Neural Network Adaptive Autotuner

Alireza Rahrooh, Bahman Motlagh
University of Central Florida

Abstract

It is critical that modern control theory techniques be integrated into assignments which involve the application of basic concepts in engineering technology to prepare students for the next millennium. The adaptive neural network discussed in this paper can be viewed as an appropriate use of these modern techniques in engineering technology curriculum. Adaptive tuning of PID controller gains in case of plant parameter variations is of great importance. There are many approaches available for PID autotuning. In this paper the PID controller gains are adaptively changed using a neural network approach. The neural network tuner is incorporated in the control system to adapt the PID gains to changing system parameters. The neural network architecture employed is a multilayer perceptron. A computer simulation is conducted to show the tracking behavior of the controller in the case of plant parameter variations and set point changes.

Introduction

Proportional-Integral-Derivative (PID) controllers are the most widely used controllers in the industry. This wide applicability is due to their simple structure and robust behavior. The proper choice of the PID gains is essential for a satisfactory performance. Many attempts for tuning these controllers are mainly achieved by off-line tests performed on the plant¹⁻³. There are also other more modern autotuners available⁴⁻⁷. The controller parameters, regardless of their method of tuning, must be retuned after a change of plant parameters due to changing operating points, aging, etc. Therefore, the design of an autotuner that adaptively changes the controller gains could be of practical interest.

In this paper, neural networks are utilized to design a tuner that changes the PID gains adaptively. Neural networks have been previously used as adaptive controllers to control unknown plants⁸⁻¹⁰. The main idea developed in this paper is to use neural networks as adaptive autotuners rather than as an adaptive controller. Simulation results are presented to show the performance of this autotuner in the case of plant parameter variations and set point changes.

The Neural Network Structure

A neural network which is widely used in the control applications is the feedforward network with two hidden layers. The learning algorithm used in the present network is backpropagation¹¹. A problem with this network is its rather long learning time. In order to reduce this time, a linear neuron is used in the outer layer and sigmoid neurons are used in hidden layers, respectively. Figure 1 shows such a network (S is sigmoid neuron, and L is linear neuron). The required learning time is considerably reduced by accelerating the convergence rate and reducing the initial value of the norm of the error vector. To further improve the performance of the network the momentum algorithm is

used in the network structure ¹². This will smoothen the behavior of the network and will prepare it to tackle the effects of error bursting.

The expression for updating the matrix of weights in the neural network with the momentum algorithm is as follows:

$$w_{k+1} = w_k + 2\beta e_k x_k + \alpha(w_k - w_{k-1}) \quad (1)$$

where w_k is the matrix of weights at the k th iteration, e_k is the error backpropagated, x_k is the input to the network, β is the learning factor, and α is the momentum factor.

Figure 1. Modified neural network used in the control architecture

The Controller Structure

Consider a unity feedback system of Figure 2 with known transfer function $G(s)$ and a digital PID controller given by:

$$u_k = T k_p e_k + T k_i z_k + T k_d (e_k - e_{k-1}) \quad (2)$$

where e_k is the error between the output y_k and the set point r , z_k is the digital integral of error given by:

$$z_k = z_{k-1} + e_k \quad (3)$$

T is the sampling time, and k_p , k_i , k_d are the proportional, integral, and derivative gains, respectively. Each of the adaptive gains shown in Figure 2 can be treated as a single input single output modified feedforward network.

Figure 2. Control system

In the backpropagation structure the desired output of the network is used to form the required error for backpropagation. However, it is noted that in the present application the only desired value available is the desired output of the plant. Therefore, in order to provide an error for backpropagation in the networks, $G(s)$ is assumed as a layer of the network¹⁰. This will allow the error in the output of the system $y(t)$, to be the backpropagated error. It should be clear that this additional layer would not be updated during the weight updating phase of the network. **Partial derivative of the error at the output of the plant with respect to weights, i.e., $\partial e_{op}/\partial w$, must be used in the backpropagation algorithm.** The expressions for this algorithm are based on the **partial derivation of the error at the output of neural network $\partial e_{on}/\partial w$.** **The following relationship solves the problem by relating these two terms:**

$$\partial e_{op}/\partial w = \partial y/\partial y_n * \partial e_{on}/\partial w \quad (4)$$

where y is the output of the plant and y_n is the output of the neural network. Since we want to control an unknown plant, $\partial y/\partial y_n$ is **not available, therefore, the following algorithm can be used to obtain an approximate value:**

$$\partial y/\partial y_n = (y_k - y_{k-1}) / (y_{nk} - y_{nk-1}) \quad (5)$$

Simulation

A simulation study is conducted to illustrate the utility of the proposed technique. The results show an excellent tracking behavior of the controller in the case of plant parameter variations and set point changes. In the following examples, the effectiveness of the proposed structure can be seen

