

CONTROL OF A PROCESS WITH TIME DELAY BY POLICY-AND-EXPERIENCE-DRIVEN NEURAL NETWORKS

MASARU ISHIDA AND JIXIAN ZHAN

Research Laboratory of Resources Utilization, Tokyo Institute of Technology,
Yokohama 227

Key Words: Process Control, Neural Network, Time Delay, Self Tuning, Global Policy Learning, Local Experience Learning

Introduction

Process control with large time delay has been a challenging problem, on which much work has been done, but it remains an obstacle to process control. In recent years, neural network control has achieved rapid development. Because of its massive parallel processing, nonlinear mapping, and self-learning abilities, the neural network is considered a promising way to deal with many difficult control problems.

For the guarantee of global reliability and local tracking ability of a controller, one of the authors proposed a PENN (Policy-and-Experience driven Neural Network) method and applied it to control water level.¹⁾ Simulation results showed that this method is a promising way to control nonlinear and dynamic processes and that even a network without hidden layer can be applied with significant toughness to noise.²⁾

In this paper, we examine the possibility of applying the PENN method to nonlinear water-level control with large time delay. This time delay is assumed to be constant. It is also assumed that the vessel configuration is known but that the detailed process characteristics are unknown. Hence no process model is available. Two neural networks are utilized. One is for prediction of future value of the controlled variable and the other is to determine the control variable based on the predicted future controlled variable. The results show that this method is of great promise.

1. Controlled Process and Controller

The controlled process is shown in Fig. 1(b). The cross-sectional area of the vessel a is expressed as

$$a = \pi r^2 = \pi [y \tan(\phi/2)]^2 \quad (1)$$

The relation between the exit flow rate q_{out} and the water level y is as follows:

$$q_{out} = \alpha \sqrt{y} \quad (2)$$

We set the parameters $\phi = 60^\circ$ and $\alpha = 60 \text{ cm}^{5/2}/\text{s}$.

In the previous work²⁾ the same problem was examined, but the controlled system had only a very short time delay and hence only one network was applied. In this study with very large time delay, two networks, a prediction network and a control network, are applied as shown in Fig. 1(a).

The prediction and control networks are shown in Figs. 1(c) and (d). The subscripts j and d respectively represent the present time and the time delay given as dead time/sampling interval. The variables are defined as follows:

$$A_j = a_j/a_{\max} \quad (3)$$

$$e_j = y_{\text{target}} - y_j \quad (4)$$

$$E_j = e_j/e_{\max} \quad (5)$$

$$dE_j = E_j - E_{j-1} \quad (6)$$

$$U_j = u_j/u_{\max} \quad (7)$$

$$dU_j = U_j - U_{j-1} \quad (8)$$

In this study we treat a constant time delay of 200 seconds, i.e. $d=10$ for the sampling interval $dt=20 \text{ s}$. E_{j+d+1} in Fig. 1(d) is the expected value of E at the time of $d+1$ sampling intervals after the present time j . In this study it is calculated as follows:¹⁾

$$E_{j+d+1} = E_{j+d}/(1 + dt/T_1) \quad (9)$$

where dt denotes the sampling interval and T_1 specifies

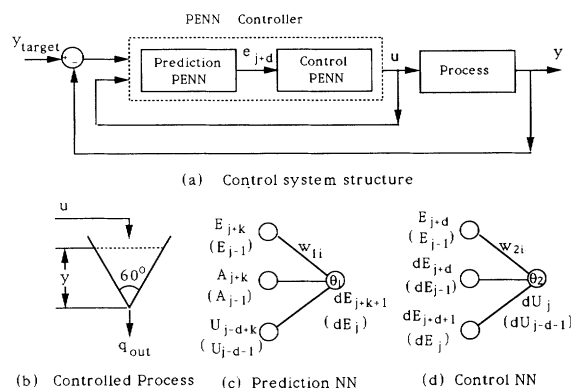


Fig. 1. Scheme of control system and process

* Received August 5, 1992. Correspondence concerning this article should be addressed to M. Ishida.

the change in speed of height.

