.R. China
Email: khizirbd@gmail.com

***Abstract*—Through the popularity of the conventional PID control system, the implementation of the neural network on PID has gained a special concern in the control technology. Sometimes the traditional PID control technology is less encouraged for its delayed convergence rate and easy to fall into local minimum. So this research analyzed an upgraded BP algorithm and tried to design an implementation process to apply on PID control system. The algorithm convergence speed for the training process is quite good. Moreover, the trained BP neural network has self learning capability and has strong adaptive capability as well. So by applying this in the PID controllers can improve the performance very well. In the paper the PID and BP neural network, control process and control algorithm and the simulation results of neural network based PID control has been analyzed.**

Keywords—BP; controller; Matlab; neural network; PID

I. Introduction

As the technology of the industry is developing, the control parameters, sensors, unknown parameters, time varying nonlinear system, random interface are becoming more and more complex. In most industrial sectors the control process are based on the conventional PID controllers. Though the growing system complexity and the necessity of efficient control, PID control parameters don't have major changes. As a result the control object can't track in real time in changing the parameters. So sometimes the conventional system can't meet the demand of control quality in production process. Researchers are trying to improve the PID control technique. Basically the development is going on in two sectors. The improvement of structure like variable structure control is one of them. But another one is the most emerging technology which is combining the intelligent control theories like artificial neural network, fuzzy theory, genetic algorithms with the conventional PID control theory. This combination is called intelligent PID control [1]. The main advantage of this intelligent PID control is that it is independent of exact mathematical system model. Moreover, It's stability over system parameters is higher. So to implement neural network in auto process control can be more reliable. Neural network control is assumed as the combination of neural network and control theory which is one of the booming and efficient arena in intelligent control. Recently some research going on in BP network and PID control technology. But slow convergence and easy fall into local minimum are major defects of general BP algorithm [8]. So this paper shows an improved BP algorithm to implement on PID control system. The simulation result shows the improved convergence speed of the training process and also shows the strong adaptive and self learning capability of the BP neural network which improves the PID controller performance.

II. Experimental Model

For the experimental purposes a specific model has been considered which has given below. The considered plant is a nonlinear time varying plant.

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

In the above equation $a(k)$ is denoted as slow time varying where it can further derive as $a(k) = 1.2(1-0.8e^{-0.1k})$. Back propagation neural network architecture is assumed as 4-5-3, each layer initial value of weighted coefficient has been taken as the random number range from [-0.5, 0.5], the learning rate η has been considered as 0.25, momentum factor α has been assumed as 0.05. the input signal to test the system has been used the step signal. Finally a design process of neural network based PID control has been described and corresponding experimental data and figures has been depicted.

III. PID and BP Neural Network

PID control is the most common used method in industrial control because its structure is simple and it is easy to implement. PID controller has good control effect, now it has been widely used. However, PID method has a few limitations. When the parameters of controlled plant are changing over time, the parameters of controller could hardly change automatically to adjust to changing environment [2] [9]. In order to have better self-adaptability, neural network control theory is applied to realize the automatic adjustment of controller's parameters. Combining the learning characteristic of artificial neural network with traditional PID control theory, it can assist in parameter adjustment of automatic controller. The structure of PID controller based on BP neural network is shown in figure 1. The controller consists of two parts-Traditional PID controller is a part, to initiate close loop control on the plant directly and it has three parameters setting online; another part is the neural network. According to the state of the system, neural network could learn to modify the weight coefficient to regulate the PID parameters [2] and to achieve optimization of a certain performance.