Example 1. This example shows the performance of the control system when the plant is a first order plant with transfer function $G(s) = 1/(s+1)$. Sampling time is 0.1 sec and the neural networks have four neurons in each of their hidden layers. The actual and desired outputs of the plant are shown in Figure 3. At $t=10$ sec., set point is changed and the output of the plant tracks it. At $t=20$ sec., the transfer function of the plant is changed to $G(s) = 2/(s+3)$. It can be seen in Figure 3 that the neural part adaptively changes the gain for the new plant after a few iterations.

Example 2. In this example we want to control a liquid mixing process which can be represented by a second order transfer function $G(s) = 5/(s^2+4s+20)$. Figure 4 shows the input (desired output) and the actual output of the process. AT $t=20$ sec., the transfer function of the process is changed and it can be seen from Figure 4 that a proper tracking occurs.

Conclusion

It has been demonstrated that that using advanced concepts from modern control theory can improve the tuning of PID controller. Furthermore, it is essential that some of these techniques be incorporated into engineering technology curriculum to prepare students for 21st century technology.

Figure 3. Actual and desired output of the plant

Figure 4. Actual and desired output of the process

References

1. Ziegler, J.G., and Nichols, N.B. "Optimum setting for automatic controllers," *Trans. ASME*, 1942, No. 65, pp. 433-444.
2. Deshpande, P.B., and Ash, R.H., "Computer control," *ISA Pub*, 1981.
3. Hang, C.C., and Astrom, K.J., and Ho, W.K., "Refinements of the Ziegler-Nichols tuning formula," *IEEE proceedings*, Vol. 138, No.2, 3/1991.
4. Kraus, T. W., and Mayron, T.J., "Self-tuning PID controllers based on a pattern recognition approach," *Control engineering*, 1984, pp. 106- 111.
5. Astrom, K. J., and Hagglund, T. , "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, 1984, No, 20, pp. 645-65 1.
6. Porter, B., "Issues in the design of intelligent control systems," *IEEE Control systems magazine*, 1/1989, pp. 97-99.
7. Porter, B., and Jones, A.H.' "Genetic tuning of digital PID controllers," *Electronic letters*, Vol. 28, No. 9, pp. 843-844, 1992.
8. Psaltis, D., Sideris, A., and Yamamura, A.A., "Multilayered neural network controller," *IEEE control systems magazine*, 1988.
9. Narendra, K.S., and Parthasarathy, K., "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, 1990, Vol. 1, No. 1, pp. 4-27.
10. Chen, C. S., "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE control system magazine*. 4/1990.
11. Rumelhart, D., McClland, J., "Parallel distributed processing," *MIT Press*, Vol. 1, 1986.
12. Roy, S., and Shynk, J.J., "Analysis of the momentum LMS algorithm," *IEEE Trans. assp*, Vol. 38, No. 12, 12/1990.

ALIREZA RAHROOH

Alireza Rahrooh is Assistant Professor of Engineering Technology at the University of Central Florida. He received his B.S., M.S., and Ph.D. in Electrical Engineering from the University of Akron. Dr. Rahrooh was previously on the faculty at the Penn State University at Harrisburg. He has many papers published in different area of Electrical Engineering. He is a reviewer for Journal of ET, and a member of ASEE, IEEE, ISCA, and IASTED.

BAHMAN MOTLAGH

Bahman Motlagh is Assistant Professor of Engineering Technology at the university of Central Florida. He received his B.S. in Engineering from Istanbul Academy of Sciences, M.S. and Ph.D. in Computer Engineering from University of Central Florida. He is a member of IEEE, ASEE, and Eta Kappa Nu.