Application of a Neural Network to Evaluation of Interactions in a MIMO Process

Takehiro Ohba and Masaru Ishida
Research Laboratory of Resources Utilization, Tokyo Institute of Technology, Yokohama 226, Japan

A neural-net controller for multivariable systems is presented. The neural network for this controller has a structure in which small neural-net controllers for SISO systems are assembled and offers a unique path from the controlled variable to the manipulated variable. By using such a structured assembly, the interactions among the controlled and manipulated variables can be evaluated. The simulation results for both a linear three-input and three-output system and a crystal-growth process indicate that the proposed controller has the ability to learn the interactions between the control variables and to disclose the features of the interactions.

Introduction

Chemical processes generally have multiple inputs and multiple outputs, and there are interactions among them. This makes control of chemical processes difficult. When a mathematical model can describe the object process, we can predict such interactions. For a complex process, however, only a rough outline of relations among input and output data is available. For this reason, self-learning and parallel processing of the neural network are applied in previous works (Chen and Billings, 1992; Morris et al., 1994; Narenda and Mukhopadhyay, 1994). Since in the neural network the learned experience is stored by the values of the weights for the connections, we cannot extract the relationships between a specified input variable and a specified output variable. Neural-network architecture was presented by Mavrovouniotis and Chang (1992) for this problem. Their hierarchical network consists of a number of loosely coupled subnets, arranged in layers. In their work, the internal representations captured by the trained network were extracted by comparing the values of weights and biases.

In this article, we propose an inverse-model controller for a MIMO process obtained by effectively coordinating a certain number of small SISO neural-net controllers. This small controller uses a policy- and experience-driven neural network (PENN) (Ishida, 1992; Ishida and Ohba, 1994), and its training method is based on the indirect learning method proposed by Psaltis et al. (1988).

Since the structure of the proposed network allows extracting of the relations of each pair of the controlled and manipulated variables, the gains among the interactions can be estimated by solving a simultaneous equation using all of the weights of the network. We apply this controller to a crystal growth process demonstrated in an earlier article (Ishida and Zhan, 1995).

Control of a Multivariable System by NN

Control example of a multivariable system: a crystal growth process

An anthracene crystal growth process that was examined by Ito and Katoh (1994) was chosen as the example multivariable process with interactions. As shown in Figure 1, five electric heaters controlled the temperature distribution along the ampoule tube. The temperature of the anthracene was measured by five thermometers located at the center of the tube at the same level as the heaters. The ampoule tube is 7 mm in diameter and 36 mm in length. The anthracene in it was heated above its melting temperature. Then, to grow crystal, it was cooled gradually from the bottom. The desired temperature profile at these five locations at some instantaneous time is shown on the right side in Figure 1. The objective was to move this profile with the specified slope G at a constant speed R from the bottom ($z = 0$) to the top ($z = 1$).

The computer simulation is performed by dividing the ampoule tube into 50 cells in the vertical direction, so that the heat balance for each cell gives

$$\frac{\partial x(z,t)}{\partial t} = \frac{\partial^2 x(z,t)}{\partial z^2} + u(z,t) - v(z,t) \quad (0 \le z \le 1), \quad (1)$$

with

Correspondence concerning this article should be addressed to M. Ishida.

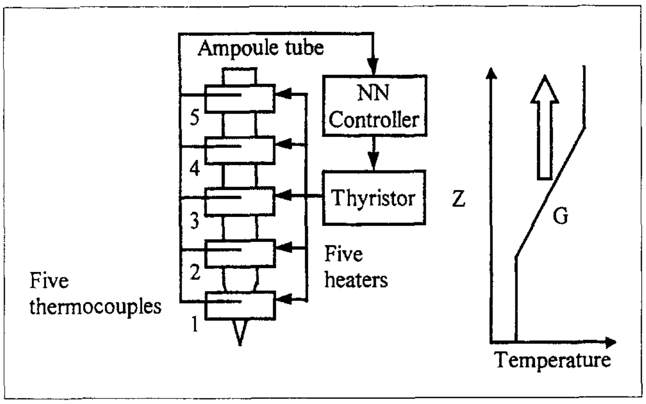

Figure 1. System and desired temperature profile.

$$x(z,0) = x_0(z) \quad \text{and} \quad \left.\frac{\partial x}{\partial z}\right|_{z=0} = \left.\frac{\partial x}{\partial z}\right|_{z=1} = 0, \qquad (2)$$

where x, z, t, u, and v, respectively, are temperature, height, time, dimensionless heat supplied by each heater, and heat loss. For the simulation, Eq. 2 is integrated by setting the temporal step size as 6 s and the spatial step size as 36/50 = 0.72 mm. Initially x is 491.15 K and u is 4.79 J/m^2·s. Ito and Katoh (1994) confirmed that the results of this simulation agree well with those observed experimentally.