Prediction and control are carried out as shown in Fig. 2. The target is to determine the control variable U_j based on present E_j . For this purpose, we predict the controlled variable at time $j+d$, E_{j+d} , stepwisely. That is, we first predict the value E_{j+1} at time $j+1$ (corresponding to $k=0$ in Fig. 2) based on E_j , A_j and U_{j-d} . Then, based on the predicted value at time $j+1$, we predict the variable at time $j+2$. This is repeated until $j+d$ (i.e., $k=d-1$). The predicted controlled variable E_{j+d} , its change dE_{j+d} and the expected change dE_{j+d+1} are used as inputs to the control network, which is the same as that used previously.²⁾ The control network then gives dU_j , sending the new value U_j to the process and we proceed to the next sampling interval.

2. Pretraining of Networks

2.1 Prediction Network

We assume that some data for the relation between the input u and the output y of the process have been obtained by manual operation or from previous runs. The solid lines in Fig. 3 show such data used in this study.

The learning is accomplished by the back-propagation algorithm. For this learning, E_{j-1} , A_{j-1} and U_{j-d-1} are inserted as inputs and $dE_j (=E_j - E_{j-1})$ as output, as shown in parentheses in Fig. 1(c). All are known values for each sampling interval. During eight times of such local learning, the following policy learning will be accomplished once:

(1) If $E_{j-1} = -1$, $A_{j-1} = 1$ and $U_{j-d-1} = U_{\text{initial}}$ then $dE_j = 0$

(2) If $E_{j-1} = 1$, $A_{j-1} = 0$, and $U_{j-d-1} = 0$ then $dE_j = 0$
 U_{initial} is the inlet flow to keep the water level at 80 cm, and hence the first policy indicates the initial state, while the second policy represents the condition for no water feed to a vacant vessel.

By such learning, the weights w_{1i} and the threshold Θ_1 are renewed at each 20s time interval, which corresponds to the sampling interval, and based on the renewed weights the future y at 200 s (=delayed time) ahead is predicted by the prediction procedure shown in Fig. 2. The results are shown by dotted lines in Fig. 3. This pretraining procedure is repeated for 200 running times.

For the first running time, the predicted values given by dotted line 1 deviated significantly from the solid-line real data. As the learning continued, the dotted prediction lines approached the solid line gradually. The weights w_{1i} and the threshold Θ_1 after 200 running times are used as initial values for the prediction network for the control run.

2.2 Control Network

Pretraining of the control network is accomplished by teaching the following five global policies, which

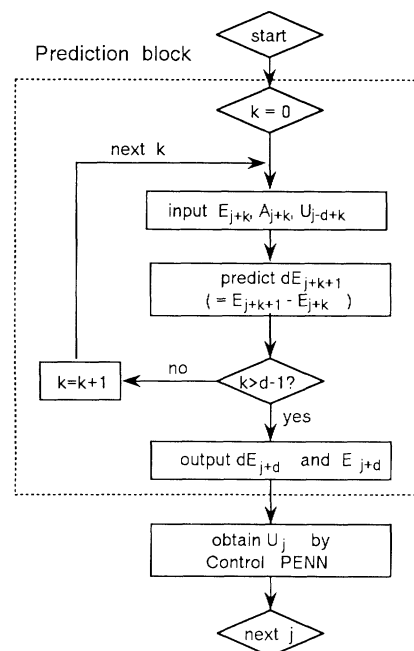


Fig. 2. Algorithm to predict future controlled variable

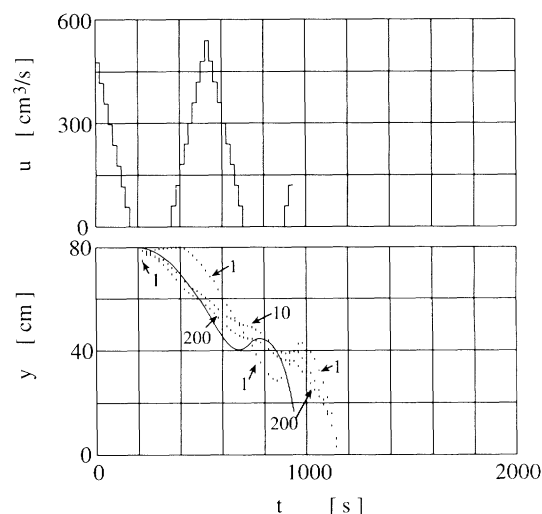


Fig. 3. Pretraining of prediction network without hidden layer

were also used previously,^{1,2)} for 80 times:

