International Journal of Engineering and Applied Sciences (IJEAS)
ISSN: 2394-3661, Volume-6, Issue-6, June 2019

Auto-tuning controllers of a class of plants using gradient descent algorithm

Nguyen Trong Toan, Nguyen Nam Trung

Abstract: **In industry, PID controller is widely used in the regulator of technology processes. Many methods of designing conventional PID controllers do not work well for time-delayed linear, vague, and nonlinear systems. Therefore, PID parameter adjustment plays an important role. To solve this problem, many recent studies have focused on the development of auto-tuning controllers. Using the method of gradient descent algorithm to the parameter space of the controller whose coefficients developed by adaptive algorithms in this paper is adjusted automatically online to minimize the performance index without knowing the exact mathematical model of the plant. The simulation and test results on real plant are presented to illustrate the performance of the proposed auto-tuning PID controllers and a comparison with its traditional PID controller is made.**

***Index Terms*—Gradient descent, Auto-tuning controller, PID-neuron, A thermal process, A class of plants.**

I. INTRODUCTION

Despite the recent evolution of alternative control techniques, a relatively larger percentage of industrial control processes is still regulated by PID controllers because of its simple structure, robustness in operation and ease of implementation. The first practical PID controller tuning method was designed based on Ziegler-Nichols principle [1-2]. However, this controller has some limitations with nonlinear system, time-delayed linear system, and vague system, time varying systems [7-8]. It is difficult to obtain good control qualities for such systems simply using the fixed PID parameters. To improve this control performance, several schemes of auto-tuning PID controllers have been proposed for implementing the control systems [3-5]. A way to tune parameters in PID controller is based on model matching [6]. This method uses plant and reference transfer functions to determine the PID parameters. Another approach is to use the neural controllers having like-PID structure to control the thermal processes [9-13]. In this paper, the key idea is the use of the gradient descent method to update the parameters of auto-tuning controllers and compare the control results show that the proposed methods has more advantage than the traditional PID in term of performance. The paper is organized as follows. In the next section, an auto-tuning PID controller is adjusted using the gradient method for a class of first-order systems with time-delay such that the integrated square error is minimized. In Section 3, a PID-neural tuning controller is proposed. Simulation and experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

Nguyen Trong Toan, Faculty of Electronics Engineering, Thai Nguyen University of Technology, Vietnam, +84977426445
Nguyen Nam Trung, Faculty of Electronics Engineering, Thai Nguyen University of Technology, Vietnam, +84978475599

II. ADAPTIVE GRADIENT METHOD

The adaptive gradient control algorithm is also one of the methods to synthesize durable parametric control system provided that the dynamic characteristics of the system are slow in compared with transition process.

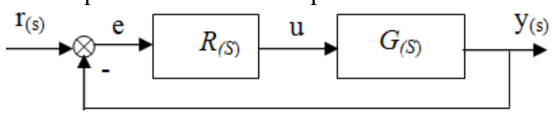

Fig. 1: System control diagram

Where,
r = the set-point (reference)
y = the actual output of the plant ($G_{(S)}$)
e = the error between the actual and desired value
And u = the control signal (output of the controller)
Fig. 1 is represented in $e = r - y$.

The problem is to find the change law of the parameter vector ξ that needs to be added to the system. The Regulator $R_{(S)}$ with parameter vectors ξ need to be automatically adjusted to minimize the deviation function:

$$J = \frac{1}{2}e^2 \rightarrow \min \quad (1)$$

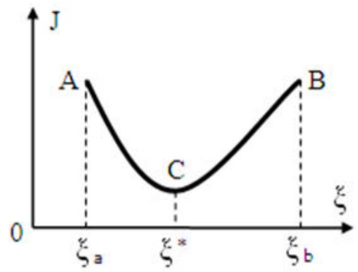

Fig. 2: Extremely characteristic system

Consider the derivative at

- Point A: $\frac{dJ}{d\xi} < 0$; - Point B: $\frac{dJ}{d\xi} > 0$

- Point C: $\frac{dJ}{d\xi} = 0$

Based on the above points, we can build the algorithm of gradient descent:

$$\frac{d\xi}{dt} = -\gamma \frac{dJ}{d\xi}$$; In which $\gamma > 0$ is an adaptive coefficient

$$J=\frac{1}{2}e^2\Rightarrow\frac{dJ}{d\xi}=\frac{\partial J}{\partial e}\frac{\partial e}{\partial \xi}=e\frac{\partial e}{\partial \xi}$$

$$e=r-y\Rightarrow\frac{\partial e}{\partial \xi}=\frac{\partial r}{\partial \xi}-\frac{\partial y}{\partial \xi}=-\frac{\partial y}{\partial \xi}$$

