INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Application of back-propagation neural networks to the modeling and control of multiple-input, multiple-output processes

Takasu, Shinji, M.S.
Rice University, 1991
APPLICATION OF BACK-PROPAGATION NEURAL NETWORKS
TO THE MODELING AND CONTROL OF
MULTIPLE-INPUT, MULTIPLE-OUTPUT PROCESSES

by

SHINJI TAKASU

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE

Ka-Yiu San,
Associate Professor of Chemical Engineering, Chair

Sam H. Davis, Jr.,
Professor of Chemical Engineering

Kyriacos Zygourakis,
Associate Professor of Chemical Engineering

Houston, Texas
April, 1991
APPLICATION OF BACK-PROPAGATION NEURAL NETWORKS
TO THE MODELING AND CONTROL OF
MULTIPLE-INPUT, MULTIPLE-OUTPUT PROCESSES

SHINJI TAKASU

ABSTRACT

Certain properties of back-propagation neural networks have been found to be useful in structuring models for multiple-input, multiple-output (MIMO) processes. The network's simplicity and its ability to identify the non-linearity can have wide impacts on the construction of model-based control system. Care must be taken to train the network with consistent data that contains sufficient dynamic information.

A predictive control system based on the network model was proposed. Although the controller is relatively simple in terms of concept and computation, it shows excellent performances both in servo and regulator problems. Model prediction error sometimes causes a cyclic behavior in process responses; however, it can be stabilized by imposing certain constraints of controller action. The constraints are also effective for noisy measurements.

Use of neural networks for modeling and control of MIMO system appears to be very promising with its ability to treat non-linearity and process interactions.
ACKNOWLEDGEMENTS

The contributions of the following people and organizations are gratefully acknowledged:

Professor Ka-Yiu San, my thesis adviser, for his consistent guidance and friendship and unselfish approach to his students.

Professors Sam H. Davis and Kyriacos Zygourakis for their suggestions and criticisms and valuable time as members of my thesis committee.

the Department of Chemical Engineering at Rice University, professors and fellow students, who are the people that make Rice the excellent institution that it is.

Nippon Mining Co., Ltd. for giving me a precious opportunity to study at Rice University and its financial support.

Finally, my wife, Noriko for her thoughtfulness and continuous encouragement during my research work.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1. Control of Multiple-Input, Multiple-Output Processes</td>
<td>1</td>
</tr>
<tr>
<td>2. Model Predictive Control and BP Neural Network</td>
<td>3</td>
</tr>
<tr>
<td>II. Back Propagation Neural Networks</td>
<td>5</td>
</tr>
<tr>
<td>1. Back Ground on BP Neural Network</td>
<td>5</td>
</tr>
<tr>
<td>2. Neuron Model of BP Neural Network</td>
<td>7</td>
</tr>
<tr>
<td>3. Configuration of BP Neural Network</td>
<td>9</td>
</tr>
<tr>
<td>4. Data Processing of BP neural network</td>
<td>12</td>
</tr>
<tr>
<td>5. Learning and Recalling</td>
<td>14</td>
</tr>
<tr>
<td>6. BP Neural Network and Process Control</td>
<td>16</td>
</tr>
<tr>
<td>III. Modeling by BP Network</td>
<td>17</td>
</tr>
<tr>
<td>1. Model MIMO System - a Mixing Tank Problem</td>
<td>17</td>
</tr>
<tr>
<td>2. BP Network used for the System</td>
<td>20</td>
</tr>
<tr>
<td>3. Training of BP Network</td>
<td>24</td>
</tr>
<tr>
<td>a. Creation of Training Data Set</td>
<td>24</td>
</tr>
<tr>
<td>b. Training of BP Network</td>
<td>33</td>
</tr>
<tr>
<td>4. Simulation with the BP Network Model</td>
<td>34</td>
</tr>
<tr>
<td>a. Introduction</td>
<td>34</td>
</tr>
<tr>
<td>b. Results of One-Step-Ahead Simulation</td>
<td>36</td>
</tr>
<tr>
<td>c. Results of Recursive Simulation</td>
<td>53</td>
</tr>
<tr>
<td>5. Discussion</td>
<td>62</td>
</tr>
</tbody>
</table>
IV. Control by BP Network 65
   1. Control System Configuration 65
      a. Flow Rates Selection by Predictor 65
      b. Error Correction of the Predictor 72
      c. Control System Configuration 75
   2. Results of the Servo Problem 75
      a. Response to the Step Change in Set Points 75
         i) Effects of Parameters on the Controller 75
         ii) Comparison with Tuned PI Controllers 91
         iii) Use of Less-Trained Network 98
      b. Response to the Ramp Change in Set Points 103
   3. Results of the Regulator Problem 107
      a. Disturbance Considered 107
      b. Response to the Disturbance Change 108
   4. Noise Effect on the Controller Performance 120
   5. Discussion 127
   V. Conclusion 130
   References 132
LIST OF FIGURES

Fig. 1  2
Examples of Multivariable Control Problems

Fig. 2  8
A Schematic of a Neuron

Fig. 3  10
The Relationship between the Input and the Output of a Sigmoid Function

Fig. 4  11
Back Propagation Neural Network Schematic

Fig. 5  18
Model MIMO System - a Mixing Tank Problem

Fig. 6  23
BP Network Used for this Study

Figs. 7 & 8  30
Response of the System to the Pseudo-Random Binary Disturbances (1) & (2)

Fig. 9  32
Procedure of Training Data Set Construction

Fig. 10  35
Total Squared Error vs. No. of Training Epochs

Fig. 11  37
One-Step-Ahead Simulation Scheme using BP Neural Network
Figs. 12 ~ 15
Response of the System to the Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (1) ~ (4)
(Network Model: TDS-900 with TSE of 0.18)

Figs. 16 ~ 19
Response of the System to the Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (5) ~ (8)
(Network Model: TDS-2000 with TSE of 0.4)

Figs. 20 ~ 23
Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (1) ~ (4)

Fig. 24
Recursive Simulation Scheme using BP Neural Network

Figs. 25 & 26
Response of the System to the Step Change in Inlet Flow Rates: Recursive Simulation (1) & (2)
(Network Model: TDS-2000 with TSE of 0.196)

Figs. 27 ~ 30
Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: Recursive Simulation (1) ~ (4)

Fig. 31
Generalized Feedback Control System and BP Network Model

Fig. 32
Flow Rate Selection by Predictor

Fig. 33
Finer Flow Rate Selection Scheme (1)
Fig. 34
Finer Flow Rate Selection Scheme (2)

Fig. 35
Error Correction of Predicted Response

Fig. 36
Predictive Control System by BP Network Model

Figs. 37 ~ 40
Response to the Step Change in Set Points by NMP Controller (1) ~ (4): (hs'=0.6→0.8, Ts'=0.6→0.4)

Figs. 41 ~ 44
Response to the Step Change in Set Points by NMP Controller (5) ~ (8): (hs'=0.2→0.2, Ts'=0.3→0.7)

Fig. 45
Effect of ΔF' on IAE and ICA (M = 30) for Set Point Changes

Fig. 46
Effect of the Step Number of Recursive Simulation on IAE and ICA (M = 30) for Set Point Changes

Fig. 47
Tuning of PI Controller Parameters

Figs. 48 ~ 50
Comparison of Process Responses between Best-Tuned PI Controller and NMP Controller (1) ~ (3):
Step Change in Set Points
Figs. 51 & 52
Response to the Step Change in Set Points by Ill-Tuned PI Controllers (1) & (2)

Figs. 53 & 54
Response to the Step Change in Set Points by NMP Controller with less-Trained Network Model (1) & (2)

Figs. 55 ~ 57
Comparison of Process Responses between Tuned PI Controller and NMP Controller (1) ~ (3):
Ramp Change in Set Points

Figs. 58 & 59
Process Response to the Disturbance by NMP Controller (1) & (2): (hs'=0.6, Ts'=0.6)

Figs. 60 & 61
Process Response to the Disturbance by NMP Controller with Constraints of Controller Action (1) & (2):
(hs'=0.6, Ts'=0.6)

Fig. 62
Effect of Controller Action Parameter on IAE and ICA for the Disturbance of α = 1.0→0.5: (hs'=0.6, Ts'=0.6)

Figs. 63 & 64
Process Response to the Disturbance by NMP Controller with Constraints of Controller Action (1) & (2):
(hs'=0.2, Ts'=0.3)

Fig. 65
Effect of the Gaussian Noise (σ= 2 %) on the Steady-State Operation (Open-Loop)
Fig. 66
Response to the Step Change in Set Points by Best-Tuned PI Controller with Noise Free Condition

Fig. 67
Response to the Step Change in Set Points by Best-Tuned PI Controller with the Gaussian Noise (σ = 2 %)

Fig. 68
Response to the Step Change in Set Points by NMP Controller with the Gaussian Noise (σ = 2 %) with No Constraints of Controller Action

Fig. 69
Response to the Step Change of Set Points by NMP Controller with the Gaussian Noise (σ = 2 %) with Constraints of Controller Action
LIST OF TABLES

Table 1  8
Neuron Transfer Function

Table 2  27
Steady-States Points for the Calculation (1)

Table 3  27
Process Parameters for the Model Simulation

Table 4  28
Steady-States Points for the Calculation (2)

Table 5  94
Tuning Results of PI Controller Parameters
for the Step Change in Set Points

Table 6  119
Tuning Results of PI Controller Parameters
for the Disturbance Change

Table 7  119
Comparison of Process Responses to the Disturbance Change
between Best-Tuned PI Controller and NMP Controller

Table 8  129
Comparison of Process Responses to the Step Change in
Set Points between Best-Tuned PI Controller and NMP
Controller
I. Introduction

1. Control of Multiple-Input, Multiple-Output Processes

Most industrial process control applications involve a number of input and output variables. These applications are often referred to as multiple-input, multiple-output (MIMO) systems to distinguish from the simpler single-input, single-output (SISO) systems. It can be argued that for virtually any important process there are at least two variables that must be controlled: product quality and throughput [Shinskey, 1988].