This process has interactions among the temperatures at the observed five points, since the heat released by a heater can raise not only the temperature at the point where the corresponding heater has been installed but also the temperatures of the surrounding area.

For controlling this process, 30 s is chosen as the interval for measuring the temperatures and for setting the new manipulated variables. We assume the lag time of this process is less than 30 s.

Network structure of structured PENN controller

Figure 2 shows structure of the control system that is used in the proposed NN controller. The input parameters of the controller are the changes in the controlled variables, and the output parameters are the changes in the manipulated variables.

The structure of the neural network is shown in Figure 3. This network is a feedforward network with three layers. The proposed network has a characteristic structure, in which the small SISO networks shown in Figure 4 are combined with each other. The input parameters, ΔY_m and ΔY_p, in Figure 4 represent the temperature changes in the example process; they are explained in detail later. The output parameter, ΔU, is the change in the rate of heat supply. The connections between the input units and the hidden units in Figure 3 are composed of such SISO networks. In the example process, these are five manipulated variables ($M = 5$) and five controlled variables ($N = 5$). Hence, 25 ($= M \times N$) SISO networks are connected between the input and hidden layers: the input layer has 10 ($= M \times 2$) units and the hidden layer has 25 ($= M \times N$) units. Then all hidden units labeled the same as $\Delta U_{q,r,h}$ are linked to the unit ΔU_q in the output layer. Consequently, there is only one unique path from a specific input pair, $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$, and to the corresponding output unit ΔU_q.

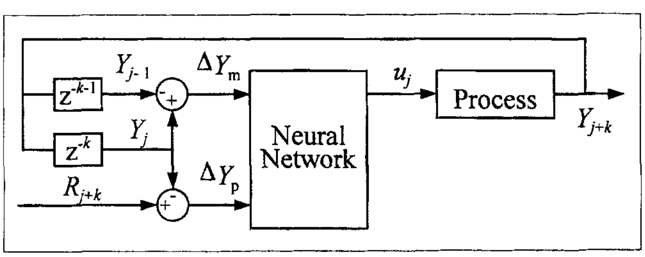

Figure 2. Control system.

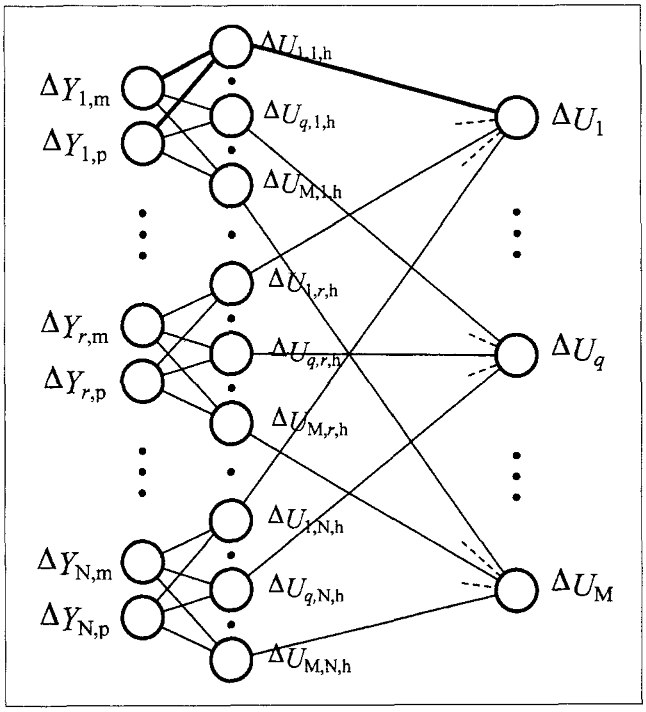

Figure 3. Structure of the proposed neural network.

Learning method

We have called the SISO network in Figure 4 a PENN controller, because we can apply two very effective kinds of learning method: learning of a local experience from the response of the process and learning of global policies that suggest the overall control features of the process. We can apply

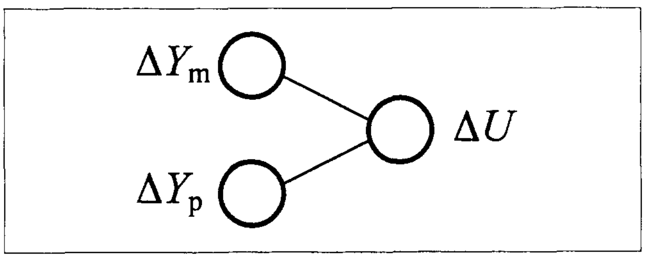

Figure 4. SISO neural-net controller.

