Process Control Using a Neural Network
Combined with the Conventional PID Controllers

Moonyong Lee and Sunwon Park

Abstract: A neural controller for process control is proposed that combines a conventional multi-loop PID controller with a neural network. The concept of target signal based on feedback error is used for on-line learning of the neural network. This controller is applied to distillation column control to illustrate its effectiveness. The result shows that the proposed neural controller can cope well with disturbance, strong interactions, time delays without any prior knowledge of the process.

Keywords: process control, neural network, distillation control, target signal, feedback error learning

I. Introduction

Many chemical processes are quite difficult to control due to large time delays and lags, strong interactions and nonlinearities. A typical approach to process control involves the construction of a mathematical model describing the dynamic system to be controlled and the application of analytical techniques. But this approach often fails because the process model is rarely available or often very inaccurate due to uncertainty or complexity. Neural network techniques have recently received widespread attention to overcome these difficulties in process control and offered some promising results[1]-[6].

When we apply a neural network to process control, we should consider or solve two problems. One is that we can not explicitly provide target outputs for the neural network because in control situations the target outputs correspond to the appropriate control inputs that give desired responses. This is an inevitable problem due to the assumption that process dynamics is unknown *a priori*. The other is the peculiar characteristics chemical process have. In robot system, once robots are made in factories they can be fully trained before field implementation. Thus both the in-operability during training and the learning speed would not serve as critical factors. This plays a significant attractive role in application of the neural network on robot control. On the other hand, in chemical processes this is certainly not the case. It is practically impossible for a neural controller to be pre-trained before plant construction. Moreover, even after plant implementation the learning range should be restricted by operational safety. Therefore in process control applications, it is highly desirable that control and training are done simultaneously.

In this paper, we present a control scheme using a neural network for process control applications. The neural network alone might be used directly as a controller, but this approach has several drawbacks: first, during the training period, the control system is not operational; second, it cannot eliminate unpredictable disturbances; and last, this approach bears a less direct connection to the design methods for traditional controller. To avoid these problems, in the proposed scheme, the conventional multi-loop PID controller is combined in parallel with the multi-layer feedforward neural network. The simulation study for distillation column control are carried out and various properties of the controller are tested.

II. Proposed control scheme

Among several architectures for neural network based control, the feedback error learning scheme by Kawato et al.[5] and the disturbance error learning scheme by Lee and Park[6] have a lot of good features for process control: first, backpropagation of the error signal through the controlled plant is not necessary at all because the feedback error is used as the error signal; second, the process can be controlled even during the training period. We modified their schemes to be aimed at the process control applications using a conventional multi-loop PID controllers. The architecture of the neural controller proposed in this paper is shown in Fig. 1. In the proposed scheme, disturbances, manipulated variables, controlled variables, and setpoints are used as input variables for the network. This scheme can handle regulatory problems, which are most important in process control, as well as servo problems. By using the conventional multi-loop PID controller in parallel with the neural network, the control scheme offers several important advantages compared with the case of neural network alone: the process could remain its flexibility and operability by the PID controller even when the neural network is inoperable; the neural network alone could not guarantee zero off-set at steady state against unpredictable random uncertainties even when the network is well trained. Although any conventional controller which can compensate the feedback error can be used with the neural network, the controller with integral action is preferable for zero off-set condition at steady state.

In the proposed scheme, the error backpropagation algorithm[7] is chosen to train the neural network. Thus, the connection weight between the ith neuron in the $(l-1)$th layer and the jth neuron in the lth layer, W^l_{ij}, at the $(k+1)$th learning step is adjusted in the steepest descent manner as follows:

$$W^l_{ij}(k+1) = W^l_{ij}(k) + \eta \Delta^l_{ij} \tag{1}$$

where Δ^l_{ij} is

$$\Delta^l_{ij} = (T^l_{pj} - O^l_{pj})\ F'(\sum_{i=1}^{N^{l-1}} V^l_{ij}\ O^{l-1}_{pi})O^{l-1}_{pi}$$

$$= \delta^l_{pj}\ O^{l-1}_{pi} \quad \text{for the output layer, i.e, } l=L \qquad (2)$$

$$\Delta^l_{ij} = \sum_{k=1}^{N^{l+1}} \delta^{l+1}_{pk}\ W^{l+1}_{jk}\ F'(\sum_{i=1}^{N^{l-1}} V^l_{ij}\ O^{l-1}_{pi})\ O^{l-1}_{pi}$$

$$= \delta^l_{pj}\ O^{l-1}_{pi} \qquad \text{for the hidden layers, i.e, } l<L \qquad (3)$$

where $F'(x)$ is the derivative of the activation function with respect to x, and η is the learning coefficient.