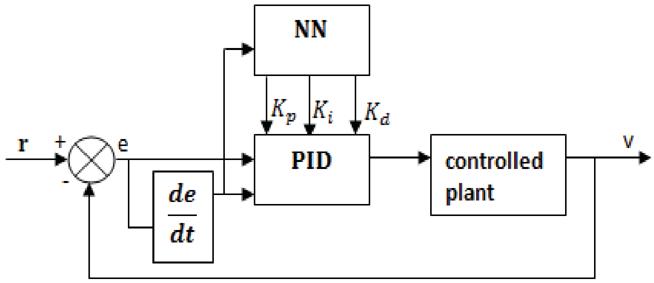

Fig. 1. PID controller structure based on neural network

Neural network (NN) in figure 1 uses BP network which has a 4-5-3 structure. This is a one-way multi-layer forward network [3]. Input nodes correspond to the state parameters of systems, for example it's necessary to normalize the error and error change of systems when necessary. The number of input variables depends on the complexity of the controlled system, output node depends on the three variable P, I and D of PID controller [4][10]. Activation function of output layer should use nonnegative Sigmoid function and the hidden layer should use positive and negative symmetric Sigmoid function because the output cannot be negative. The BP network structure which was selected by system is shown in figure 2.

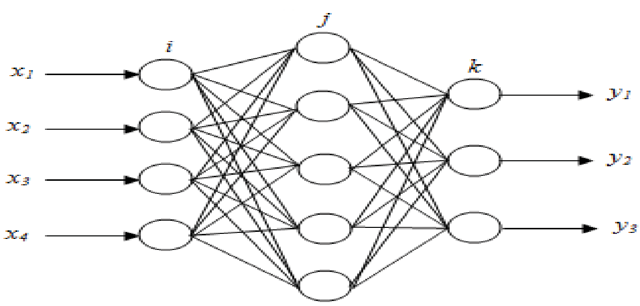

Fig. 2. BP network structure

The learning process of network consists of forward transmission and reverse transmission [11]. If the output layer could not get the expected output, the network would turn into reverse transmission process, by modifying the weights of neuron in each layer, making the output error signal minimal and by this the 4-5-3 BP structure and the improved conjugate gradient algorithm enhance the convergence rate which has been depicted in the simulation output [5]. Output layer nodes correspond to the three adjustable parameters which is K_p, K_i, K_d. Take the performance index function as:

$$E(k) = \tfrac{1}{2}(r(k) - y(k))^2$$

Using the gradient descent method to correct the weight coefficient of the network, that is to follow the negative gradient direction of $E(k)$ to weight coefficient to search and adjust, and attach an inertia item to make fast convergence global minimum [6].

$$\Delta\omega_{il}^{(3)}(k) = -\eta\frac{\partial E(k)}{\partial \omega_{il}^{(3)}} + \alpha\Delta\omega_{il}^{(3)}(k-1)$$

η is the learning rate and α is the momentum factor.

$$\frac{\partial E(K)}{\partial \omega_{il}^{(3)}(k)} = \frac{\partial E(k)}{\partial y(k)} \bullet \frac{\partial y(k)}{\partial u(k)} \bullet \frac{\partial u(k)}{\partial o_i^{(3)}} \bullet \frac{\partial o_i^{(3)}}{\partial net_i^{(3)}} \bullet \frac{\partial net_i^{(3)}(k)}{\partial \omega_{il}^{(3)}(k)}$$

And

$$\begin{cases} \dfrac{\partial net_i^{(3)}}{\partial \omega_{il}^{(3)}} = o_i^{(2)}(k) \\ \dfrac{\partial E(k)}{\partial y(k)} = -e(k) \end{cases}$$

$u(k)$ is the output of the controller at K moment; $o_i^{(3)}$ is the known as all node output. $o_1^{(3)} = k_p, o_2^{(3)} = k_i, o_3^{(3)} = k_d; net_i^{(3)}$ is the input of all nodes in output layer.