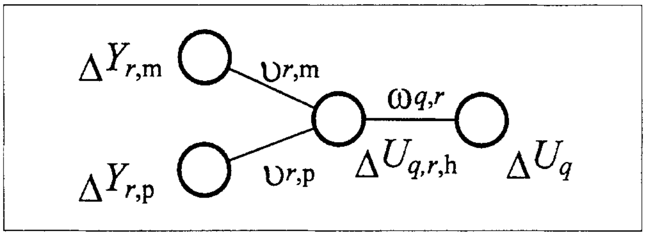

Figure 5. Unique path from y_r to u_q in MINO network.

these two kinds of learning to the proposed MIMO network as follows.

Figure 5 shows the part of the MIMO network (see Figure 3) that learns the relation between the manipulated variable u_q and the controlled variable y_r. The input parameters, $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$, are the changes in y_r, as defined by Eq. 3: $Y_{r,j}$ is the controlled variable at the instantaneous time j, and $\Delta Y_{r,m}$ is the difference between the present value $Y_{r,j}$ and the value $Y_{r,j-1}$ at the time of the sampling period preceding the present one; $\Delta Y_{r,p}$ is the difference between the value at the time k sampling periods after the present $Y_{r,j+k}$ and the value at the present time $Y_{r,j}$. The value of k is chosen by considering the delay property of the process. In the example process, we set $k=1$. The subscript m means "minus," the past from the present time j, and p means "plus," the future from j.

$$Y_{r,j} = (y_{r,j} - y_{r,\min})/(y_{r,\max} - y_{r,\min})$$
$$\Delta Y_{r,m} = Y_{r,j} - Y_{r,j-1}$$
$$\Delta Y_{r,p} = Y_{r,j+k} - Y_{r,j}. \tag{3}$$

The output parameter ΔU_q is the change in the manipulated variable u_q, and it denotes the difference between $u_{q,j}$, at the present time j, and $u_{q,j-1}$, at the time of the previous sampling period. The value of $\Delta u_{q,\max}$ is defined as the maximum change in the manipulated variable during the whole operation:

$$\Delta U_q = (u_{q,j} - u_{q,j-1})/\Delta u_{q,\max}. \tag{4}$$

Let us suppose we have operated this process by manually changing the manipulated variables and have recorded the temperature changes at these five locations. These changes give the relations among $\Delta Y_{r,m}$, $\Delta Y_{r,p}$, and ΔU_q. We can use these data for off-line learning, because the values of the temperature both at time j and at the future time $j+k$ are known. The learning of local experience is performed on the overall network in the structured PENN. Hence, a learning data set for the crystal growth process consists of 10 input data (five temperatures × 2) and 5 teaching data (five heaters). In this manner, the weights $\omega_{q,r}$, $\nu_{r,m}$ and $\nu_{r,p}$ are updated based on the back-propagation algorithm with the delta rule.

On-line local experience learning is summarized in Figure 6. One of the input datum, $\Delta Y_{r,p}$, is the difference between the present temperature, $Y_{r,j}$, and the value $Y_{r,j-k}$ at a time k sampling periods before the present. The other input datum, $\Delta Y_{r,m}$, is the difference between $Y_{r,j-k}$ and $Y_{r,j-k-1}$. The output datum ΔU_q is obtained as the difference between $u_{q,j-k}$ and $u_{q,j-k-1}$. In this way, the local experience acquired is used promptly during the control period.

In the SISO process, the global policies that indicate the control strategies or characteristics of the process can be determined easily. The following data are the policies for the process control and are applied when the relation between the manipulated and controlled variables is positive:

1: IF $\Delta Y_{r,m}=0$ AND $\Delta Y_{r,p}=0$ THEN $\Delta U_q=0$
2: IF $\Delta Y_{r,m}=-1$ AND $\Delta Y_{r,p}=0$ THEN $\Delta U_q=+1$
3: IF $\Delta Y_{r,m}=+1$ AND $\Delta Y_{r,p}=0$ THEN $\Delta U_q=-1$
4: IF $\Delta Y_{r,m}=0$ AND $\Delta Y_{r,p}=-1$ THEN $\Delta U_q=-1$
5: IF $\Delta Y_{r,m}=0$ AND $\Delta Y_{r,p}=+1$ THEN $\Delta U_q=+1$

These data are often used for fuzzy control systems. Rule 1 indicates the characteristics of the process at the steady state; Rule 2 corresponds to the case where the controlled variable Y_r has changed from the maximum value 1.0 at $j-1$ to the minimum value 0.0 at the present time j and is kept at 0.0 until $j+k$. Then the manipulated variable should be increased by the maximum rate, resulting in $\Delta U_q=+1$. These five rules roughly cover the whole control space of $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$, and each rule shows a special extreme condition. Rule 1 need not be taught when we assume that all thresholds are equal to zero in a layered network with a sigmoidal activation function.

