CONTROL BY A NEW POLICY- AND EXPERIENCE-DRIVEN NEURAL NETWORK TO FOLLOW A DESIRED TRAJECTORY

MASARU ISHIDA* AND TAKEHIRO OHBA

Research Laboratory of Resources Utilization, Tokyo Institute of Technology, 4259 Nagatsuta, Midoriku, Yokohama 227

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

Introduction

In previous papers[1, 2], the concept of a policy- and experience-driven neural network (PENN) was proposed to provide adaptive and self-tuning features for process control. Rule-based global policies give general control information covering the whole control space, whereas local experiences achieved by previous runs give detailed information about the process features. There the setpoint was assumed to be constant.

In this paper, the case where the setpoint is changing with time is discussed and a very simple network is proposed. The results for a level-control experiment by simulation show that this method has high potential for wide application.

1. Controlled Process and Control Strategy

1.1 Controlled process and PENN controller

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

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

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

$$q_{out} = \alpha\sqrt{y} \tag{2}$$

We set the parameters $\phi = 60°$ and α = 60 cm$^{5/2}$/s. We assume a constant time delay of 10 seconds, i.e., $d = 1$ for the sampling interval of 10s.

The PENN controller is shown in Fig. 1(c). It has a simple 2-input and 1-output structure. The input is dy_m, the difference between y at present and that at the previous sampling time, and dy_p, the difference between the expected setpoint y_{target} in the next specified future and y at present. The output is du, the difference between the controller output u to be obtained and the previous one. They are given as dimensionless values defined as follows:

$$\begin{aligned} &Y_j = y_j / y_{max} \\ &dY_m = dY_j = Y_j - Y_{j-1} \\ &dY_p = dY_{j+k} = Y_{target,\,j+k} - Y_j \quad (k \geq d+1) \\ &dU_j = (u_j - u_{j-1}) / u_{max,\,j} \end{aligned} \tag{3}$$

where j denotes the present time, and m and p, respectively, denote –, i.e., past, and +, i.e., future. A feature of PENN control is that the expected change of y is used as one of the control inputs. A new feature of this network is that the present error, $y_{target,\,j} - y_j$, is excluded from the input layer of the network.

1.2 Global policies and local experience

Another feature of PENN control is that both global policy and local experience are used to train the network. Based on some knowledge in advance, we make the following rules:

(1) IF $dY_m = 0$ and $dY_p = 0$, THEN $dU = 0$.
(2) IF $dY_m = 1$ and $dY_p = 0$, THEN $dU = -1$.
(3) IF $dY_m = 0$ and $dY_p = 1$, THEN $dU = 1$.
(4) IF $dY_m = -1$ and $dY_p = 0$, THEN $dU = 1$.
(5) IF $dY_m = 0$ and $dY_p = -1$, THEN $dU = -1$.

Rule (1) specifies the steady state. Rules (3) and (4) represent the cases when dU_{max} is outputted, and rules (2) and (5) are the inverse cases of rules (4) and (3). These global policies will contribute to the result in which the output dU_j is calculated in the network not by extrapolation but by interpolation of learned knowledge.

The following latest data are utilized as local control experience to train the network about the specific feature of the process:

INPUT: $dY_m = Y_{j-k} - Y_{j-k-1}$ and $dY_p = Y_j - Y_{j-k}$
OUTPUT: $dU = (u_{j-k} - u_{j-k-1})/u_{max,\,j-k}$

Since the term $Y_{target,\,j+k} - Y_j$ is an input unit, the value of Y_j can be used directly to specify the value of dY_p in the learning stage; hence effective learning can be achieved. The value of k is equal to or greater than 2, since the time delay is 10 s ($d = 1$). In this study it is set as 3. The effect of k will be demonstrated in detail in another paper.

* Received September 20, 1993. Correspondence concerning this article should be addressed to M. Ishida.