Two examples of processes with two controlled variables and two manipulated variables are shown in Fig. 1. These examples illustrate a characteristic feature of MIMO control problems, the presence of process interactions. That is, each manipulated variable can affect both controlled variables. Consider the in-line blending system shown in Fig. 1(a). Two streams containing species A and B, respectively, are to be blended to produce a product stream with mass flow rate \( W \) and composition \( X \), the mass fraction of A. Adjusting either flow rate, \( W_a \) or \( W_b \), affects both \( W \) and \( X \). Similarly, for the distillation column in Fig. 1(b), adjusting either reflux flow rate \( R \) or steam flow rate \( S \) will affect both distillate composition \( X_D \) and bottom composition \( X_B \).

The process interactions can produce undesirable control loop interactions during closed-loop operation. The relative
(a) In-line Blending Systems

(b) Distillation Column

Fig. 1 Examples of Multivariable Control Problems
gain array (RGA) approach [Bristol, 1966] provides a convenient way of determining the degree of process interaction and the best pairing of controlled and manipulated variables for a multiloop control scheme. Also, an analytical approach based on singular value analysis [Smith, Moore, and Bruns, 1981; Moore, 1986] can be used to determine the multiloop control configuration. If the control loop interactions are unacceptable for multiloop control, then a number of alternatives are available. One approach is to reduce the interactions by detuning one or more of the control loops. Alternatively, it may be possible to reduce the interactions by a different choice of manipulated or controlled variables. If this is not possible, decoupling control techniques provide other options. Although these model-based control strategies can provide significant improvements over conventional multiloop control, their sensitivity to modeling errors has to be carefully considered.

2. Model Predictive Control and BP Neural Network

One of the most advantageous methods for MIMO control problems is the model predictive control technique, which is based on optimization of a quadratic objective function involving the error between the set points and the predicted outputs. Design methods in this category are based on a particular type of discrete-time model (a convolution model)
and include Dynamic Matrix Control (DMC) [Prett and Garcia, 1988; Cutler and Ramaker, 1980] and Model Algorithmic Control (MAC) [Richalet, Rault, Testud, and Papon, 1978; Mehra, Rouhani, and Eterno, 1982]. Both of these techniques have been used successfully in commercial control applications involving MIMO processes.

However, the process of structuring and verifying process models is labor intensive. Model development and verification often accounts for 75% of the expenditures in advanced control projects [Bhat, Minderman, and McAvoy, 1988]. Even if the manpower is available to construct the model, the process may change with time and so must its model. Updating this model requires a consistent commitment of manpower. Furthermore, this kind of work requires a high degree of expertise, which is probably beyond the scope of a typical chemical engineer, and may require a special consultant.

BP neural networks offer a way of addressing these problems and allow greater uses of model-based control techniques by its simplicity of model construction. This issue will be addressed in detail later.
II. Back Propagation Neural Networks

1. Background on BP Neural Network

The back propagation (BP) is the most widely used learning neural network to map a matrix of inputs to a matrix of outputs. It was modeled after the human brain and can work out the relationship between inputs and outputs. The name of back propagation originated from the manner in which adjustable parameters (weights) in the network are changed [Klimasauskas and Guiver, 1988]. Specifically, the error between the actual output and the network's output is propagated back through the network and used to alter the weights in a manner that improves the mapping.

The attractive properties of BP neural networks are the ability to learn by examples and to structure non-linear mappings. If a BP network is shown a consistent set of examples of input data and corresponding output data, then the net can form a mapping for this set. When the net has learned (or has been trained) sufficiently, it can predict the outputs for a given input data, although its interpolation and extrapolation capabilities are limited. For modeling purposes, data from the process can be used to form a set of examples that is needed for the training. As more examples of different dynamic situations are included in the network training set, the mapping more closely approaches the actual dynamic system.
Another attractive property of BP network is the ability to capture the non-linear nature of a process. Suppose that a function can be normalized in the absolute range of zero to one. Then, there exists a three-layered BP network that can approximate the function to within an arbitrarily small mean-squared error. This is a paraphrase of a mathematical theorem [Hecht-Nielsen, 1988]. The theorem does not restrict the class of functions, and both linear and non-linear functions can be mapped in this manner. Since most real processes are non-linear, this non-linear mapping ability introduces an important new prospect for process model construction.

Applications to date of BP network include speech synthesis and recognition, visual pattern recognition, analysis of sonar signals, analysis of natural gas market, bond rating, defense applications, medical diagnosis, and learning in control systems [Bhat and McAvoy, 1989]. In the field of chemical process control, BP network is successfully used for sensor interpretation [McAvoy, Wang, Naidu, and Bhat, 1989], dynamic modeling [Bhat and McAvoy, 1989], and the system design of distillation column control [Birky and McAvoy, 1988]. Also, some preliminary examination of the mapping quality of BP networks as applied to optimal predictive control has been discussed [Donat, Bhat, and McAvoy, 1990]. Additional examination of the prospect to
replace the system model in Internal Model Control has been examined [Hernandez and Arkun, 1990].

2. Neuron Model of BP Neural Network

A schematic of a processing element (neuron) in a BP neural network is shown in Fig. 2. As can be seen, signals received by the neuron are summed and processed to yield an output signal. Those signals between neurons are communicated through connections, which also designate the order of signals. The output from a neuron is sometimes called its activation. The activation signals from a sending neuron travel down the connection to a receiving neuron. Before the activation is received, it is multiplied by the weight associated with the connection. A weight can take on any value and it is adjusted during the learning process. The product of the output and the weight is the signal transmitted on the connection to a receiving neuron. The sum of the inputs into a neuron is called the net input. If the net input is above an upper threshold value, then the neuron fully fires and emits a signal whose magnitude is one. If it is below a lower threshold, then the neuron doesn't fire and emits a signal of zero. If it is somewhere between thresholds, then the neuron emits an intermediate fractional output. A neuron's output signal is duplicated and disseminated to each connection at its output.
Fig. 2 A Schematic of a Neuron

Table 1 Neuron Transfer Function

1 Unity Transfer Function

Neuron Output = \( 1 \times \sum \text{Neuron Input} \)

2 Sigmoid Transfer Function

Neuron Output = \( \frac{1}{1 + \exp(-\sum \text{Neuron Input})} \)
The strength of the signal emitted by a neuron is determined by its activation function, sometimes called the transfer function. In this study, the transfer function is either a unity function or a sigmoid function. Equations that characterize these functions are listed in Table 1. If the transfer function is unity, then the net input is passed unchanged to the output. If the transfer function is sigmoidal, then the output to the net input behaves in an on/off manner as shown in Fig. 3(a). As can be seen, a net input greater than an upper threshold of +3 results in an output of approximately one. Similarly, an input below a lower threshold of -3 results in an output nearly zero.

With a linear combination of sigmoid functions, a "bump" can be created as shown in Figs. 3(b) - 3(d). A "bump" of any size, any location, and any slope of its sides can be created in this manner. By adding more "bumps" together, arbitrarily complex functional relationship between one or more inputs and a single output can be created.\textsuperscript{1}

3. Configuration of BP Neural Network

A typical BP network configuration, which is used in this work, is shown in Fig. 4. The boxes and circles are

\textsuperscript{1} The sigmoidal function may not always be the best function. At times, it may be better to use sine or other type of functions. Using the sinusoidal function is equivalent to Fourier decomposition of the inputs [Klimasauskas, 1989].
Fig. 3: The Relationship between the Input and Output of a Sigmoid Function: (a) A curve of sigmoidal function. (b) The curve is shifted by adding a negative bias. (c) The curve is shifted by adding a positive bias. (d) The resulting function of \( Y_3 = Y_2 - Y_1 \) looks like a "bump".
Fig. 4 Back Propagation Neural Network Schematic
neurons and the lines between the neurons are connections. The network shown in Fig. 4 has three layers, namely the input, hidden, and output layers.\(^2\) Every input layer neuron is connected to every hidden layer neuron. The two layers are said to be fully connected. Likewise, the hidden layer and output layer are fully connected. The input and output layers are not connected at all.\(^3\) A bias neuron is connected to every hidden and every output neuron, but is not connected to the input neurons. Each input neuron and bias neuron has a unity transfer function. They simply pass through the net input and disseminate copies to the connections at their output. Neurons in the hidden and output layer have a sigmoidal function and thus exhibit on/off behavior.