(1) If $E_{j-1} = 0$, $dE_{j-1} = 0$ and $dE_j = 0$ then $dU_{j-d-1} = 0$

(2) If $E_{j-1} = 1$, $dE_{j-1} = 0$ and $dE_j = 1/(1 + dt/T_1) - 1$ then $dU_{j-d-1} = 1$

(3) If $E_{j-1} = 0$, $dE_{j-1} = -dE_{\text{max}}$ and $dE_j = 0$ then $dU_{j-d-1} = -1$

(4) If $E_{j-1} = -1$, $dE_{j-1} = 0$ and $dE_j = -1/(1 + dt/T_1) - (-1)$ then $dU_{j-d-1} = -1$

(5) if $E_{j-1} = 0$, $dE_{j-1} = dE_{\text{max}}$ and $dE_j = 0$ then $dU_{j-d-1} = 1$

3. Execution of Prediction and Control with On-Line Learning

The pretrained prediction and control networks are

used for process control. Also at this stage, on-line learning is continued. That is, we perform global learning and local learning for the prediction network as described in 2.1. Second, we predict E_{j+d} by applying the network repeatedly. Third, we perform global learning and local learning for the control network. For the latter learning, E_{j-1} , dE_{j-1} and dE_j are inserted as input and dU_{j-d-1} as output of the network, as shown in parentheses in Fig. 1(d). Fourth, we calculate dU_j by the control network. Fifth, we send U_j to the process.

Simulation is performed for the control of water level in Fig. 1(b). The water level is assumed to be at a steady state of 80 cm at the beginning. We set a new target level as 40 cm. We suppose the water feed rate to range from $u_{\min}=0$ to $u_{\max}=600 \text{ cm}^3/\text{s}$. We also set the upper bound of changing ability of the water feed rate $du_{\max}=60 \text{ cm}^3/\text{s}$. The sampling period is 20 s and after each sampling new U_j is calculated.

Figure 4 shows the control results when the running times were 1, 2, 10, and 100. It can be observed that the performance of the controller improved noticeably as learning and execution continued. After only several times of running, the control performance becomes nearly steady. This means that the learning speed of the networks is very fast.

The dotted lines in Fig. 4 are the values of y predicted by the prediction network and used as input to the controlled network. Although some deviation from the process output still remains, quite excellent control is achieved, even for a time delay as large as 200 s.

Figure 5 shows the case when on-line learning for the prediction network is not performed. Hence w_{1i} and Θ_1 are kept as the pretrained values. It is found that the water level does not approach the set point of 40 cm, indicating the importance of on-line learning of the global policy and local experience for the prediction network. The effect of the running time in Fig. 5 is similar to that in Fig. 4. Consequently, the effect of the running time on the control result observed in Fig. 4 is mainly caused by the training of the control network.

The networks shown in Fig. 1(c) and (d) have no hidden layers. It was demonstrated in the previous paper²⁾ that the simple network in Fig. 1(d) is quite strong for noise. Hence, a simple network without hidden layer is used for prediction, too. **Figures 6** and **7** show the results when two neurons are added as a hidden layer of the prediction network. Except for the existence of the hidden layer, the procedure for learning and the execution is exactly the same.

As illustrated in Fig. 6, addition of the two units as a hidden layer slows the learning speed. By 200 runs of pretraining the control performance shown in Fig. 7 is obtained. This shows that in the early runs