For a multivariable system with complex interactions, we cannot generally predict policies among the manipulated variables and the controlled ones. However, we can find a manipulated variable that has the closest relation to each controlled variable. For each of these pairs, we apply the global policies just listed. In this application, since the supplied heat increases the temperature at the nearest position, Rules 2–4 are taught to the pairs for u_1 and y_1 through u_5 and y_5. By this learning, and by keeping $\omega_{q,r}$ unchanged, only $\nu_{r,m}$ and $\nu_{r,p}$ are updated for the selected pair. On the other hand, for the rest of the pairs that do not have close relations, the following rules are taught for the weights $\nu_{r,m}$ and $\nu_{r,p}$:

6: IF $\Delta Y_{r,m}=-1$ AND $\Delta Y_{r,p}=+1$ THEN $\Delta U_{q,r,h}=+1$
7: IF $\Delta Y_{r,m}=+1$ AND $\Delta Y_{r,p}=-1$ THEN $\Delta U_{q,r,h}=-1$

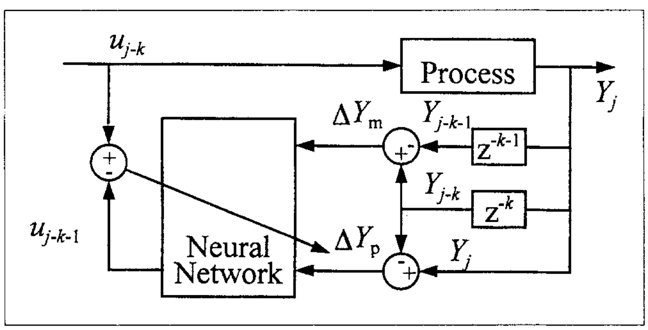

Figure 6. Local learning architecture.

By teaching these rules, the weights, $\nu_{r,m}$ and $\nu_{r,p}$, tend to learn the positive relation. Hence positive $\omega_{q,r}$ indicates the positive interaction, while negative $\omega_{q,r}$ indicates the negative interaction.

Control method

Suppose the weights of the connections have reached appropriate values by repeated learning. Control should be started by calculating $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$, and then obtaining $u_{q,j}$ at the present time j based on the neural-network output ΔU_q, by using the following equations:

$$\Delta Y_{r,p} = R_{r,j+k} - Y_{r,j}$$
$$\Delta Y_{r,m} = Y_{r,j} - Y_{r,j-1}$$
$$u_{q,j} = u_{q,j-1} + \Delta U_q \cdot \Delta u_{q,\max} \qquad (5)$$

where $\Delta Y_{r,p}$ for the input unit in the neural network now means the difference between the set point value at the time k sampling periods after the present, $R_{r,j+k}$, and the present value, $Y_{r,j}$.

Control procedure

In summary, the following procedures are adopted to control the process by the proposed structured PENN controller, which is composed of SISO-PENN controllers.

Step 1: Initialization of the Weights. The weights of the connections between the input layer and the hidden layer in the structured PENN controller are initialized randomly, and the weights between the hidden layer and the output layer are set to zero.

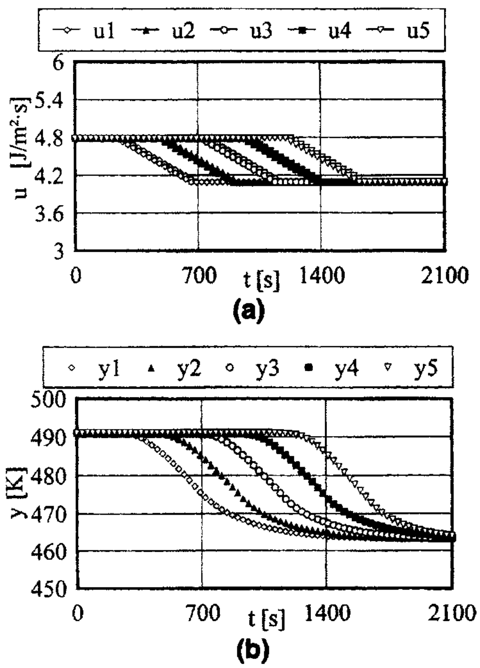

Figure 7. Previous control result.
(a) Heat supply; (b) obtained changes in temperature.