We would like to modify the weights in the network so that the square error $\|\mathbf{R\text{-}Y}\|_2$ will be less at the end of the next run. To train the network, we need to know the target output, which minimizes the square error $\|\mathbf{R\text{-}Y}\|_2$, of the network. However, unfortunately, only the error in the final plant state, $(\mathbf{R\text{-}Y})$, is available. To avoid this problem, we introduce the concept of "*target signal*". The target signal is similar with the target output in the sense that the neural network adjusts its weights according to the target signal. The neural network compares its output with the target signal instead of the target output which can not be known *a priori*. In the proposed scheme, the sum of outputs by the network and outputs by the proportional plus derivative action in the PID controller is used as the target signal for on-line learning. The target signal at the kth sampling time step, $T^l_j(k)$, is the sum of outputs at the $(k\text{-}1)$th sampling time step, and is described as:

$$T^l_j(k)=U_{nj}(k\text{-}1)\ +\ U_{pj}(k\text{-}1)\ +\ U_{dj}(k\text{-}1). \qquad (4)$$

Note that the integral action from the PID controller must be excluded from the target signal because it results in the double integral action. The target signal $T^l_j(k)$ is then compared to the output of the neural network at the $(k\text{-}1)$th sampling time step, $O_j(k\text{-}1)$. Thus, at the kth time step, δ^l_j for the output layer in Equation (2) becomes

$$\delta^l_j(k) = \{U_{pj}(k\text{-}1)\ +\ U_{dj}(k\text{-}1)\}F'(\sum_{i=1}^{N^{l-1}} V^l_{ij}\ O^{l-1}_i)\ . \qquad (5)$$

This target signal is different from the target output in the sense that it does not always give a desired response. Instead, initially the target signal may quite differ from the desired target output. But it gradually approaches the target output when learning is successfully accomplished. Since the proposed target signal provides the correct gradient direction for the network training, learning is achieved in such a way that the square error $\|\mathbf{R\text{-}Y}\|_2$ is minimized. We wish to train the neural network with the proposed target signal so that the sum of the outputs by proportional and derivative actions of the PID controller is minimized. Once the neural network is successfully trained, the performance of the controller is naturally improved. The better learning is achieved, the better the process is controlled because $U_p + U_d$ is closer to zero. It should be noted that, while backpropagation can be proven to implement gradient descent for the desired input/output

mapping in static cases, the same does not necessarily hold for dynamic cases. The learning algorithm of the proposed neural controller is, at best, a heuristic for applying the neural network to a class of control problems. No theoretical analysis of convergence yet exists. It, however, is clear that no stability problems are expected as long as the learning rate is sufficiently slower than the time constants of the other components of the control system, as mentioned by Psaltis et al.[4].

In Fig. 1, at each time k, all of the controlled variables Y(k), manipulated variables U(k), disturbance D(k), setpoints R(k) are measured. The conventional controller outputs U_c(k) are then computed. Not only current but also dominant past information on state is necessary for considering a dynamic relationship between input and output patterns. The buffer and pre-processor(BPP) module plays a role of storing and scaling those signals. After one learning step is performed, the network receives past and current values of Y, D, R and U as the input signals from the BPP module and produces network outputs U_n(k). U_n(k) are then added to U_c(k) to be applied to the process. This entire process is repeated at each sampling time.

Initially the neural network has little influence over the control action and most control action is performed by the PID controller. As learning proceeds, the neural network tries to conFig. itself so that the outputs of the PID controller are as small as possible. Therefore, finally, most control action is in turn carried out by the neural controller instead of the PID controller. Since the error signal is the input to the feedback controller, the training of the network will lead to a gradual switching from feedback to feedforward action as the error signal becomes small. During training, features of the plant that are initially unknown not taken into account by the control algorithm are learned. Both system identification and process control are done by the network simultaneously. An immediate consequence of the increased use of feedforward control action is to speed up the response of the system.

III. Application to distillation column control

To evaluate the proposed neural controller, the well known model by Wood and Berry[8] for their methanol-water distillation column was chosen. The model is given by:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21.0s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.8e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} \dfrac{3.8e^{-8s}}{14.9s+1} \\ \dfrac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} d \qquad (6)$$

The controlled variables Y_1 and Y_2 are the overhead and bottom methanol compositions, respectively. The manipulated variables U_1 and U_2 are the reflux and steam rates, respectively. The disturbance D is the feed rate. The steady state values of the overhead and bottom methanol compositions are 96.25 mol % and 0.5 mol %, respectively. Parameters of the multi-loop PID controller used were taken from the values found by the original authors[8] as $K_{c1}=0.2$, $K_{c2}=-0.04$, $\tau_{I1}=4.44$, $\tau_{I2}=2.67$. A sampling period of 3 min

was used. Fig. 2 shows the BPP module and the specific configuration of the network employed in this work. A three-layered network was chosen. The input layer contains 27 neurons and receives signals comprised of Y_i(k-n), U_i(k-n), R_i(k-n), R_i(k-n+1), and D(k-n), where n=0,1,2, and i=1,2. The hidden layer has 10 neurons. The output layer has 2 neurons and produces the controller signals as its outputs. The weights of the network are chosen initially with small random numbers. All of the neurons except those in the input layer have the hyperbolic tangent activation function as α $^{\beta}$ -e$^{-\beta}$)/(e$^{\beta}$ +e$^{-\beta}$). The parameter α is closely related to the constraints of actuators. In our simulation, since we arbitrarily assumed the constraint of each actuator as +0.2 lb/min from their steady-state values, the value of 0.2 was used. The parameter β was empirically set as 10 and the learning coefficient η 0.4 was used.