These components of BP network can be created in computer hardware as electronic or optical "hard-coded" entities [Eliot, 1987]. In order to use conventional computer hardware available for this study, they are instead simulated in software [Klimasauskas, 1989].

4. Data Processing of BP Neural Network

The manner in which BP neural network processes inputs

\(^2\) Although a single hidden layer is shown, there can be more. Extra layers add a supplemental degree of freedom to the mapping. This may be advantageous for applications such as image processing. However, they are not needed for the mappings encountered in process identification.

\(^3\) The input layer and output layer can be connected. A three-layered structure with the Direct Linear Feedthrough (DLF) weights are used for system identification [Haesloop and Holt, 1990].
to yield outputs is as follows. When the input layer receives a vector data that is normalized between zero and one, each neuron passes the corresponding vector component unchanged to the connection at the output of the neuron. These connections lead to the hidden layer. Before the signals from the input layer are received by the hidden layer, they are multiplied by the respective weights associated with the connections between two layers. Each hidden layer neuron also receives input from the bias neuron whose output is always one. The combined input from the input neurons and the bias neuron into the hidden layer neuron is its net input. This net input is used in the sigmoidal transfer function to determine the neuron's output. The resulting output is duplicated and disseminated to the connections between this neuron and output layer neurons. Before the signal is received by output layer neurons, it is multiplied by the weights between the two layers.

Similarly, each output layer neuron also receives input from the bias neuron. Finally, the combined input from the hidden neurons and the bias neuron is transferred to the neuron's output, which is a component of the BP network output vector.

As mentioned before, the bias neuron has connections to every hidden layer neuron and every output neuron. The input to the bias neuron can be assumed to be one and its transfer function is assumed to be unity; thus its output is always
one. This output is duplicated and disseminated to each connection as well as other neurons. Each output signal is multiplied by the weight on the connection and this net output from the bias neuron becomes an input to its associated neuron.

The presence of a bias neuron is equivalent to the associated neuron having an adjustable threshold. This equivalency can be demonstrated by an example. Suppose that the net input to a neuron with a sigmoid transfer function from the previous layer sums to -3 and suppose that there is no input from the bias neuron. The net input becomes -3 and will not cause the neuron to fire. If the input from the bias neuron to this neuron is made to be 6, then the net input becomes 3 and the neuron fires. This situation can be thought as a change of the threshold. Namely the threshold for firing the neuron is changed from +3 to -3 by the bias input. Since the weight on the connection from the bias is fully adjustable, the bias input provides the same effect as an adjustable threshold.

5. Learning and Recalling

BP neural networks are operated in two distinct modes: learning and recalling. The learning mode is a relatively lengthy process while the recalling mode is practically instantaneous. The BP net learns by sequentially examining a series of inputs and corresponding outputs called a training
set. This set is usually constructed from the experimental data, although it is constructed from a physical model of the system to be controlled for this study. When an input vector is presented to the BP net, the net produces its output and that output is compared to the actual output. The difference between the net output and the actual output is used to adjust the weights such that the error is reduced in the generalized delta rule (GDR) learning method.

This method is an implementation of a gradient descent to the weights that minimize the sum of squared differences between the network and actual output values. A detailed explanation of this method for adjusting the weights is given in the book of "Parallel Distributed Processing" [Rumelhart and McClelland, 1987]. The network slowly but eventually steps toward the desired mapping. However, its extent is limited by the properties of gradient descent and the amount of computation time invested.

The recalling mode is quite simple. The weights are fixed during this mode. A vector or matrix of normalized inputs is presented to the input layer. The signal is processed in succession by the hidden and output layers to yield an output in the manner previously discussed. The output is also a matrix or a vector of normalized values, which is calculated instantaneously since the calculations involves only simple summation and transfer function computations.
6. BP Neural Network and Process Control

As mentioned above, a BP network is able to model a chemical process by examining data from that process. Although this network modeling requires some labor, it does not require much effort and special expertise. However, there is an unconventional nature of the neural net mapping. In conventional model-based predictive control, the process model is stated in a matrix form. Standard optimization techniques can be applied to compute the profiles of manipulated variables that minimize the error between the outputs and their desired values. Neural net mappings are non-linear transformations that cannot be restated into a matrix form. Therefore, they are not yet invertible. For this reason, network mappings do not fit easily into the framework of current advanced control research. The non-invertability of the network means that the exact solution using the network model can not be obtained in a direct manner. Instead, a solution can be chosen from the results of a number of case studies, which can be directly performed based on the network model. The control system proposed in the current study uses the case study method, which is discussed in Chapter IV.
III. Modeling by BP Network

1. Model MIMO System - a Mixing Tank Problem

A simple chemical process is needed to explore the particulars of BP neural network for the modeling application of MIMO processes. In this study, the following mixing tank control problem with a constant crosssectional area, A (Fig. 5) is used as a model system. The liquid level, h, and the temperature, T, in the stirred tank are the controlled variables. The inlet flow rates of the hot and cold stream, \( F_h \) and \( F_c \), are the manipulated variables. The mixed fluid flows out of the tank at a rate proportional to the square root of the liquid level, \( h \). It is assumed that the temperature of the inlet streams, \( T_h \) and \( T_c \), are maintained constant and that the liquid physical properties, such as density and specific heat, are constant. Then, energy and material balances for the system lead to equations (1) and (2). In the equations, \( t \) denotes time and \( \alpha \) denotes a proportional constant determined by the characteristic of the outlet valve.

\[
A \frac{dh}{dt} = F_h + F_c - \alpha \sqrt{h} \quad (1)
\]

\[
Ah \frac{dT}{dt} = F_h T_h + F_c T_c - (F_h + F_c)T \quad (2)
\]
Fig. 5 Model MIMO System - a Mixing Tank Problem
As the inlet flows vary, the liquid level will change until the appropriate hydrodynamic head is established to match the outlet flow with the sum of inlet flows. Also the temperature will change toward the value that is determined by the ratio of the hot stream flow rate to that of the cold stream. In other words, this mixing tank is treated as a disturbance-free two-input, two-output process.

The process has two difficult features for process control: process interactions and non-linearity. Suppose that the liquid level and the temperature in the stirred tank (controlled variables) are controlled by two proportional-integral (PI) controllers with hot and cold stream flow rates (manipulated variables), respectively. Any adjustments of the cold flow rate to increase or decrease the temperature will affect the liquid level. Similarly, adjustments of the hot flow rate to increase or decrease the liquid level also affect the temperature. Thus, undesirable control loop interactions will be produced frequently.

The second feature is a non-linearity. Because of the presence of non-linear term \( \sqrt{h} \) in equation (1) and \( h \) (left-hand side) in equation (2), the response of liquid level or temperature to the change in inlet flow rates varies at each liquid level. As a result, controller parameters such as proportional gain and integral time tuned at one liquid level will not be appropriate for the other. Thus, although the mixing tank though is a simple process, it has enough
interesting features to be used for the examination of potential BP network applications.

2. BP Network used for the System

The BP network is a discrete-time model and consists of contiguous time histories of all variables. As a first step in constructing a BP network model, a set of variables that completely characterizes the system is selected. For the mixing tank problem, one such set includes the liquid level, the temperature in the tank, and the two inlet flow rates. All the variables in differential equations (1), (2) are included in this set.\(^4\)

A mapping arrangement must also be selected. That is, the determination of the structure of input-output pair. A particularly useful arrangement employs current and past data to predict the behavior of the controlled variables in the next time step. In this context, histories of the liquid level, the temperature in the tank, and two inlet flow rates, which are the past and present values of those variables, are chosen to form the input vector. The output variables of the mapping contain two items: the liquid level and the temperature in the tank at the next time step.

\(^4\) In a case that temperature of inlet flows, \(T_h\) and \(T_c\), are not constant and that their changes are measured, these temperatures should be included in the variable set of BP network. However, this 4-input, 2-output system is not treated in this study.
Another decision necessary to structure a BP network is the length of the time history of the variables in the mapping. This can be determined from disturbance response data or a knowledge of process delay. As realized from the equations (1), (2), no process delay is considered in this study. Therefore, the minimum history length will be one (sampling period) for the open-loop operation of this system. The actual length chosen in this study is three for all the variables, which is slightly larger than is necessary, but not too large. If the choice of the history length is too small, the mapping will ultimately fail. While, if the choice is too large, many elements of the input vector ultimately have no effect on the output. Much of the process data will be used to negate the effect of these extra input elements. Subsequently, more additional dynamic examples are required, leading to a larger training set and a larger network structure. The extra requirement will ultimately pose an additional computation burden and slow the training process down. It must be noted that the same history length of three is used for all variables in this study; however, different lengths can be used for different variables.5

5 Statistical techniques exist to identify elements in the window that are most correlated [Box and Jenkins, 1976; Jenkins and Watt, 1968]. These techniques can be used to gauge the size of the window for a system in which the process order is not easily identified. They can also be used to select the most effective non-contiguous elements in the history.
The final decision in structuring a BP network is the number of hidden neurons. The choice of the number of hidden neurons is important for the success of the mapping. These neurons represent the actual memory capacity of the network [McClelland and Rumelhart, 1988]. If the number is too small, the mapping error will not converge to the required degree during the learning process. The resulting model does not have enough accuracy and its usefulness is doubtful. On the other hand, addition of hidden neurons has the effect of increasing training times; therefore, it is desirable not to be excessive about their number. No exact method exists for determining the required number; however, some general guidelines result from experiences. As a rule of thumb for system identification, the number of hidden neurons that at least equals the number of the input neurons seems to give a successful mapping [Broussard, 1990]. As a more conservative choice, two and a half times of the number of input neurons were used for the number of hidden neurons in this study. In most cases, these recommendations are much larger than the number necessary to represent the adequate mapping; however, they are on the safe side, and may not retard learning substantially.