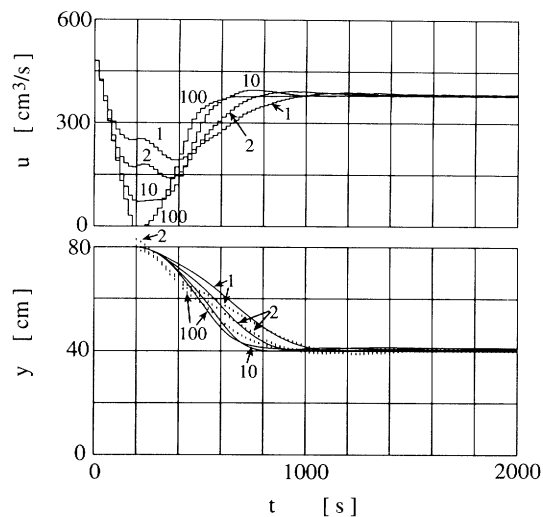


Fig. 4. Control performance with time delay of 200 s

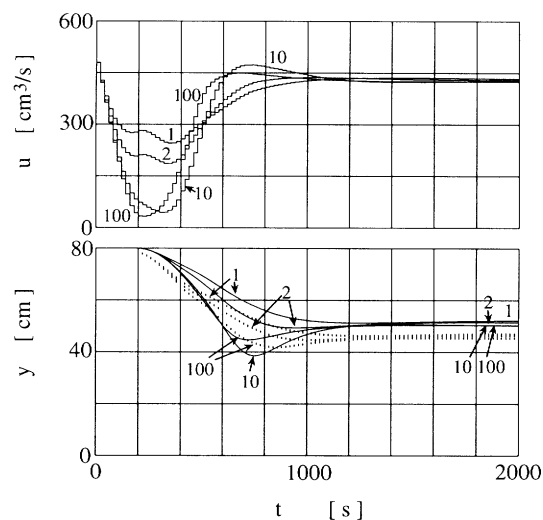


Fig. 5. Control performance when on-line training of prediction network is stopped

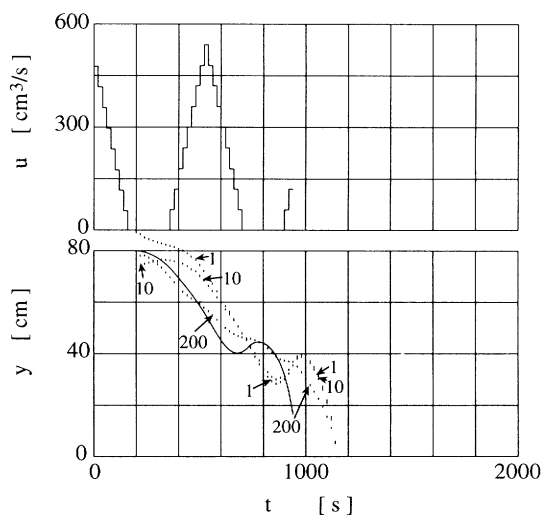


Fig. 6. Pretraining of prediction network with two units in hidden layer

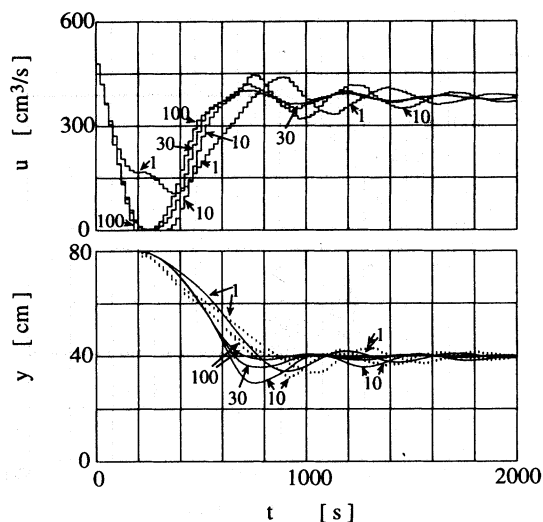


Fig. 7. Control performance when prediction network has two units in hidden layer

oscillation is significant, but that after 30 runs performance is satisfactory. The 100th run is quite close to the 100th run in Fig. 4. When the pretraining is increased to 400 running times, the control performance is found to have no oscillation and to become quite similar to that in Fig. 4. This indicates that about twice the learning is required for the network with two hidden units. And no merits have been seen for the addition of a hidden layer in this problem.

4. Discussion and Conclusion

The method adopted here is very straightforward. First the future value of the controlled variable is predicted by step-by-step application of the first network and then the control variable is obtained by the second network, based on the predicted value. For prediction of each E_{j+d} , the prediction network is applied 10 times because the time delay is 200 s and the sampling interval is 20 s. It is to be noted that even for such repeated application of the prediction network, the prediction error is kept small and the controller provides excellent control for such a large time delay. The advantages of the present method is the application of the smallest network containing only the key variables as inputs and output and usage of the PENN (Policy-and-Experience-driven Neural Network) concept. By making the networks as small as possible and dividing the controller into prediction and control networks, learning can be accomplished quite effectively.

Constant time delay is assumed in this study. If the time delay is a function of only the control variable, the present method may be extended promptly. For a time-varying time delay, estimation of time delay is required.

Literature Cited

- 1) Ishida, M.: *J. Chem. Eng. Japan*, **25**, 108 (1992).
- 2) Ishida, M. and J. Zhan: *J. Chem. Eng. Japan*, **25**, 485 (1992).