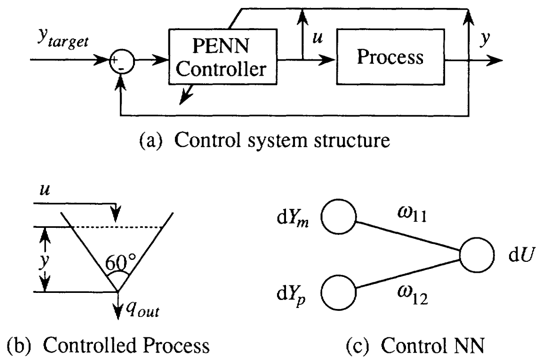

Fig. 1 Scheme of control system and process

2. Execution of Control with On-Line Learning

2.1 Desired trajectory

The above control scheme was examined by simulation of water-level control when the water level is to be changed from 80 cm to 40 cm by the following path.

$$
\begin{aligned}
&y_{max} = 80\ \mathrm{cm} && \\
&Y_{target} = 1 && \text{for } t < 100\ \mathrm{s}. \\
&Y_{target} = 1 - 0.001\,(t - 100) && \text{for } 100\ \mathrm{s} < t < 600\ \mathrm{s} \\
&Y_{target} = 0.5 && \text{for } t < 600\ \mathrm{s}.
\end{aligned} \tag{4}
$$

Hence the target water level decreases linearly during the period between 100 s and 600 s.

2.2 Control results

Figure 2 shows the result when control was exercised by setting the assumption that $u_{max,\,j}$ is proportional to the cross-sectional area of the conical vessel, i.e.,

$$u_{max,\,j} = \left(u_{max,\,j} \text{ at } Y_{max}\right) \times \left(y_j / y_{max}\right)^2. \tag{5}$$

The value of $u_{max,\,j}$ at Y_{max} was chosen as 600 $\mathrm{cm^3/s}$. Before the first run, the five global rules above were taught 80 times in advance. Then global learning of one of the above five rules and experience learning were performed for each sampling period. The sequence of the choice of rules for global learning was: 1, 2, 1, 3, 1, 4, 1, 5, indicating that the first rule for the steady state was taught frequently.

The results for the first, the 5th, the 10th and the 40th runs are shown. Even for the first run, the water level changed along the target trajectory. For the 5th run, the control is already quite satisfactory. It is found that the height y after the 10th run followed the given trajectory excellently.

Figure 3 shows the applicability of the same network to a different trajectory. After the 20th run in Fig. 2, the properties of the neural network were saved. By using this saved network, a run was executed without

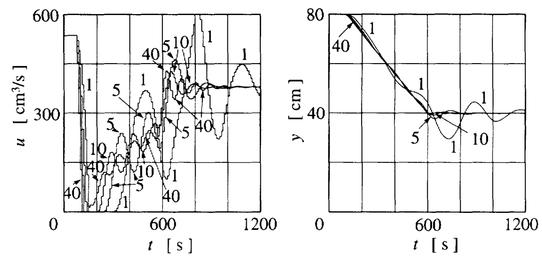

Fig. 2 Change of water level by following the specified trajectory

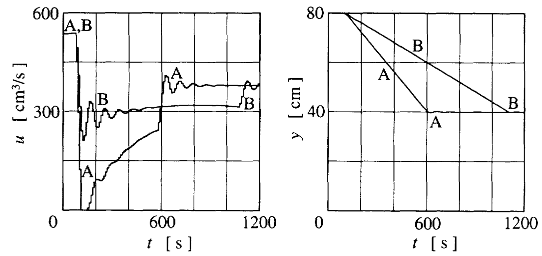

Fig. 3 Applicability of the same network to another trajectory

on-line learning. The result, shown as A in Fig. 3, is of course nearly identical to the 20th run in Fig. 2. Then the desired trajectory was changed to

$$
\begin{aligned}
&Y_{target} = 1 && \text{for } t < 100\ \mathrm{s}. \\
&Y_{target} = 1 - 0.0005\,(t - 100) && \text{for } 100\ \mathrm{s} < t < 1100\ \mathrm{s} \\
&Y_{target} = 0.5 && \text{for } t < 1100\ \mathrm{s}.
\end{aligned} \tag{6}
$$

The result obtained by using the saved network without further learning is shown as B in Fig. 3. It is found that the saved network can cover the control for a new trajectory which has not been experienced.

3. Discussion and Conclusions

The proposed method is very straightforward. First the future value of the target controlled value is used to obtain the increment of the manipulated variable. This term is used quite effectively for local learning. Second, the global learning supports control when the process proceeds along the new trajectory. The advantage of the present method is 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. Then excellent control is achieved. The value of $u_{max,\,j}$ is given by Eq. (5) rather arbitrarily. This problem will be discussed in another paper.

Literature Cited

1) Ishida, M.: *J. Chem. Eng. Japan*, **25**, 108-111 (1992)
2) Ishida, M. and J. Zhan: *J. Chem. Eng. Japan*, **25**, 485-489 (1992)