The BP network shown in Fig. 6 is the structure adopted in this study. There are 12 input neurons, 30 hidden neurons, two output neurons, one bias neuron, and 452 (= 12×30 + 30×2 + 30+2) connections in the network. The 452 parameters are
Fig. 6 BP Network used for this study
adjusted to form a desired mapping during the learning process. It must be noted that variables with a prime refer to normalized data. The normalization process is discussed in the following section.

3. Training of BP Network

a. Creation of Training Data Set

As mentioned earlier, all the input and output data to the BP net have to be scaled into the range 0 to 1 for convergence reasons [Hecht-Nielsen, 1988]. In this study, the following normalization is used for each variable.

\[
F'_h = \frac{F_h}{F_o} , \quad F'_c = \frac{F_c}{F_o} , \quad h' = \frac{h}{h_o} ,
\]

\[t' = \frac{t}{t_o} , \quad T' = \frac{(T - T_c)}{(T_h - T_c)} = \frac{(T - T_c)}{\Delta T}\]

where, \(F_o\): maximum capacity of the pumps

\(h_o\): maximum liquid level

\(t_o\): sampling period

\(\Delta T\): temperature difference between hot and cold stream

Using these normalization parameters, the components of input and output vectors are always scaled between 0 and 1.\(^6\)

\(^{6}\) Depending on the neuron transfer function, networks usually have the normalization range of -0.5 to 0.5 [SAIC Inc., 1988] or zero to one. The latter is the range used in this study.
The key in normalization is to use as much of the normalization range as possible to achieve good resolution, but never to exceed the range. A variable out of the range causes instability in the BP learning algorithm and results in a mapping failure.

As a standard procedure, this data normalization is performed after the collection of experimental data. In this study, a simulated mixing tank is used and all the base data are obtained through numerical calculations of the system governing equations. Therefore, instead of normalizing each datum, the following normalized differential equations obtained from equations (1), (2) and (3) were used to generate the normalized data directly.

\[
\frac{dh'}{dt'} = \left( \frac{t_o F_o}{h_o A} \right) \left( F'_h + F'_c - \left( \frac{\alpha \sqrt{h_o}}{F_o} \right) \sqrt{h'} \right)
\]

(4)

\[
\frac{dT'}{dt'} = \left( \frac{t_o F_o}{h_o A \Delta T} \right) \left( \frac{1}{h'} \right) \left\{ F'_h T'_h + F'_c T'_c - \left( F'_h + F'_c \right) \left( \Delta T \cdot T'_h + T'_c \right) \right\}
\]

(5)

The data collection from the process can be done by either closed-loop tests or open-loop tests. It is ideal to collect the data from day-to-day closed-loop operations. However, closed-loop dynamics usually contain less information because controllers tend to mute the relationship
between input disturbances and output responses. Therefore, open-loop tests, which allow quick access to the information needed in process identification, are chosen in this study. It must be noted that the useful information can be extracted from a large number of closed-loop operation data with a data screening process. This closed-loop identification system has been successfully used for the tuning of PI level control system [Broussard, 1990].

The open-loop test involves a pseudo-random binary disturbance that is formulated to perturb the process but keep it stable and near the average set point. This signal must excite the system in a proper manner to produce data that is rich in the information needed in model construction. In this study, pseudo-random binary disturbances of hot and cold stream flow rates (manipulated variables) are used around steady-state liquid levels and temperatures (controlled variables). The number of steady points used in this study was originally nine. However, the model based on this data set alone does not predict liquid levels and temperatures precisely over the entire range. As a result, the number of steady-state points was increased to 21. These steady-state points and the process parameters used in the simulation are listed in Tables 2 ~ 4.

A Runge-Kutta 4th-order method was used for the numerical integration of equations (4) and (5). Each calculation was carried out up to 103 or 53 of sampling times
Table 2  Steady-States Points for the Calculation

<table>
<thead>
<tr>
<th>$h'$</th>
<th>$T'$</th>
<th>$F'h$</th>
<th>$F'c$</th>
<th>Calc.Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.3</td>
<td>0.2324 ± 0.15</td>
<td>0.5422 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>0.3873 ± 0.30</td>
<td>0.3873 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td>0.5422 ± 0.30</td>
<td>0.2324 ± 0.15</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3</td>
<td>0.3000 ± 0.20</td>
<td>0.7000 ± 0.20</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5000 ± 0.30</td>
<td>0.5000 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>0.7000 ± 0.20</td>
<td>0.3000 ± 0.20</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3</td>
<td>0.3550 ± 0.30</td>
<td>0.8283 ± 0.15</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5</td>
<td>0.5916 ± 0.30</td>
<td>0.5916 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>0.8283 ± 0.15</td>
<td>0.3550 ± 0.30</td>
<td>100</td>
</tr>
</tbody>
</table>

Total Number of Vector Pairs : 900

Table 3  Process Parameters for the Model Simulation

$t_0 = 4 \text{ sec}, \quad F_0 = 5 \text{ cm}^3/\text{sec}, \quad T_h = 85 \text{ }^\circ\text{C}, \quad T_c = 25 \text{ }^\circ\text{C},$

$A = 12 \text{ cm}^2, \quad h_0 = 50 \text{ cm}, \quad \alpha = 1$
Table 4  Steady-States Points for the Calculation

<table>
<thead>
<tr>
<th>( h' )</th>
<th>( T' )</th>
<th>( F'h )</th>
<th>( F'^c )</th>
<th>Calc. Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.0447</td>
<td>0.4025 ± 0.20</td>
<td>50</td>
</tr>
<tr>
<td>0.1</td>
<td>0.3</td>
<td>0.1342 ± 0.10</td>
<td>0.3131 ± 0.10</td>
<td>100</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>0.2236 ± 0.10</td>
<td>0.2236 ± 0.10</td>
<td>100</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7</td>
<td>0.3131 ± 0.10</td>
<td>0.1342 ± 0.10</td>
<td>100</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9</td>
<td>0.4025 ± 0.20</td>
<td>0.0447</td>
<td>50</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1</td>
<td>0.0775 ± 0.05</td>
<td>0.6971 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3</td>
<td>0.2324 ± 0.15</td>
<td>0.5422 ± 0.35</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>0.3873 ± 0.25</td>
<td>0.3873 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7</td>
<td>0.5422 ± 0.35</td>
<td>0.2324 ± 0.15</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>0.9</td>
<td>0.6971 ± 0.25</td>
<td>0.0775 ± 0.05</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.1000 ± 0.05</td>
<td>0.9000 ± 0.05</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3</td>
<td>0.3000 ± 0.25</td>
<td>0.7000 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5000 ± 0.25</td>
<td>0.5000 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7</td>
<td>0.7000 ± 0.25</td>
<td>0.3000 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>0.9</td>
<td>0.9000 ± 0.05</td>
<td>0.1000 ± 0.05</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1</td>
<td>0.1183</td>
<td>1.0649 (NA)</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.3</td>
<td>0.3550 ± 0.30</td>
<td>0.8283 ± 0.15</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5</td>
<td>0.5916 ± 0.25</td>
<td>0.5916 ± 0.25</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>0.8283 ± 0.15</td>
<td>0.3550 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9</td>
<td>1.0649 (NA)</td>
<td>0.1183</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>0.1342</td>
<td>1.2075 (NA)</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.3</td>
<td>0.4025 ± 0.35</td>
<td>0.9391 ± 0.05</td>
<td>100</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5</td>
<td>0.6708 ± 0.30</td>
<td>0.6708 ± 0.30</td>
<td>100</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7</td>
<td>0.9391 ± 0.05</td>
<td>0.4025 ± 0.35</td>
<td>100</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>1.2075 (NA)</td>
<td>0.1342</td>
<td></td>
</tr>
</tbody>
</table>

Total Number of Vector Pairs :  2000
which include first three steps of the steady-state. Two typical results of the process response are shown in Figs. 7 and 8. As can be seen, after a period of steady-state, the system was exited by the pseudo-random binary disturbances. However, resulting output perturbations of liquid level and temperature are maintained in the range of ± 0.15.