If the system adopt the incremental digital PID control algorithm, then

$$\frac{\partial u(k)}{\partial o_1^{(3)}} = e(k) - e(k-1)$$

$$\frac{\partial u(k)}{\partial o_2^{(3)}} = e(k)$$

$$\frac{\partial u(k)}{\partial o_3^{(3)}} = e(k) - 2e(k-1) + e(k-2)$$

IV. CONTROL PRINCIPLE

The incremental digital PID control algorithm is

$U(K)=U(K-1)+K_P(error(K)-error(K-1))+K_i\ error(K)+K_d(error(K)-2error(K-1)+error(K-2))$

Where K_p, K_i, K_d is proportional, integral and differential coefficient respectively. In this experiment, BP network which has a 4-5-3 structure has been used. Input of input layer is

$$O_j^{(1)} = x(j) \qquad j = 1,2,3,4$$

Input and output of hidden layer is

$$net_i^{(2)}(k) = \sum_{j=0}^{4} w_{ij}^{(2)} O_j$$

$$O_i^{(2)}(k) = f(net_i^{(2)}(k)) \qquad (i = 1,\ldots 5)$$

Where $w_{ij}^{(2)}$ is weighted coefficient of hidden layer; superscript (1), (2), (3) stands for input layer, hidden layer and output layer respectively. Activation function of the hidden layer should use positive and negative symmetric Sigmoid function [7]:

$$f(x)=\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

Input and output of output layer is

$$net_l^{(3)}(k)=\sum_{i=0}^{5} w_{li}^{(3)} O_i^{(2)}(k)$$

$$O_l^{(3)}(k)=g(net_l^{(3)}(k)) \qquad (l=1,2,3)$$

$$O_1^{(3)}(k)=k_p$$

$$O_2^{(3)}(k)=k_i$$

$$O_3^{(3)}(k)=k_d$$

Output nodes of output layer respectively correspond to the three adjustable parameters K_p, K_i, K_d . Activation function of output layer should use nonnegative Sigmoid function because the output cannot be negative [5].

$$g(x)=\frac{1}{2}(1+\tanh(x))=\frac{e^{x}}{e^{x}+e^{-x}}$$

Take the performance index function as:

$$E(k)=\tfrac{1}{2}(r(k)-y(k))^2$$

Where *r(k)* is system input, *y(k)* is system output. Take input signal as step signal, that is *r(k)*=1.0. Using the gradient descent method to correct the weight coefficient of the network, that is to follow the negative gradient direction of $E(k)$ to weight coefficient to search and adjust and attach an inertia item to make fast convergence global minimum [7].

$$\Delta\omega_{il}^{(3)}(k)=-\eta\frac{\partial E(k)}{\partial\omega_{il}^{(3)}}+\alpha\Delta\omega_{il}^{(3)}(k-1)$$

η is the learning rate and α is the momentum factor. In this experiment, $\eta=0.25$, $\alpha=0.05$.

$$\frac{\partial E(K)}{\partial\omega_{il}^{(3)}(k)}=\frac{\partial E(k)}{\partial y(k)}\bullet\frac{\partial y(k)}{\partial u(k)}\bullet\frac{\partial u(k)}{\partial o_i^{(3)}}\bullet\frac{\partial o_i^{(3)}}{\partial net_i^{(3)}}\bullet\frac{\partial net_i^{(3)}(k)}{\partial\omega_{il}^{(3)}(k)}$$

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

Since $\frac{\partial y(k)}{\partial u(k)}$ is unknown, replace it with sign function $\mathrm{sgn}(\frac{\partial y(k)}{\partial u(k)})$. Now the equation becomes,

$$\frac{\partial u(k)}{\partial O_1^{(3)}(k)}=error(k)-error(k-1)$$

$$\frac{\partial u(k)}{\partial O_2^{(3)}(k)}=error(k)$$

$$\frac{\partial u(k)}{\partial O_3^{(3)}(k)}=error(k)-2error(k-1)+error(k-2)$$

According to the analysis the weighted coefficient learning algorithm of output layer of network is

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

$$\delta_i^{(3)}=error(k)\,\mathrm{sgn}(\frac{\partial y(k)}{\partial u(k)})\frac{\partial u(k)}{\partial O_1^{(3)}(k)}g'(net_l^{(3)}(k))$$