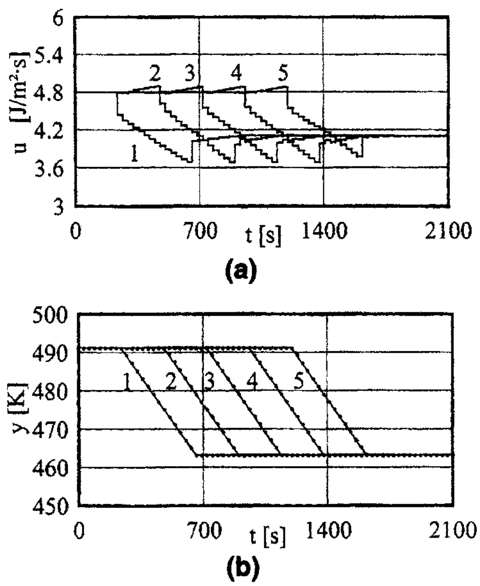

Figure 8. Control result.
(a) Heat supply; (b) temperature.

Step 2: Prelearning. Collect some process data, which may be the data in previous runs or may be obtained by manual operation. These data will be used for off-line learning. In the example crystal growth process, the data sets in Figure 7 are used. These data are obtained by manually changing the heat supplied by the heaters, from the initial value to the final value, as shown in Figure 7a, and observing the changes in the temperatures at the five locations. At time $j = 1$ ($t = 30$ s) to time $j = 68$ ($t = 2{,}070$ s), the values of the five temperatures and the new heat supplies are collected as local-experience data sets. The pre-local experience learning of these 68 data sets is repeated three times.

In this example, we have chosen prelearning data sets that are very close to the control test, by assuming that only limited operation is allowed. When we can change the heat supplies at the five points randomly, the acquired data contain information over wider control regimes, and prelearning can be accomplished more easily.

Step 3: Control of the Process with On-Line Learning. Start the control. The values of $u_{q,j}$ for the five heaters are obtained by the neural network. Simultaneously, the on-line learning of eight recent controlled results is accomplished.

During this sampling period (30 s), this local training is repeated three times and the global policies are taught only once. The repeated learning of eight data sets is required to get a satisfactory result on the second control run, as shown in Figure 8. When local learning is repeated only twice, we need four runs to get a result with agreement similar to the result shown in Figure 8. When we observe some noises in the measurement of the target temperatures at the five locations, repeated applications of global policies during the sampling period diminishes the effect of the noises.

Control result

The variations of the set points for the target temperatures R_r are set as shown by the solid lines in Figure 8b. The result of the second control run—that a satisfactory control result has been obtained—is shown in Figure 8. Let us look at Figure 8a. When the decrease in the manipulated variable u_1 was started, u_2 was increased to keep the temperature y_2 constant. This indicates that the structured PENN has acquired the cross-interactions in this crystal growth process by self-learning.

Comparison of this method with the PID method

The velocity-form equation for a PID controller is given as

$$\Delta U_q = K\Bigg[(E_{r,j} - E_{r,j-1}) + \frac{T_S}{T_I}E_{r,j} - \frac{T_D}{T_S} \times \{(E_{r,j} - E_{r,j-1}) - (E_{r,j-1} - E_{r,j-2})\}\Bigg]. \quad (6)$$

We can find close correspondences between the PENN and the traditional PID controller. When we consider constant-value control, the target value of the controlled variable R_r is constant:

$$R_{r,j} = R_{r,j-1} = R_{r,j+k}. \quad (7)$$

Then the inputs of the PENN are given by

$$\begin{aligned}\Delta Y_{r,m} &= Y_{r,j} - Y_{r,j-1} \\ &= (R_{r,j} - Y_{r,j}) - (R_{r,j-1} - Y_{r,j-1}) \\ &= E_{r,j} - E_{r,j-1}\end{aligned} \quad (8)$$

and

$$\Delta Y_{r,p} = E_{r,j}. \quad (9)$$

In this sense, the PENN controller is similar to the PI controller. In the calculation for a hidden unit, the derivative term—the last term in the righthand side of Eq. 6—is also taken into consideration. It should also be pointed out, however, that the output ΔU_q has the high–low limits, since this NN uses a sigmoidal activation function. Further, the weights in the NN are tuned by learning.

Evaluation of Interactions from NN

The hesitation in applying neural-net controllers to control is caused by unclear relationships between the controlled and manipulated variables. In this proposed PENN controller,

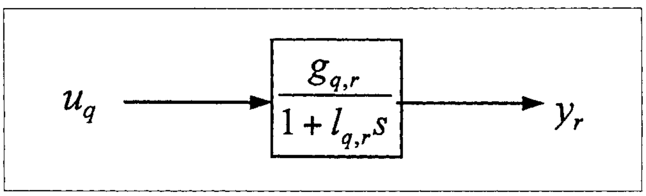

Figure 9. SISO process with first-order transfer equation.