When the sample time exceeds the time history length, the first vector pair is generated. As mentioned above, the components of the input vector are the past and present values of liquid level, temperature, and two inlet flow rates. On the other hand, the components of the output vector are liquid level and temperature of one step ahead in the future. After the first generation of a vector pair, the window moves one step down the data base and another vector pair is generated. A training data set was constructed by repeating the same procedure. A flow diagram of the procedure is shown in Fig. 9. For the nine steady-state perturbations, a training data set that consists of 900 vector pairs (TDS-900) was constructed from the steady-state conditions listed in Table 2. Similarly, another training data set that consists of 2000 vector pairs (TDS-2000) was constructed spanning a wider range of steady-state conditions (Table 4).\(^7\)

\(^7\) A data screening process can be used to eliminate redundant data pairs and reduce the learning time of BP network \([\text{Broussard, 1990}].\) However, all the 900 or 2000 vector pairs were used in this study. This size of data set will not increase the learning time substantially, and the procedures of model construction are more straight-forward.
Fig. 7  Response of the System to the Pseudo-Random Binary Disturbances (1)
Fig. 8  Response of the System to the Pseudo-Random Binary Disturbances (2)
Normalized Raw Data

<table>
<thead>
<tr>
<th>t'</th>
<th>h'</th>
<th>T'</th>
<th>Fh'</th>
<th>P'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3.0000e-01</td>
<td>3.0000e-01</td>
<td>2.3238e-01</td>
<td>5.4222e-01</td>
</tr>
<tr>
<td>1</td>
<td>3.0000e-01</td>
<td>3.0000e-01</td>
<td>2.3238e-01</td>
<td>5.4222e-01</td>
</tr>
<tr>
<td>2</td>
<td>3.0000e-01</td>
<td>3.0000e-01</td>
<td>2.3238e-01</td>
<td>5.4222e-01</td>
</tr>
<tr>
<td>3</td>
<td>3.0000e-01</td>
<td>3.0000e-01</td>
<td>3.8238e-01</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>4</td>
<td>2.9511e-01</td>
<td>3.2110e-01</td>
<td>8.2379e-02</td>
<td>8.4222e-01</td>
</tr>
<tr>
<td>5</td>
<td>3.0021e-01</td>
<td>2.9828e-01</td>
<td>3.8238e-01</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>6</td>
<td>2.9530e-01</td>
<td>3.1948e-01</td>
<td>3.8238e-01</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>7</td>
<td>2.9061e-01</td>
<td>3.3956e-01</td>
<td>8.2379e-02</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>8</td>
<td>2.7633e-01</td>
<td>3.3635e-01</td>
<td>8.2379e-02</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>9</td>
<td>2.6268e-01</td>
<td>3.3310e-01</td>
<td>8.2379e-02</td>
<td>2.4222e-01</td>
</tr>
<tr>
<td>10</td>
<td>2.4965e-01</td>
<td>3.2982e-01</td>
<td>3.8238e-01</td>
<td>8.4222e-01</td>
</tr>
</tbody>
</table>

(Vector Pair 1)

3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01

(Vector Pair 2)

3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01  3.8238e-01  2.4222e-01
2.9511e-01  3.2110e-01

(Vector Pair 3)

3.0000e-01  3.0000e-01  2.3238e-01  5.4222e-01
3.0000e-01  3.0000e-01  3.8238e-01  2.4222e-01
2.9511e-01  3.2110e-01  8.2379e-02  8.4222e-01
3.0021e-01  2.9828e-01

Fig. 9 Procedure of Training Data Set Construction
b. Training of BP Network

There are two learning parameters to be specified in the BP algorithm. These are the learning rate and the momentum. The learning rate determines the degree of change in the weights per learning cycle or epoch and determines the speed at which the BP algorithm converges to the solution. The momentum determines whether the direction of learning progresses in the same manner as the previous epoch. Both are fractions. In this study, the learning rate is set at 0.3 and the momentum is set at 0.7. With these values, successful convergence is always obtained.

Once a training set is constructed and all the parameters required are set, the learning process is ready to begin. All the weights are set at small random values, and learning progresses in epochs. During each epoch, small adjustments in the weights are made according to the learning algorithm.

Another variable that affects the network representation is the degree of learning, specifically the number of times that the whole data set is presented to the network (learning

---

8 If all the weights start out with equal values and if the solution requires unequal weights, the network system can never learn. This is because error is back-propagated in proportion to the weight of each connection. This means that all hidden neurons connected directly to the output neurons will get identical error signals. Since the weights changes depend on the error signals, the weights from those neurons to the output neurons must always be the same [McClelland and Rumelhart, 1988]
epochs). With the increase of this learning epoch, the recalled outputs of the training set come closer to their respective target values. For this reason, the sum of the squared error between output and target for the training, the Total Squared Error (TSE), is one measure of the degree of convergence of the mapping. Typically, the TSE is checked after each learning epoch and if it is below a prescribed value, then learning stops, otherwise it continues.

TSE behaviors for the training data sets, TDS-900 and TDS-2000, are shown in Fig. 10. In this study, TSE of 0.18, which means the average absolute error between the output and the target is one percent, was used for TDS-900. Using the same criterion, TSE of 0.4 was used for the training data set, TDS-2000. The effect of the TSE value on the mapping is also examined for TDS-2000. This issue will be discussed in detail in section 4. The result of each mapping is saved as a data file that consists of 452 weight values.

4. Simulation with the BP Network Model
   a. Introduction

Once the mapping is available, the neural network can be used to simulate system responses as a process model. In this study, the responses of liquid level and temperature to a step change in inlet flow rates will be simulated based on the neural network model created in the last section. The accuracy of neural network model prediction will be evaluated
Fig. 10 Total Squared Error vs. No. of Training Epochs
using the numerical integration results of the differential equations as the "actual" model.

Two types of simulation are used based on the BP network model, namely one-step-ahead simulation and recursive simulation. In the one-step-ahead simulation, past and present data from the actual system are used as the input to the network. From these input data, the network predicts the system behavior one step into the future. In the recursive simulation, the same manner as that of one-step-ahead simulation is used for the first-step prediction. However, the predicted value from the previous step is used to construct the input data for the next step. In this manner, the network model is used to move the simulation forward in time. In the one-step-ahead simulation, modeling error is not included as part of the input to the network, whereas in the recursive simulation it is.

b. Results of One-Step-Ahead Simulation

The process of one-step-ahead simulation is shown in Fig. 11. During the first three steps, the liquid level and the temperature in the mixing tank are kept at steady-state values. Two flow rates are also kept at the values corresponding the steady-state point. Then, a first input vector to the network is constructed. This vector is submitted to the network model, and the first prediction of liquid level and temperature is performed. Whereas, correct
Fig. 11 One-Step-Ahead Simulation Scheme using BP Neural Network
liquid level and temperature are obtained from the actual process. These correct data become a part of second input vector as well as the specified hot and cold stream flow rates. Components of the first input vector except for the oldest ones also become a part of second input vector. However, no network model prediction is included as part of the input vector. The input vector is always derived from the actual system at each sample time in the one-step-ahead simulation.

Comparisons of the response curves between the one-step-ahead model simulation and those from the actual model (differential equations (4) and (5)) are shown in Figs. 12 ~ 15 and Figs. 16 ~ 19. In these simulations, step changes in the hot and cold stream flow rates are given to the system, which are shown in the upper panel. The system responds to these step changes and travels to a new steady-state point as time goes on. Results shown in Figs. 12 ~ 15 are based on the BP network model using the training data set TDS-900 (Table 2) with total squared error (TSE) of 0.18. Figs. 16 ~ 19 are the results using the training data set TDS-2000 (Table 4) with TSE of 0.4. It must be noted that the difference between two network models is expressed only in the values of weight data since there is no difference of the network structure between them.

As can be seen, the response predicted by the network model shows good agreement with the actual response.
Fig. 12  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (1)
(Network Model: TDS-900 with TSE of 0.18)
Fig. 13  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (2)
(Network Model: TDS-900 with TSE of 0.18)
Fig. 14  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (3)
(Network Model:TDS-900 with TSE of 0.18)
Fig. 15  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (4)
(Network Model: TDS-900 with TSE of 0.18)
**Fig. 16** Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (5)
(Network Model: TDS-2000 with TSE of 0.4)
Fig. 17  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (6)
(Network Model: TDS-2000 with TSE of 0.4)
Fig. 18  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (7)
(Network Model: TDS-2000 with TSE of 0.4)
Fig. 19  Response of the System to the Step Change in Inlet Flow Rates:
One-Step-Ahead Simulation (8)
(Network Model: TDS-2000 with TSE of 0.4)
However, the prediction error in the region above 0.8 or below 0.2 is a little notable in Figs. 12 ~ 15. This is a typical disadvantage of the BP network. The network model used in these simulations was trained by the data set TDS-900. It does not contain enough data of liquid level and temperature whose value is above 0.8 or below 0.2. All the training data of TDS-900 were generated around the steady-state points of 0.3, 0.5, and 0.7.