Therefore: $\frac{d\xi}{dt}=-\gamma e\frac{\partial y}{\partial \xi}$

We can prove another way by choosing the J as the Lyapunov functions:

$$\frac{dJ}{dt}=\frac{\partial J}{\partial e}\frac{\partial e}{\partial \xi}\frac{d\xi}{dt}=e\frac{\partial e}{\partial \xi}(-\gamma e\frac{\partial e}{\partial \xi})=-\gamma e^2\left(\frac{\partial e}{\partial \xi}\right)^2\le 0 \quad (2)$$

From the equation above (2), we know that J will decrease monotonically with time. Therefore, it can be concluded that:

$$\lim_{t\to\infty}(\xi^*-\xi)=0$$

On the other hand, follow the general diagram:

$$\frac{\partial y}{\partial \xi}=e\frac{R}{1+RG}.\frac{\partial R}{\partial \xi}$$

Therefore, we obtain the following tuning algorithm:

$$\frac{d\xi}{dt}=-\gamma e^2\frac{R}{1+RG}.\frac{\partial R}{\partial \xi} \quad (3)$$

The expression (3) is the general expression of the auto-tuning algorithm according to the Gradient. The auto-tuning parameter here is vectors ξ dependent on the e and the transfer function of the controller R:

If the controller has a differential component then the system will be affected by disturbances. Therefore, we only consider PI auto-tuning controller:

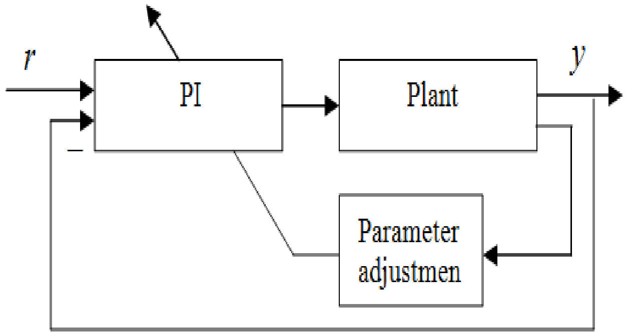

Fig 3: Diagram of the auto-tuning two-parameter controller

Align both parameters: The proportional part (K_1) and the integral part K_2 of the controller:

$$G_{PI}=K_1+\frac{K_2}{s} \quad ; \quad \frac{dK_1}{dt}=-\gamma_1 e^2\frac{R}{1+RG}.\left(1+\frac{K_2}{K_1 s}\right)$$

$$\frac{dK_2}{dt}=-\gamma_2 e^2\frac{R}{1+RG}.\frac{1}{s} \quad (4)$$

Firstly, let's assume that the plant has a form of first-order system with delayed time and a PI auto-tuning controller is used. Then, the transfer function of the resistor furnace is identified:

$$G_{(S)}=\frac{0.578}{452s+1}e^{-45s} \quad (5)$$

Control algorithm (3) is designed for plant (5) based on Simulink of Matlab:

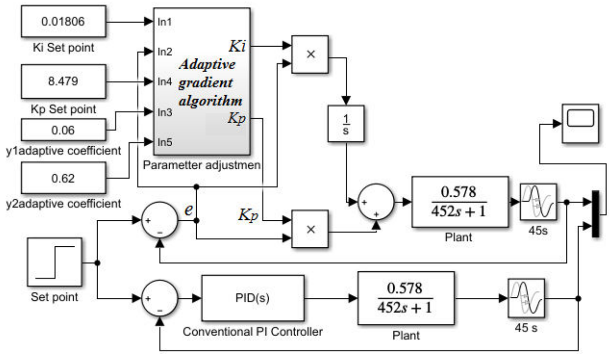

Fig. 4: Block diagram of controllers using conventional PI and adaptive gradient controller

The initial set values for the auto-regulating PI controller, *K=[0.01806 8.479]*, are determined according to the conventional methods. This controller will adjust its parameters to bring the object to stability and setting mode. Thus parameters $\xi=K=[K_1, K_2]$ will change according to the deviation e until the target J reaches the minimum. Adaptive coefficients are chosen $\gamma_1=0.06$; $\gamma_2=0.62$ in Mat lab/Simulink (*Fig. 4, 7)*.

III. PID-NEURAL CONTROLLER

The PID-Neural controller, a NN-like PID control method, has ability to make plant stable because of its nonlinearity. In addition, training algorithm enables the controller to adapt with changes of plant or noise. The PID-neural controller, having 3 inputs and 1 output, is shown in *Fig 5*. The inputs of neural network (e_P, e_I, e_D) are created by proportion, integration and derivation of error e.