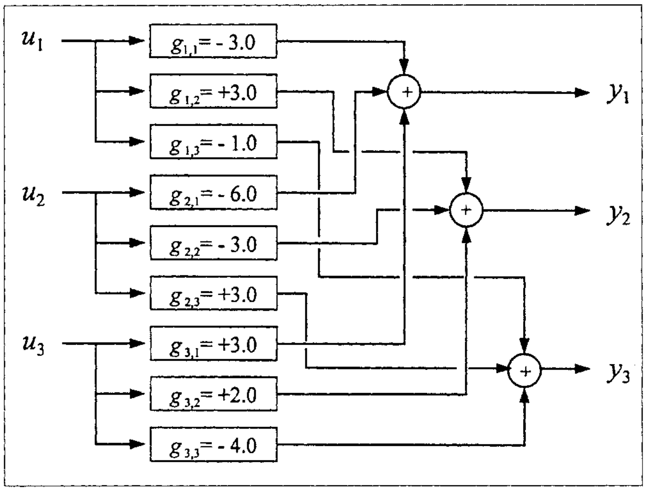

Figure 10. Example 3×3 process block diagram.

there is a unique path from a controlled variable to a manipulated variable. By using global learning, which updates the weights between the input and hidden units, the interaction term is represented more strongly by the weight between the hidden and the output units.

In this section, we try to find the correlation between each manipulated variable and controlled variable quantitatively. First, we discuss this problem by using a smaller network with three manipulated variables and three controlled variables. After that, the network is applied to the crystal growth process discussed earlier.

Control example: A 3-input and 3-output system

The first-order transfer function between u_q and y_r shown in Figure 9 is applied to a 3×3 system, where $l_{q,r}$ is the lag-time constant and $g_{q,r}$ is the gain constant. In this examination, we assume that all lag-time constants are equal to 0.2 and that the gain constants have various values, as shown in Figure 10.

Conditions of the control

The sampling interval is set at 0.5, and the parameters, $u_{q,\min}$, $u_{q,\max}$, $\Delta u_{q,\max}$, $y_{r,\min}$, and $y_{r,\max}$, are, respectively, 0.0, 100.0, 20.0, 0.0, and 100.0. The initial values of the three controlled variables, $y_{r,\text{initial}}$, are 50.0, and those of the three manipulated variables are 50.0. Since the lag-time constant in the first-order transfer function is small enough, we set $k = 1$ in Eq. 3.

The set points are chosen along the following equations

$$\begin{aligned}y_1 &= y_{1,\text{initial}} + 10.0 - 10.0 \times \sin\{36.0(t/50.0) + 0.5\pi\} \\ y_2 &= y_{2,\text{initial}} + 10.0 - 10.0 \times \sin[33.0\{(t-1.0)/50.0 + 0.5\pi\}] \\ y_3 &= y_{3,\text{initial}} + 10.0 - 10.0 \times \sin[30.0\{(t-3.0)/50.0 + 0.5\pi\}].\end{aligned} \quad (10)$$

The used prelearning data sets are shown in Figure 11. These data, which contain 98 learning data, have been obtained by randomly changing Δu_q.

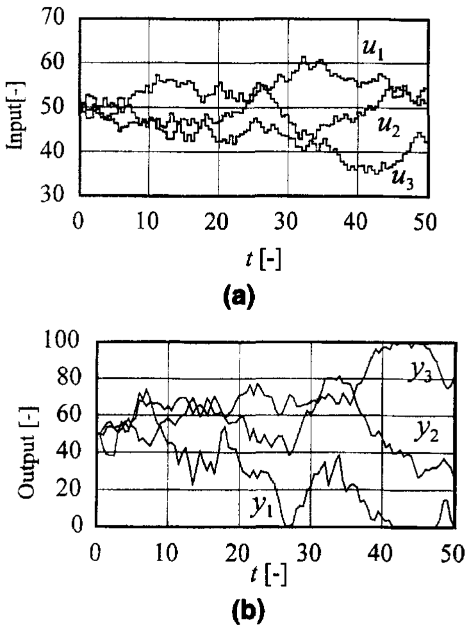

Figure 11. Observed data set.

(a) Arbitrarily chosen values of u_q; (b) results of y_r.

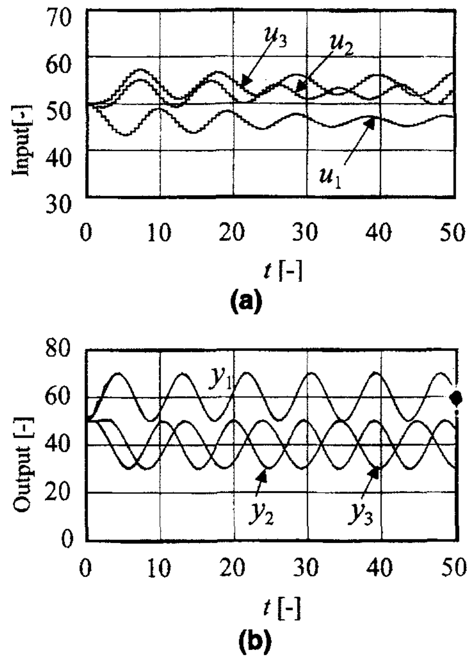

Figure 12. Control results of 3×3 process.