The capacity of a BP network to interpolate and extrapolate is limited. The network can usually follow a general trend; however, it will generate an offset from the correct value if asked to extrapolate too far. When the system enters an operating region on which the network has no knowledge, then the network model will probably perform poorly as shown in these figures. One way to circumvent this problem is to construct a training data set which covers the complete dynamic range of the system. The training data set TDS-2000, which includes the process data around the steady-state points of 0.1 and 0.9, corresponds to this complete data set. As can be seen in Figs. 16 ~ 19, the network model trained by this data set shows excellent predictions over the entire range of the system.

The effect of learning depth is also examined in this one-step-ahead simulation. The same step change responses were simulated by four BP models. Each model was trained by TDS-2000, and has a different learning depth, namely TSE
values of 0.4, 0.196, 0.144, or 0.1. These values correspond
to the average absolute error between the outputs and targets
of 1, 0.7, 0.6, or 0.5 %, respectively. The relationship
between number of training epochs and TSE is previously
shown in Fig. 10. The effects of TSE on the response curves
are shown in Figs. 20 - 23. Although the best value of TSE
varies in simulation by simulation, TSE of 0.7 % shows a good
overall performance. It predicts the best response except in
Figs. 21(b) and 23(b), and gives no prediction error larger
than 0.01 in all the cases. In addition, 0.7 % of TSE is a
good choice from the view point of learning time. As shown in
Fig. 10, it takes only 350 epochs to improve the TSE from
1.0 % to 0.7 %, whereas it takes 1000 epochs from 0.7 % to
0.6 %. In other words, the normalized slope of the TSE curve
becomes almost one-tenth of the previous value around this
point. This slope of TSE also can be used as a criterion to
stop the network training instead of TSE itself. However,
behaviors of TSE curve vary with the network structure and
the training data set, and such an obvious inflection point
can not be always obtained.\footnote{When the number of vector pairs in the training data set changes, a
method called tested training can be used [Pollard, 1990]. This
procedure consists of training the network for a short number of
learning cycles then performing a test to check if learning is adequate.
The test consists of measuring error between target and network output
for a test set of vector pairs, whose number is fixed at an appropriate
value.}


does not provide further context or a question.
Fig. 20  Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (1)  
(Fh':0.5367→0.7367, Fc':0.3577→0.0577)
Fig. 21 Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (2) (Fh': 0.5367→0.2367, Fc': 0.8050→0.3050)
Fig. 22  Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (3)  
\( \Phi_h':0.3578 \rightarrow 0.1578, \  F_c':0.0894 \rightarrow 0.5894 \)
Fig. 23  Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: One-Step-Ahead Simulation (4)
(Fn': 0.6572→0.8572, Fc': 0.4382→0.5382)
found by experience for each case.

**c. Results of Recursive Simulation**

Even though an accurate one-step-ahead prediction is important, it may not be sufficient to construct a predictive control system or make a decision on changes in the existing controller behavior. A view of many steps into the future is often required for such applications. For this reason, recursive simulation technique is also performed in this study. The process of recursive simulation is shown in Fig. 24. Just as in the one-step-ahead simulation, it begins by forming a vector from the initial conditions of the input variables. This input vector is submitted to the network model, and the first prediction of liquid level and temperature is performed. These predicted values are used to form the second input vector together with the specified hot and cold stream flow rates. Components of the first input vector except for the oldest ones also become a part of second input vector. Again, this second vector is submitted to the network model, and liquid level and temperature in the next time step are predicted. In this manner, the simulation proceeds into the future. Any modeling error in predicting the liquid level and temperature are used as the components of next input vector. This magnifies the prediction error and may produce a different response of the system from the actual response.
Fig. 24 Recursive Simulation Scheme using BP Neural Network
Comparisons of the response curves between the recursive model simulation and those of the actual model are shown in Figs. 25 and 26. Figures of the step change in flow rates are abbreviated from these figures, and each figure contains two simulations. The model used in these simulations is constructed from the training data set TDS-2000, and TSE of 0.196 (0.7%) is used as a criterion of the mapping convergence. As mentioned in the previous section, this selection gave the best performance for the one-step-ahead simulations.

Although the deviation from the actual response is larger than that of the one-step-ahead simulations, the predicted behavior by the recursive simulation is still excellent. The prediction error of liquid level or temperature is kept within 6% in all four simulations. From these simulation results, the conclusion is made that the network model can identify the non-linear two-input, two-output system accurately.

Effects of TSE value on the same system responses are also examined for recursive simulations. The results are shown in Figs. 27~30. Since the prediction error of the recursive simulation can be magnified with time steps, the difference of the system responses between the network models is more distinctive. In other words, recursive simulation is more useful for verifying a network model whose accuracy is affected by the choice of TSE value. As can be seen in these
Fig. 25 Response of the System to the Step Change in Inlet Flow Rates: Recursive Simulation (1) (Network Model: TDS-2000 with TSE of 0.196)
(a) \( Fh': 0.3578 \rightarrow 0.1578, \ Fc': 0.0894 \rightarrow 0.5894 \)

(b) \( Fh': 0.6572 \rightarrow 0.8572, \ Fc': 0.4382 \rightarrow 0.5382 \)

Fig. 26 Response of the System to the Step Change in Inlet Flow Rates: Recursive Simulation (2) (Network Model: TDS-2000 with TSE of 0.196)
Fig. 27 Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: Recursive Simulation (1) (Ph': 0.5367 → 0.7367, Fc': 0.3577 → 0.0577)
Fig. 28 Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: Recursive Simulation (2)
(Fh':0.5367→0.2367, Fc':0.8050→0.3050)
Fig. 29 Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates:
Recursive Simulation (3)
\((F_{h}': 0.3578 \rightarrow 0.1578, F_{c}': 0.0894 \rightarrow 0.5894)\)
Fig. 30  Effect of TSE on the Response of the System to Step Change in Inlet Flow Rates: Recursive Simulation (4)

$P_{h'}: 0.6572 \rightarrow 0.8572, F_{c'}: 0.4382 \rightarrow 0.5382$
figures, TSE of 0.196 (0.7%) is the best choice for Figs. 27(a), 28(a), 29(a) & (b), and 30(a). This is consistent with the results from the one-step-ahead simulations.

5. Discussion

In this chapter, a two-input, two-output mixing tank system was introduced as an example of MIMO processes. The BP network was trained with open-loop data of this system, and the trained network was used for the simulation of process responses. Basically, the resulting network predicts the process response only one step in the future as used in the one-step-ahead simulation. However, the model can be modified to predict the response many steps in the future if the predicted response (output vector of the network) is used to create the next input vector recursively.

The results of the recursive as well as the one-step-ahead simulation show an excellent agreement with the actual responses, and the validity of the network model was confirmed. It must be noted that the network must be well-trained to give an accurate prediction. "Well-trained" means that the network is trained with an appropriate training data set and an appropriate learning depth.

Although only open-loop system responses are considered in this chapter, closed-loop systems can be treated. Consider a feedback control system shown in Fig. 31. If the network model is used instead of the actual process, this control
Fig. 31 Generalized Feedback Control System and BP Network Model

C: Controlled Variable
M: Manipulated Variable
D: Disturbance
E: Error Signal
B: Primary Feedback Signal
R: Set Point
system can be simulated in a standard manner. The network model receives the signal of specified flow rate from the controller, and produces the change in liquid level or temperature. This is nothing but a modified recursive simulation process. Tuning of existing controllers can be done by this closed-loop "network" system. If two PI controllers are used in this system, the best combination of two proportional gains and two integral times can be found by the case study simulations. However, many case studies will be required to identify the best combination of four parameters.

In principle, once a well trained BP network model is available, then it can be used in a straightforward manner for process control. Instead of tuning the existing PI(D) controllers, a predictive control system using the network model is proposed and examined in the following chapter. This is because even two well-tuned PI(D) controllers will not solve the difficult features of the mixing tank problem, namely process interaction and non-linearity.
IV. Control by BP Network

1. Control System Configuration

a. Flow Rates Selection by Predictor

As mentioned earlier, the exact solution of controller actions using the network model cannot be obtained in a direct manner because of its non-invertability.\textsuperscript{10} Instead, a solution can be chosen from the results of a number of case studies, which can be directly performed based on the network model.

Due to the special feature of the network mapping, an algorithm to select the desirable controller action is proposed. The proposed approach is a modification of Forward Modeling Controller [Otto, 1986]. For the sake of clarity, a SISO system is used to illustrate the basic working principles. Suppose that a set point change in liquid level or temperature is given to the process at time $t_k$. The type of set point change is not critical. It might be a step, a ramp, or of any functional form. A step change is chosen in Fig. 32. Upon knowing the profile of the desired set point in the future, several open-loop recursive simulations are

\textsuperscript{10} To estimate controller actions, a different BP network is required. Namely, manipulated variables one step in the future are used for the network output, and controlled variables one step in the future are used for the network input. This inverse mapping model can be obtained by another series of mapping procedures, vector pair construction, network training, model verification, and so on. Also, the raw data used for the original mapping are often inadequate for the inverse mapping.
Fig. 32 Flow Rate Selection by Predictor
performed by the BP network model (predictor) with various admissible flow rates. A total of six evenly spaced flow rates between 0 and 1 are shown in this example. The number of steps into the future to be simulated, \( N_r \), is one of the few parameters to be specified in this algorithm. Two steps of recursive simulation are shown in Fig. 32. For each simulation, the Integral Absolute Error (IAE) between the predicted response and the set point is calculated. Based on the IAE evaluation, the flow rate which gives the minimum value is selected for the next step. Although the performance of flow rate is rated on \( N_r \) steps into the future, the optimal flow rate is only implemented for one sampling time. One step later, at time \( t_{k+1} \), another set of simulations is performed and the next "best" flow rate is selected. The process will respond to the new selected flow rate again. In this manner, the liquid level or the temperature is controlled by the discrete flow rate. (In this particular example, the flow rate will not take any values except for 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0.) If only two flow rates such as 0.0 and 1.0 are used, this is nothing but an on-off or two-position controller.