$(l=1,2,3)$, In the same way, the weighted coefficient learning algorithm of hidden layer of network can be found

$$\Delta w_{ij}^{(2)}(k)=\alpha\Delta w_{ij}^{(2)}(k-1)+\eta\delta_i^{(2)}O_j^{(1)}(k)$$

$$\delta_i^{(2)}=f'(net_i^{(2)}(k))\sum_{l=1}^{3}\delta_l^{(3)}w_{li}^{(3)}(k) \qquad (i=1,\ldots 5)$$

Where $g'(x)=\frac{\partial g(x)}{\partial x}=\frac{\partial(\frac{e^{x}}{e^{x}+e^{-x}})}{\partial x}=\frac{2}{(e^{x}+e^{-x})^2}$

$$f'(x)=\frac{\partial f(x)}{\partial x}=\frac{\partial(\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}})}{\partial x}=\frac{4}{(e^{x}+e^{-x})^2}$$

V. Control Algorithm

The adjustable PID control algorithm Based on Neural Network is as follows:

1) In the experiment, BP network input layer node is 4 and the number of hidden layer node is 5. The initial value $w_{ij}^{(1)}(0)$ and $w_{li}^{(2)}(0)$ of the weighted coefficient of each layer is given (for convenience some random number range [-0.5, 0.5] has been taken in the experiment). Learning rate and momentum factor has been taken as η=0.25, α = 0.05) and k = 1.
2) $r(k)$ and $y(k)$ has been calculated by sampling and calculate error at this time using $error(k)=r(k)-y(k)$.
3) The output of output layer of NN is the three adjustable parameters k_p, k_i, k_d of PID controller and to calculate the output of PID controller $u(k)$.
4) Start the learning process and adjust the weighted coefficient $w_{ij}^{(1)}(0)$ and $w_{li}^{(2)}(0)$ to achieve optimum PID controller parameters. Make $k=k+1$ and return to 1.

VI. SIMULATION PROGRAM

Since a BP network has been chosen which has a 4-5-3 structure, so to set the primary value of input and hidden layer it's better to use a random number [11]. The initial value of weighted coefficient of the input layer is a 4*5 matrix and the initial value of weighted coefficient of the hidden layer is a 5*3 matrix [3]. The three output values of the output layer is the three parameters of the PID controller k_p 、 k_i 、 k_d .

IN=4; H=5; Out=3

wi=0.5*rands(H,IN) wo=0.5*rands(Out,H)

After debugging the initial value of the weighted coefficient, finally obtained a best set of random numbers. Initial value of weighted coefficient of input layer is

```
wi = [-0.4394   -0.2696   -0.3756   -0.4023;
    -0.4603   -0.2013   -0.4024   -0.2596;
    -0.4749    0.4543   -0.4820   -0.5437;
    -0.3625   -0.0724   -0.6463   -0.2859;
     0.1425    0.0279   -0.5406   -0.4660];
```

Initial value of weighted coefficient of hidden layer is

```
wo = [ 0.4576 0.2616 0.5820 -0.1416 -0.1325;
     -0.1146 0.2949 0.4352   0.2205  0.4508;
      0.4201 0.4566 0.7672   0.4962  0.3632];
```

VII. EXPERIMENTAL RESULT

For a given step input signal, through running and debugging, simulation data and control curves are shown below. In figure 3, PID controllers parameters has been shown by which the convergence rate of the training sample and the controller convergence curve and system response curve has been shown.