Simple training patterns lasting for 900 min, which include consecutive random step changes in each setpoint and also consecutive disturbances in the feed flow, were repeated until a desired performance was accomplished. For the purpose of comparison, the Integral Square Error (ISE) for one training cycle was used as the performance measure. The ISEs both in bottom composition and in top composition decreased significantly during 110 training cycles.

Fig. 3 and 4 show the servo and regulatory control behaviors both of the trained neural controller and the untrained neural controller to the inexperienced consecutive changes in setpoints and disturbance, respectively. As shown in the Fig.s, the neural controller after training performs tasks of both servo tracking and disturbance rejection well and shows remarkable improvement in performance compared to the conventional controller alone. Note that the overall control action of the neural controller with untrained network is the same as that of the conventional PI controller because the network is initially set so that it has little influence over the manipulated variables U ,i.e. $U_n \cong$

Although the overall control action U is a simple sum of the output by the PI controller U_c and the output by the network U_n , these two play totally different roles in controlling the process. Fig. 5 shows how to output by the PI controller U_c and the output by the network U_n act in the trained neural controller. As the neural network adapts the system dynamics, the portion of the control signal generated from the network takes over the control of the system. The result shown in Fig. 5 confirms that most control action is performed by the neural network and only a relatively small

portion of action for feedback error is achieved by the PI controller.

It is also demonstrated through the extensive simulation study that the proposed neural controller has many other desirable features such as the natural learning capability by random input pattern, robustness against fault in connection weights, and adaptability for system parameter changes.

IV. Conclusions

In this paper, we present a new control scheme combining the neural network with the conventional multi-loop PID controller to aim for process control applications. The proposed neural controller is applied to the well known distillation column system which has significant interactions and time delays and lags. The result shows that the proposed control scheme gives the superior performance both to servo and regulatory problems with many desirable properties. The proposed neural controller appears to have the potential to deal with complex process control problems.

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can difficult learning control problems." *IEEE Trans. Syst. Man, Cybern.* SMC 13, No. 5, pp. 834-846, 1983.

[2] W. Miller, "Sensor-based control of robotic manipulators using a general learning algorithm." *IEEE J. Robotics and Automation* RA-3, no. 2, pp. 157-165, Apr., 1987.

[3] A. Guez and J. Selinsky, "A neuromorphic controller with a human teacher." *IEEE Int. Conf. on Neural Networks,* San diego, California, 1988.

[4] D. Psalitis, A. Sideris, and A. A. Yamamura, "A multilayered neural network controller." *IEEE Control Systems Magazine,* April, pp. 17-21, 1988.

[5] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchial neural network model for voluntary movement with application to robotics." *IEEE Control Systems magazine,* April, pp. 5-16, 1988.

[6] M. Lee and S. Park, "A new scheme combining neural feedforward control with model-predictive control," *AIChE J.,* vol. 38, pp. 193-200, 1992.

[7] R. Hecht-Nielsen, "Theory of the backpropagation neural network." *Proc. IEEE 1989 Intl. Conf. Neural Networks,* pp. 593-605, 1989.

[8] R. K. Wood and M. W. Berry, "Terminal composition control of a binary distillation column," *Chem. Eng. Sci.* 28, pp. 1707, 1973.

Moonyong Lee

He (1959) received the B.S. degree in chemical engineering from Seoul National University in 1982 and M.S. and Ph. D. degrees in chemical engineering from KAIST in 1984 and 1991, respectively. He had been with SK company as a senior process engineer from 1984 to 1994, and was with ETH in Switzerland as a visiting professor in 1997. He is now an associate professor in the school of chemical engineering and technology in Yeungnam University. His research interests are process control and identification, nonlinear dynamics, process modeling and design.

Sunwon Park

He(1948) received the B.S. degree in chemical engineering from Seoul National University in 1970 and M.S. and Ph. D. degrees in chemical engineering from Oklahoma State University in 1974 and University of texas at Austin in 1979, respectively. He received MBA from University of Houston Clear lake in 1987. He had been with Hoechst Celanese as a staff engineer from 1979 to 1988. He is now a professor in the department of chemical engineering in KAIST. His research interests are process control, design, and optimization.