Since the selected flow rate is used only for the next step, too many steps of recursive simulation will not give good process response for the next step. Furthermore, computation time required for the simulation increases in proportion to the number of recurrences. On the other hand,
too few steps limit the degree of freedom in the choice of the controller action. Such limitation may lead to drastic cycling in the control action. In this study, Nr=2 was selected according to the simulation results of servo problem as discussed in section 2.a.i).

The two-input and two-output system requires 36 (=6×6) combinations of hot and cold stream flow rates. In each combination, the responses of liquid level and temperature are predicted by the recursive simulation, and the sum of IAE-1 (for liquid level) and IAE-2 (for temperature) is calculated. The flow rates that give the minimum of total IAE are selected for the next step. The total IAE is given by the following equation.\(^{11}\)

\[
\text{IAE (Total)} = \text{IAE-1 (liquid level)} + \text{IAE-2 (temperature)}
\]

\[
= \sum_{k=m}^{m+Nr} \{ |h_s'(k) - h'(k)| + |T_s'(k) - T'(k)| \}
\]

where, \(m\) : sample time

\(Nr\) : number of steps for each recursive simulation

\(h_s'(k), T_s'(k)\) : set point of normalized liquid level or temperature at sample time \(k\)

\(^{11}\) Although two IAE are treated equally in this study, the weighing factor (WF) can be used. The equation for the total IAE is given by

\[
\text{IAE (Total)} = \text{WF} \ast \text{IAE1} + (1-\text{WF}) \ast \text{IAE2}
\]

WF takes the value between 0 and 1 depending on the priority of two controlled variables.
h'(k), T'(k) : predicted values of normalized liquid
level or temperature at sample time k

If more case studies are allowed for each step, a finer
selection of flow rates will be possible. The modified scheme
is shown in Fig. 33. Suppose that a flow rate of 0.4 is
selected at the first stage. If another selection is
performed based on the first selection of 0.4, a finer flow
rate of 0.44 can be chosen at the second stage. The minimum
difference between two flow rates \( \Delta F' \) decreases from 0.2 to
0.08, and the number of case studies increases from 36 to
only 52. Similarly, \( \Delta F' \) becomes 0.02 with 66 case studies as
shown in Fig. 34. The finer flow rate selection has an
advantage not only for controllability but also for smooth
controller action. This issue is also examined in section
2.a.i).

A grid search method, which is less efficient but can
provide a fuller picture of the effects of flow rate
selection on controller performance, was adopted in this
study. In actual implementation, other more efficient search
algorithms such as the simplex method or bisection search can
be used to determine the optimal flow rate. It should be
pointed out that the magnitude of \( \Delta F' \) depends on the process
gain. The larger the gain is, the smaller the \( \Delta F' \) that should
be used. With a more efficient algorithm and a faster
computation machine (or specialized neural network chip), the
a) 1st Selection Only;

\[ F_h' \text{ or } F_c' \]

\[
\begin{array}{cccccc}
0.0 & 0.2 & 0.4 & 0.6 & 0.8 & 1.0 \\
\end{array}
\]

for 2x2 System: \( 6 \times 6 = 36 \) case studies

\[ \Delta F' = 0.2 \]

---

b) Up to 2nd Selection;

\[ F_h' \text{ or } F_c' \]

\[
\begin{array}{cccc}
0.2 & 0.4 & 0.6 & 0.8 \\
\end{array}
\]

1st Selection

If 1st Selection = 0.4

\[
\begin{array}{cccccc}
0.20 & 0.28 & 0.36 & 0.44 & 0.52 & 0.60 \\
\end{array}
\]

2nd Selection

for 2x2 System: \( 4 \times 4 + 6 \times 6 = 52 \) case studies

\[ \Delta F' = 0.08 \]

---

Fig. 33 Finer Flow Rate Selection Scheme (1)
c) Up to 4th Selection:

\[
\begin{array}{c}
\text{Fn'} \text{ or Fe'} \\
\downarrow \\
(0.0) \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ (1.0) \\
\downarrow \\
(0.2) \ 0.28 \ 0.36 \ 0.44 \ 0.52 \ (0.6) \\
\downarrow \\
(0.36) \ 0.40 \ 0.44 \ 0.48 \ (0.52) \\
\downarrow \\
0.44 \ 0.46 \ 0.48 \ 0.50 \ 0.52 \ (1.0) \\
\end{array}
\]

:1st Selection
If 1st Selection = 0.4

:2nd Selection
If 2nd Selection = 0.44

:3rd Selection
If 3rd Selection = 0.48

:4th Selection

for 2x2 System:
\[4 \times 4 + 4 \times 4 + 3 \times 3 + 5 \times 5 = 66 \text{ case studies}\]

\[\Delta F' = 0.02\]

Fig. 34 Finer Flow Rate Selection Scheme (2)
search time for small ΔF' probably will not be a major problem.

b. Error Correction of the Predictor

Although the network model can predict the process response with reasonable accuracy, a small error usually exists between the predicted and the actual responses. Since the control algorithm is an open-loop type, any modeling error will subsequently cause an offset from the desired value. Even if the model is extremely accurate and imposes no error during the identification step, changes in the process may occur over time due to such phenomenon as fouling or catalyst degradation. Under such a scenario, the prediction error must also be considered. In other words, under realistic operating conditions, an error correction is required to maintain satisfactory performance of a model predictive controller. Needless to say, the model must be updated or the neural network must be retrained if the process behavior changes significantly. Although model update was also considered in some cases, a detailed discussion about this issue is not treated in this study.

The error correction of predicted response used in this study is schematically shown in Fig. 35. At time t_{k-1}, an one-step-ahead simulation is performed by the network model, and the process response at time t_k is predicted. One step later, the actual response at t_k is measured, and the prediction
Fig. 35 Error Correction of Predicted Response
error, \( \varepsilon_k \) is calculated from the difference between the predicted value and the actual process value. (In the figure, this value is illustrated as a negative value.) As mentioned above, a number of recursive simulations are performed to select the desired control actions. Only one of the two predicted responses is shown in the figure. In each simulation, the prediction error \( \varepsilon_k \) is used to correct the predicted response at every recursive step. After the first step prediction, \( \varepsilon_k \) is subtracted from the predicted response at \( t_{k+1} \), and the corrected response at \( t_{k+1} \) is obtained. The second step prediction starts from this corrected response. In other words, the corrected response is used in the input vector instead of the predicted response. In the same manner, the recursive simulation is continued up to the given number of recurrence. After finishing all the case studies by this corrected recursive simulation, the best flow rate is selected and implemented. With this selected flow rate, one-step-ahead simulation is performed and the predicted response at \( t_{k+1} \) is used for the error correction of the next step.\(^{12}\) During implementation, only one prediction error is always stored for each controlled variable, and it is updated every step.

\(^{12}\) Instead of doing the one-step-ahead simulation, the same result is obtained by storing the response of the recursive simulation with the selected flow rate. For the first prediction, there is no difference of predicted values between two simulation processes.
c. Control System Configuration

Using the flow rate selection and the error correction scheme mentioned in the previous section, a model predictive control system is constructed. The control system configuration is shown in Fig. 36. The model predictive controller receives set point data \( R \) for the next step and current process data \( C \). Using these data together with the stored historical data of \( C \) and \( M \) (manipulated variables: flow rates), the controller action, \( M \), for the next step is selected from some discrete values, which minimize certain objective functions. Thus, this can be referred to as a multi-positional feedback controller.

Two predictors are actually involved in this system. Predictor-1 is used for the selection of flow rates through recursive simulation. Predictor-2 is an one-step-ahead simulator that determines the error correction term to be used by the predictor-1. Both predictors are based on the same network model. In this chapter, the network model trained up to 0.7 % of TSE with the data set TDS-2000 is used for simulations of the controller performance except for some specifically noted cases.

2. Results of the Servo Problem

a. Response to the Step Change in Set Points

i) Effects of Parameters on Controller Performance

The closed-loop system behavior for set-point changes
C: Controlled Variable
M: Manipulated Variable
D: Disturbance
R: Set Point
T: Model Prediction
$\epsilon$: Prediction Error

Predictor-1: Recursive Simulation
Predictor-2: 1-step-ahead Simulation

Fig. 36 Predictive Control System by BP Network Model
referred as the servo problem is examined for the network model predictive (NMP) controller. In this system, only step changes in set points will be considered. It is assumed that no disturbance occurs, i.e. $D = 0$.