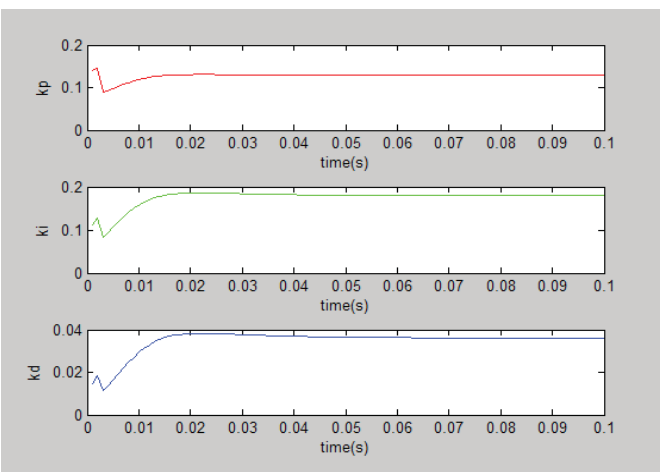

Fig. 3. Curve of PID controller parameters

For the hidden layers and the learning rate η = 0.25 the curves of the three parameters k_p 、 k_i 、 k_d of PID controller are shown in figure 3. The convergence curve of the training sample has shown in figure 4 and the convergence curve of controller input u(k) of PID controller is shown in figure 5. The challenge of the existing neural network based PID controller is the slow convergence rate. So this paper gives a special emphasis to reduce the convergence rate of training sample and controller. From the following figure 4 and 5 it is obvious that the convergence rate is very fast and few milliseconds which makes it more efficient.

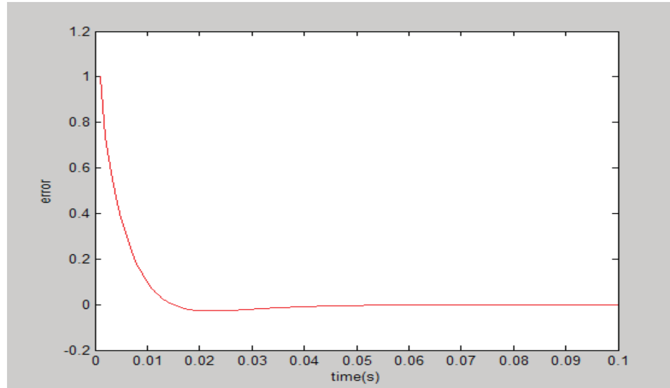

Fig. 4. Convergence curve of training samples

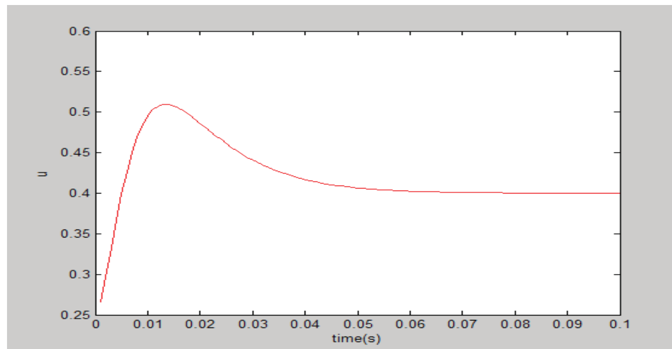

Fig. 5. Convergence curve of controller intput u(k)

Finally the curves of system input signal and the output response are shown in the following figure 6. In the figure it is clear that the system rise time and the overshoot is very low and in very efficient level. Within very short time it become steady state and without having any extra or unexpected ripple.