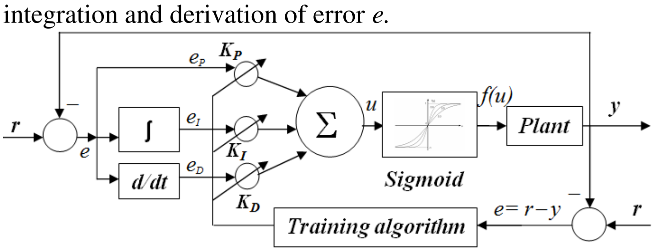

Fig 5: PID-neural control system

$$u=K_P e_P+K_I e_I+K_D e_D \quad (6)$$

Activation function f of the neuron, producing the neural output $f(u)$, is sigmoid:

$$f(u)=\frac{2}{1+e^{-bu}}-1 \quad ; \quad f'(u)=\frac{2be^{-bu}}{\left(1+e^{-bu}\right)^2} \quad (7)$$

Our goal is to minimize the following error:

$$J=\frac{1}{2}(r-y)^2=\frac{1}{2}e^2\rightarrow\min \quad (8)$$

e is error between reference *(r)* input and output *(y).*

The weights, PID controller parameters, can be adjusted by the algorithm of gradient descent. The update rules for these control parameters are expressed as:

$$\begin{cases} K_P=K_P-\eta\frac{\partial E}{\partial K_P}=K_P-\eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_P}=K_P+\eta ef'(u)e_P \\ K_I=K_I-\eta\frac{\partial E}{\partial K_I}=K_I-\eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_I}=K_I+\eta ef'(u)e_I \\ K_D=K_D-\eta\frac{\partial E}{\partial K_D}=K_D-\eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_D}=K_D+\eta ef'(u)e_D \end{cases} \quad (9)$$

where the learning step η is chosen 0.01

if parameters (K_P, K_I, K_D) are adapted according to (9) then the cost function J will decreas with time [14]. The PID-neural controller is emulated in simulink toolbox of Matlab on the basis of the gradient descent algorithm for the resistance furnace.

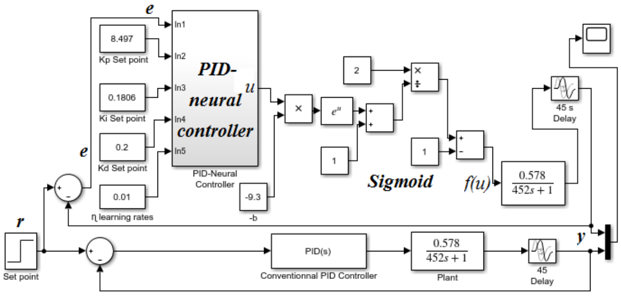

Fig 6. Block diagram of PID-neural and conventional PI control system

(9)

IV. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation results

Using the classical method to determine the preset value of the PI controller, these parameters will be adjusted to stabilize and set value.

Control diagram is as follows:

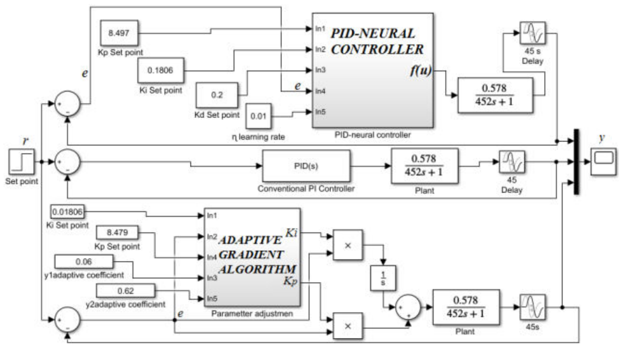

Fig. 7: The overall model of a furnace temperature control system using three controllers in Matlab / Simulink.

Matlab-based simulation results of the temperature control system controlled by the controllers are shown in *Fig. 8* and the step response parameters are shown in Table. 1.

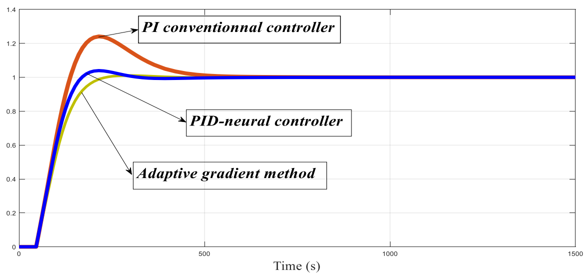

Fig. 8: Responses to a step function with different controllers

Table 1: Step response parameters compared with the traditional PID

	Settling Time ($T_{5\%}$ sec)	Rise Time (T_r sec)	Overshoot (σ%)
Conventional PID	380	210	25
Auto-tuning PID	150	230	0
PID-neural Controller	130	200	2,5

It is noteworthy that the two methods proposed to stand in the second and third rows in the table give step responses with negligible overshoot, faster setting time and shorter rise time than conventional PID.