(a) Manipulated variable; (b) controlled variable.

The applied structured neural network has six input units, three for $\Delta Y_{r,m}$ and three for $\Delta Y_{r,p}$; nine hidden units; and three output units, ΔU_q. First, for control the weights of the connections are initialized, as mentioned before. Second, the previous control results in Figure 11 are taught three times. Third, control is started, and the on-line learning is applied while the control is running.

Control result

The control result with the structured PENN controller is shown in Figure 12. This control result for the first run shows that the controlled variables, y_1, y_2, and y_3, change satisfactorily along the set points.

Evaluation of the gains from NN

It requires two steps to extract the gains from the network. First, a representative parameter, $p_{r,q}$, is obtained for each pair of the controlled and manipulated variables. Next, the gain constants are calculated based on these representative parameters.

The output value of the hidden unit, $\Delta U_{q,r,h}$, is calculated by the sigmoidal activation function with the input parameters, $\Delta Y_{r,m}$ and $\Delta Y_{r,p}$, as follows, since the thresholds are not used in this network:

$$\Delta U_{q,r,h} = f(\Delta Y_{r,p}, \Delta Y_{r,m}) = \frac{1 - e^{-(\Delta Y_{r,p}\nu_{r,p} + \Delta Y_{r,m}\nu_{r,m})}}{1 + e^{-(\Delta Y_{r,p}\nu_{r,p} + \Delta Y_{r,m}\nu_{r,m})}}. \quad (11)$$

Similarly, the value of the output unit ΔU_q is given by

$$\Delta U_q = g(\Delta U_{q,r,h}) = \frac{1 - e^{-\Delta U_{q,r,h}\omega_{r,p}}}{1 + e^{-\Delta U_{q,r,h}\omega_{r,p}}}. \quad (12)$$

Consider the response of $\Delta Y_{r,p}$ to ΔU_q when some change takes place from the steady state, $\Delta Y_{r,m} = 0$; we call this term PENN parameter, $p_{r,q}$:

$$p_{r,q} = \left.\frac{\partial \Delta U_q}{\partial \Delta Y_{r,p}}\right|_{\Delta Y_{r,m}=0} = \frac{\nu_{r,p}\,\omega_{q,r}}{4}. \quad (13)$$

Hence, the PENN parameter, $p_{r,q}$, indicates a linearized relation between the controlled and manipulated variables:

$$\Delta U_q = \sum_r p_{r,q} \Delta Y_{r,p}. \quad (14)$$

In this example 3×3 process, ΔU_1, ΔU_2, and ΔU_3, respectively, become $p_{1,1}$, $p_{1,2}$, and $p_{1,3}$ when the changes of $\Delta Y_{1,p}$, $\Delta Y_{2,p}$, and $\Delta Y_{3,p}$ from the steady state are given by 1, 0, and 0; similarly, $p_{2,1}$, $p_{2,2}$, and $p_{2,3}$ for 0, 1, and 0, and $p_{3,1}$, $p_{3,2}$, and $p_{3,3}$ for 0, 0, and 1.

On the other hand, the gain constants, $g_{q,r}$ are given by

$$\Delta Y_{r,p} = \sum_q g_{q,r} \Delta U_q. \quad (15)$$

Hence, $g_{q,r}$ is evaluated as $g'_{q,r}$ by

$$\Delta Y_{1,p} = 1 = g'_{1,1}p_{1,1} + g'_{2,1}p_{1,2} + g'_{3,1}p_{1,3}$$
$$\Delta Y_{2,p} = 0 = g'_{1,2}p_{1,1} + g'_{2,2}p_{1,2} + g'_{3,2}p_{1,3}$$
$$\Delta Y_{3,p} = 0 = g'_{1,3}p_{1,1} + g'_{2,3}p_{1,2} + g'_{3,3}p_{1,3}$$
$$\Delta Y_{1,p} = 0 = g'_{1,1}p_{2,1} + g'_{1,2}p_{2,2} + g'_{1,3}p_{2,3}$$
$$\Delta Y_{2,p} = 1 = g'_{1,2}p_{2,1} + g'_{2,2}p_{2,2} + g'_{3,2}p_{2,3}$$
$$\Delta Y_{3,p} = 0 = g'_{1,3}p_{2,1} + g'_{2,3}p_{2,2} + g'_{3,3}p_{2,3}$$

$$\Delta Y_{1,p} = 0 = g'_{1,1} p_{3,1} + g'_{2,1} p_{3,2} + g'_{3,1} p_{3,3}$$

$$\Delta Y_{2,p} = 0 = g'_{1,2} p_{3,1} + g'_{2,2} p_{3,2} + g'_{3,2} p_{3,3}$$

$$\Delta Y_{3,p} = 1 = g'_{1,3} p_{3,1} + g'_{2,3} p_{3,2} + g'_{3,3} p_{3,3}. \quad (16)$$