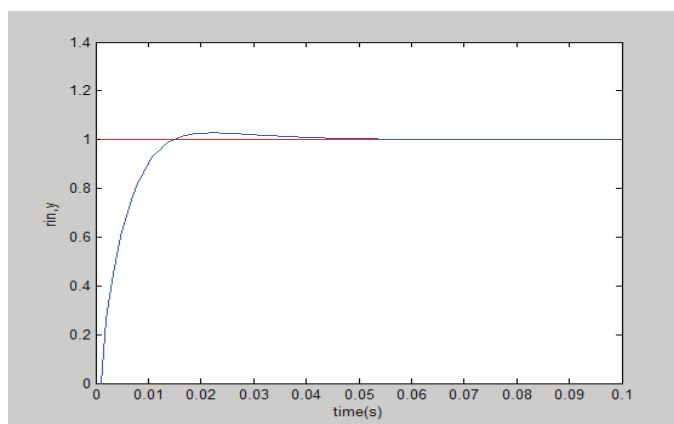

Fig. 6. Curve of the system response

From simulation results, it can observe that the speed and stability of PID control based on BP neural network is better than ordinary PID control. In the system step response, the parameters of PID controller are updated real-time with the online learning of BP network which makes the system approximate quickly to the target function and illustrates the approximation ability of BP neural network to any function [1]. There is no need for PID control Based on BP neural network to establish the mathematical model of the controlled system and it is convenient to adjust the controller parameters [12]. The suggested neural network based PID controller dynamic and static characteristics are better for large lag, time-varying and nonlinear complex systems. Learning speed of BP neural network primarily depends on learning rate η. For preliminary stage the more bigger the value of η the more faster the learning speed. But for the optimum operation the value of η should keep small. If the value is bigger then the weighted coefficient will produce repeated oscillation and can never be converged [1] [7][13]. Variable learning rate program can slowly create the value of η smaller when the learning process in progress and the result become more better. BP neural network learning algorithm is a global approximation method, so it has good generalization capability. However, there exist multiple extreme points for target functions, so using the gradient descent method to learn is simple for fallkng to local minimum.

VIII. Conclusion

BP neural network is an effective and reliable neural network structure. For any arbitrary precision BP neural network can approach any nonlinear function. Moreover, it's structure is more simple and it has better approximation performance. So the combination of conventional PID controller and BP neural network is more robust in behavior. In the paper, some limitation of the general BP algorithm like easy fall to local minimum and slow convergence has been improved. Based on that improved algorithm an intelligent PID control system has been proposed. The simulation result shows that the trained BP neural network's self learning and adaptive capability is much better and it enhance the PID controller performance.

References

[1] Jianhua Yang, Wei Lu, Wenqi Liu, "PID Controller Based on the Artificial Neural Network", Springer, ISNN 2004, LNCS 3174, pp. 144–149, 2004

[2] Lie Jasa , Ardyono Priyadi, Mauridhi Hery Purnomo, "PID Control for Micro-Hydro Power Plants based on Neural Network, Proceedings of the IASTED Conference, April 2 - 4, 2012 Phuket, Thailand

[3] J. Rivera-Mejía, A.G. Léon-Rubio, E. Arzabala-Contreras, "PID Based on a Single Artificial Neural Network Algorithm for Intelligent Sensors", IEEE Transactions, Vol. 10, No.2, April 2012

[4] Beitao Guo, Hongyi Liu, Zhong Luo, Fei Wang, "Adaptive PID Controller Based on BP Neural Network", International Joint Conference on Artificial Intelligence, China, 2009

[5] Michiyo Suzuki, Toru Yamamoto, Toshio Tsuji, "A Design of Neural-Net Based PID Controllers with Evolutionary Consumption", IEICE Trans. Fundamentals, Vol. E87-A, No. 10, October, 2004

[6] Liu Luoren, Luo Jinling, "Research of PID Control Algorithm Based on Neural Network", ELSEVIER, ESEP 2011, Singapore, 9-10 December, 2011

[7] R.Vinodha, S. Abraham Lincoln, J. Prakash, "Multiple Model and Neural based Adaptive Multi-loop PID Controller for a CSTR Process", World Academy of Science Engineering and Technology, 2010

[8] REN Zi-wu, Jun-shan GAO, "Neural Network-Based PID Controller", Techniques of Automation and Applications, China National Knowledge Infrastructure, 2004

[9] Chen, Junghui, Tien-Chih Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control", Journal of Process Control, ELSEVIER, 2004

[10] Chang, Wei-Der, Jun-Juh Yan, "Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems", Chaos, Solitons & Fractals, ELSEVIER, 2005

[11] Zhuo Li, Xiao Deyun, He Shizhong, "A Fuzzy Adaptive PID Controller Based on Neural Network", Control and Decision, China National Knowledge Infrastructure, 1996

[12] Gou-JenWang, Chuan-Tzueng Fong, Kang J.Chang, "Neural-network-based self-tuning PI controller for precise motion control of PMAC motors", Industrial Electronics, IEEE Transactions on 48.2 (2001), 2001

[13] C-L. Chen, F-Y. Chang , "Design and analysis of neural/fuzzy variable structural PID control systems", IEEE Proceedings-Control Theory and Applications 143.2 (1996): 200-208, 1996