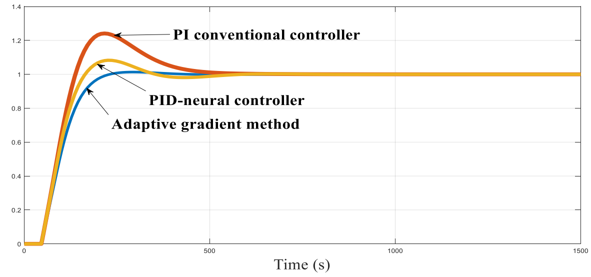

Fig. 9: Responses to a step function with the change of the transfer function of the plant in the auto-tuning system

If the transfer function of the plant in the auto-tuning system within a range of K_1 to K_2 and T_1 to T_2 changes, the response of the system changes insignificantly. Therefore, the auto-tuning algorithm can be applied to a class of the first order inertia systems with delayed time because the transfer function of the object in the auto-tuning system has little effect on the system's response. In fact, many industrial devices can be used with these transfer functions. Therefore, this controller is widely used.

The coefficients of the conventional PID controller are the initial conditions of the auto-tuning PID controller. Comparing *Fig. 8* and *Fig. 9*, the velocity response of the proposed auto-tuning PID control scheme is similar that of the conventional PID control scheme. The conventional PID control scheme has a larger tracking error than the proposed auto-tuning PID control scheme.

B. Experimental results

An auto-tuning PID control strategy, based on a deduced model, is proposed for implementing a thermal process. Identification of resistor furnace model:

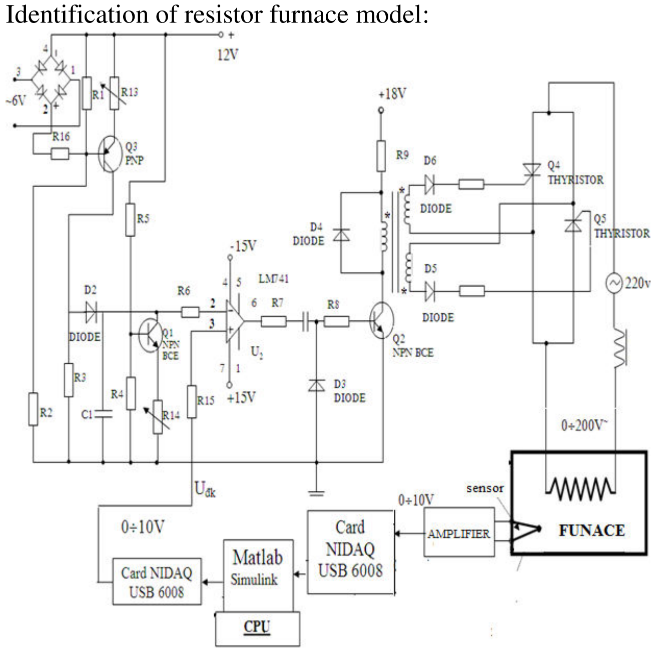

Fig 10: Block diagram of experimental setup

The system uses NIDAQ USB-6008 card to output the control voltage to the single-phase AC-AC converter, thereby changing the voltage supplied to the furnace. The temperature in the furnace is measured by the E type thermocouples (sensor), after via the buffer amplifier is also taken to the NI Card, which converts into digital signals and sends temperature data to computer for the identification of the resistor furnace model:

$$G_{(S)} = \frac{0.578}{452s+1}e^{-45s} \qquad (10)$$

Fig. 11: Snapshot of Real plant

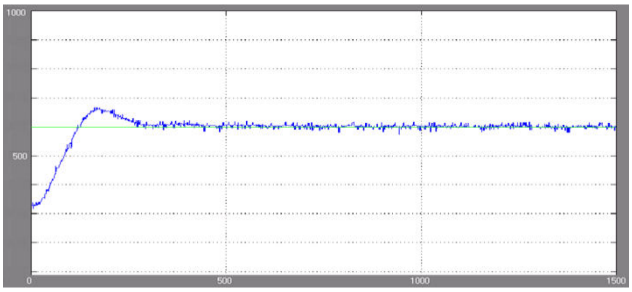

Fig. 12: Experimental results of auto-tuning PI controller.