In this manner, we can evaluate the gain matrix, G', by solving these equations using the conversions in Eqs. 3 and 4.

Result of evaluation

The gains in the example 3×3 process are evaluated as follows:

$$G' = \begin{bmatrix} -2.419 & -4.934 & +2.852 \\ +2.990 & -2.067 & +1.502 \\ -0.771 & +1.170 & -3.720 \end{bmatrix}.$$

The actual gain parameters in Figure 10 are

$$G = \begin{bmatrix} -3.000 & -6.000 & +3.000 \\ +3.000 & -3.000 & +2.000 \\ -1.000 & +1.000 & -4.000 \end{bmatrix}.$$

The differences between the evaluated parameters and the actual parameters are less than 18% on the basis of the maximum value 6.0.

Evaluation of Interactions in a Crystal Growth Process

The same procedure that was just discussed is applied to the crystal growth process, giving rise to the gains in Table 1. From this result we find that the electrical heater affects the temperature, especially at the same level. The gain of Δu_4 on Δy_5 is -0.113, and this is the largest interaction in the crystal growth process. The gains of Δu_3 on Δy_4, Δu_2 on Δy_3, and Δu_1 on Δy_2 are large; then comes the gain of Δu_2 on Δy_1.

Table 1. Gains Evaluated for a Crystal Growth Process

Δu \ Δy	1	2	3	4	5
1	4.862	−0.070	0.038	−0.025	0.006
2	0.061	4.871	−0.102	0.076	−0.045
3	0.013	0.042	4.848	−0.112	0.062
4	0.006	0.034	0.025	4.814	−0.113
5	−0.031	0.010	−0.074	−0.069	4.729

Conclusion

1. The structured PENN controller for a MIMO process was proposed. This controller consists of small SISO PENNs that represent the interactions in the MIMO processes.

2. The control of a crystal growth process is simulated, which shows that this structured PENN controller can control the multivariable system.

3. A method for evaluating the gain constants of the interactions among the control parameters from the weights of the connections in the network is proposed. This method is applied to the crystal growth process, revealing its interactions.

Acknowledgment

This work was supported by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, Sports and Culture of Japan.

Notation

E = error from the set point
O = normalized outputted data
T = normalized teaching data
U = normalized manipulated variable
ΔY_m = increment of the controlled variable in a previous time step
ΔY_p = increment of the controlled variable in the next k steps
h = index showing a hidden layer

Literature Cited

Chen, S., and S. A. Billings, "Neural Networks for Nonlinear Dynamic System Modeling and Identification," *Int. J. Control*, **56**, 319 (1992).

Ishida, M., "Control by a Policy- and Experience-Driven Neural Network," *J. Chem. Eng. Jpn.*, **25**, 108 (1992).

Ishida, M., and T. Ohba, "Control by a New Policy- and Experience-Driven Neural Network to Follow a Desired Trajectory," *J. Chem. Eng. Jpn.*, **27**, 137 (1994).

Ishida, M., and J. Zhan, "Neural Model-Predictive Control of Distributed Parameter Crystal Growth Process," *AIChE J.*, **41**, 2333 (1995).

Ito, Y., and N. Katoh, "A New Bridgeman Process for Single Crystal Growth Based on Control of Distributed Parameter System," *Kagaku Kougaku Ronbunshu*, **20**, 288 (1994).

Mavrovouniotis, M. L., and S. Chang, "Hierarchical Neural Networks," *Comput. Chem. Eng.*, **16**, 347 (1992).

Morris, A. J., G. A. Montaque, and M. J. Willis, "Artificial Neural Networks: Studies in Process Modeling and Control," *Trans. Inst. Chem. Eng.*, **72**, 1 (1994).

Narendra, K. S., and S. Mukhopadhyay, "Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks," *Neural Networks*, **7**, 737 (1994).

Psaltis, D., A. Sideris, and A. Yamamura, "A Multilayered Neural Network Controller," *IEEE Control Syst. Mag.*, **8**, 17 (1988).

Manuscript received Dec. 5, 1997, and revision received Apr. 24, 1998.