V. CONCLUSION

The Gradient method makes it easy to find controller parameters but only converges in a narrow value domain. It is difficult to check the stable area of the system, especially for higher order systems, which cannot determine the exact stability area.

An auto-tuning PID control strategy, based on a deduced model, is proposed for implementing a thermal process. The controller parameters are tuned automatically, on-line, to overcome the disturbances and parameter variations. Experimental results are presented to demonstrate the reliability and effectiveness of the proposed control scheme in improving the system performance and reduce its sensitivity to parameter variations and load disturbance. The proposed control schemes are simple and practical to implement, assure system stability and have relative lower requirements for precision of controlled objects. Compared with traditional PID controller, the PID-neural controller has the characteristics of adaptive, on-line adjustment, better robust on the plant. Therefore, it is not necessary to identify dynamics of the plant because of the controller's learning ability. It has a good application in industrial control processes.

REFERENCES

[1] Ziegler, J.G., Nichols, N.B, "Optimum settings for automatic controllers", Trans. ASME, 1942, 64, pp. 759–768.

[2] Hang, C.C., Astrom, K.I., Ho, W.K, "Refinements of the Ziegler–Nichols tuning formula", IEE Proc. D, Control Theory Appl., 1991, 138, (2), pp. 111–118.

[3] Gorez R. and G. Calcev (1997):"A survey of PID auto-tuning methods" Proc. I Ith Conference on Control Systems and Computer Science. Vol. I. pp. 18-27.

[4] Åström, K.J. and T. Hägglund, 1988. Automatic Tuning of PID Controllers. Instrument Society of America, Research Triangle Park.

[5] Astrom K.J.T. Hagglund, C. C. Hang, and W. K. WO (1993):"Automatic tuning and adaptation for PID controllers survey "Control Engineering Practice, Vol. I, pp. 699-714.

[6] Aguirre, L.A, "PID tuning based on model matching", Electron. Lett, 1992, 28, (25), pp. 2269–2271.

[7] Zhuang, M., Atherton, D.P, "Automatic tuning of optimum PID controllers", IEE Proc. D, Control Theory Appl, 1993, 140, (3), pp. 216–224.

[8] Yu Yongquan, Huang Ying and Zeng Bi, "A PID Neural Network Controller," Proceeding of the International Joint Conference on Neural NetWorks, IEEE Copmuter Society Press,California, vol. 3, pp. 1933-1938, 2003.

[9] Iwasa,T.,Morizumi,N.andOrmatu,S. "Temperature Control in a Batch Process by Neural Networks,"Proceeding of IEEE World Congress on Computational Intelligence, IEEE Press,New York, vol. 2, pp. 992-995, 1992

[10] Michael Ayomoh, Mosud Ajala, "Neural Network Modeling of a Tuned PID Controller" European Journal of Scientific Research ISSN 1450-216X Vol.71 No.2 (2012), pp. 283-297.

[11] Andrasik A. Mészáros and De Azevedob S. F. (2004), On-line tuning of a neural PID controller based on plant hybrid modeling, Computers and Chemical Engineering 28, pg. 1499–1509.

[12] Fang, M.C. Zhuo,Y.Z. and Lee, Z.Y. (2010), The application of the auto-tuning neural network PID controller on the ship roll reduction in random waves, Ocean Engineering, Volume 37, Issue 7, Pp. 529-538.

[13] Shu, H. and Pi, Y. (2005), Decoupled Temperature Control System Based on PID Neural Network, ACSE 05 Conference, CICC, www.icgst.com, Cairo, Egypt. Pp.107-111.

[14] Robert D. Brandt, Feng Lin, "Adaptive Interaction and Its Application to Neural Networks", Elsevier, Information Sciences 121 (1999) 201–215

MSc Nguyen Trong Toan received his bachelor degree of Control Engineering from Thai Nguyen University of Technology (TNUT), Vietnam in 2009, and master degree of Automatic Control from Hanoi University of Science and Technology (HUTS), Vietnam in 2012. He is currently working as a lecturer and researcher at Faculty of Electronics Engineering, TNUT. His research interests are Automation, Control Systems, robotics, adaptive control, intelligent systems and their industrial applications.

MSc Nguyen Nam Trung obtained his bachelor degree of Electrical Engineering from Thai Nguyen University of Technology (TNUT), Vietnam in 1992, and master degree of Automatic Control from Hanoi University of Science and Technology (HUTS), Vietnam in 2007. Currently he is a lecturer at Faculty of Electronics Engineering, TNUT. His research interests are optimal control, intelligent systems, motion control, and robotics.