Simulation results for the set point changes $h_0: 0.6 \rightarrow 0.8$ and $T_0: 0.6 \rightarrow 0.4$ are shown in Figs. 37 ~ 40. In each figure, the upper panel shows the actions of manipulated variables, $F_0$ and $F_c'$, and the lower panel shows the responses of controlled variables, $h'$ and $T'$. Fig. 37 shows a result with no predictor correction and with only first flow rate selection ($\Delta F' = 0.2$). Both liquid level and temperature (lower panel) respond to their set point changes, and they approach the new set points smoothly. However, a small steady-state error (offset) remains in the liquid level. This offset is caused by the model prediction error, and will persist without the error correction.

Results with predictor correction are shown in Figs. 38, 39, and 40. The simulations are performed with flow rate selection schemes of $\Delta F' = 0.2$, $\Delta F' = 0.08$, and $\Delta F' = 0.02$, respectively. As a result of error correction, offset in the liquid level and temperature were eliminated in all simulations. In addition, there is no remarkable difference in the response of liquid level or temperature among the three cases. However, as shown in the upper panel of Figs. 38, 39, and 40, a finer flow rate selection scheme yields a smoother controller action. The frequency and amplitude of
Fig. 37 Response to the Step Change in Set Points by NMP Controller (1)
- No Predictor Correction
- 1st Flow Rates Selection Only: ΔF' = 0.2
  (hs' = 0.6 → 0.8, Ts' = 0.6 → 0.4)
Fig. 38  Response to the Step Change in Set Points by NMP Controller (2)
• with Predictor Correction
• 1st Flow Rates Selection Only: $\Delta F' = 0.2$
  ($h_s' : 0.6 \rightarrow 0.8$, $T_s' : 0.6 \rightarrow 0.4$)
Fig. 39  Response to the Step Change in Set Points by NMP Controller (3)

• with Predictor Correction
• up to 2nd Flow Rates Selection: \( \Delta F' = 0.08 \)
  \( (hs': 0.6 \to 0.8, Ts': 0.6 \to 0.4) \)
Fig. 40  Response to the Step Change in Set Points by NMP Controller (4)  
- with Predictor Correction  
- up to 4th Flow Rates Selection: $\Delta F' = 0.02$  
  ($hs': 0.6 \rightarrow 0.8$, $Ts': 0.6 \rightarrow 0.4$)
the controller action become smaller with a decrease in $\Delta F'$. Simulation results for another set point change $hs':0.2\rightarrow 0.2$ and $Ts':0.3\rightarrow 0.7$ are shown in Figs. 41 ~ 44. Since the liquid level is kept at a lower level, the temperature response is more sensitive than the previous simulation. Again, the liquid level and the temperature show excellent responses to these set point changes. However, Figs. 41 and 42, in which $\Delta F'=0.2$ is used, show a slight offset in liquid level or temperature. As a result of increased process gain, a finer flow rate selection as well as the predictor correction is required for satisfactory operation at such a low liquid level.

The process responses between the results in Figs. 43 ($\Delta F'=0.08$) and 44 ($\Delta F'=0.02$) are almost indistinguishable. As expected, the result with $\Delta F'=0.02$ shows better behavior with regard to the controller action again. Rapid changes of the manipulated variable action tend to cause accelerated failure of the controller actuator, namely control valve or pump. Therefore, if smooth controller action is more important in judging the quality of control, a small value of $\Delta F'$ will be required.

The effect of $\Delta F'$ on controller performance is summarized in Fig. 45. In this figure, two parameters are used to characterize the process response and the controller action, respectively. The first parameter is the Integral Absolute Error (IAE) between process responses and set
Fig. 41  Response to the Step Change in Set Points by NMP Controller (5)
- No Predictor Correction
- 1st Flow Rates Selection Only: $\Delta F' = 0.2$
  ($hs': 0.2 \rightarrow 0.2, Ts': 0.3 \rightarrow 0.7$)
Fig. 42  Response to the Step Change in Set Points by NMP Controller (6)
- with Predictor Correction
- 1st Flow Rates Selection Only : $\Delta F' = 0.2$
  ($h' = 0.2 \rightarrow 0.2$, $T' = 0.3 \rightarrow 0.7$)
Fig. 43 Response to the Step Change in Set Points by NMP Controller (7)
- with Predictor Correction
- up to 2nd Flow Rates Selection : $\Delta F' = 0.08$
  ($h_s': 0.2 \rightarrow 0.2$, $T_s': 0.3 \rightarrow 0.7$)
Fig. 44 Response to the Step Change in Set Points by NMP Controller (8)  
• with Predictor Correction  
• up to 4th Flow Rates Selection : ΔF' = 0.02  
  (hs': 0.2 → 0.2, Ts': 0.3 → 0.7)
points, which is also used in the flow selection scheme. The second one is called the Integral Controller Action (ICA). These parameters are defined by the following equations.

\[
\text{IAE} = \sum_{k=0}^{M} \{|h_{s}(k) - h'(k)| + |T_{s}(k) - T'(k)|\} \\
\text{ICA} = \sum_{k=0}^{M} \{|F_{h}(k+1) - F_{h}(k)| + |F_{c}(k+1) - F_{c}(k)|\}
\]

where, \( k \): sample time
\( M \): final sample time for calculation of IAE or ICA
\( h_{s}(k), T_{s}(k) \): set point of normalized liquid level or temperature at sample time \( k \)
\( h'(k), T'(k) \): normalized liquid level or temperature at sample time \( k \)
\( F_{h}(k), F_{c}(k) \): normalized flow rate of hot or cold stream at sample time \( k \)

For the calculation of IAE or ICA, the number \( M \), defined in equations (7) and (8), must be chosen. A large \( M \) tends to emphasize the cyclic behavior or offset caused by modeling error and rough flow rate selection. On the other hand, \( M \) should be large enough to allow the process response to reach its steady-state values. On the basis of these consideration,
Fig. 45 Effect of $\Delta F'$ on IAE and ICA ($M = 30$) for Set Point Changes
a value of 30 is used in this study.

The result shown in Fig. 45 indicated that values of $\Delta F'$ smaller than 0.08 will not significantly improve the IAE. Actually, IAE takes its minimum value at $\Delta F' = 0.08$ (Fig. 45-a). Whereas, smaller values of $\Delta F'$ are favorable with respect to ICA. Consequently, the choice of $\Delta F'$ depends on the priority of IAE or ICA. $\Delta F'$ of 0.02 was chosen in the following study. It must be noted that the computation time required for the optimization (flow rate selection) increases almost in proportion to the number of case studies. As shown in Figs. 33 and 34, the number of case studies is 36 for $\Delta F' = 0.2$ and 66 for $\Delta F' = 0.02$. In other words, only twice as much computation time is required for the choice of $\Delta F' = 0.02$ compared with that of $\Delta F' = 0.2$.

As mentioned earlier, many recursive simulations are performed to select the flow rates in every step, and the number of recurrences for each simulation, $N_r$, is one of the parameters to be specified in the algorithm. Although $N_r = 2$ is used in the simulations up to this point, its effect is also examined with the same set point changes. The results are summarized in Fig. 46. IAE has a minimum value at $N_r = 2$ for both cases. Whereas, ICA slightly decreases with the increase in this number, and the computation time for the case studies will increase proportionally. As in the case of $\Delta F'$, the choice of $N_r$ also depends on the priority of IAE or ICA (or the speed of computer), and $N_r = 2$ was chosen in this study.
Fig. 46 Effect of the Step Number of Recursive Simulation on IAE and ICA (M=30) for Set Point Changes: $\Delta F' = 0.02$
ii) Comparison with Tuned PI Controllers

In this section, the performance of the network model predictive (NMP) controller is compared with that of the conventional PI controllers. To provide a fair comparison, PI controllers must be tuned. For SISO systems, the Process Reaction Curve Method [Ziegler-Nichols, 1942; Cohen and Coon, 1953] and the Continuous Cycling Method [Ziegler and Nichols, 1942; Perry and Green, 1984] are frequently used. However, it is difficult to tune MIMO processes because of their interactions. Although a generalization of the Continuous Cycling Method [Niederlinski, 1971] is proposed for MIMO systems, it has not been widely applied.

In this study, the four controller parameters (two proportional gains and two integral times) are optimally determined to minimize the value of IAE by the case studies using a grid-search method as shown in Fig. 47. At first, five candidates for each parameter are selected, and 625 case studies are performed for all the combinations of four parameters. In each case study, the differential equations (4) and (5), which are treated as the actual process in this study, are numerically integrated and the IAE is evaluated for each parameter combination. Then, the combination that gives a minimum IAE is selected as the first result. Based on this first result, second candidates of parameters are selected and 625 case studies are performed again in the same manner. Repeating the same procedure several times (6 ~ 7
### 1st Candidates

<table>
<thead>
<tr>
<th>No.</th>
<th>K_I</th>
<th>tau_l</th>
<th>K_T</th>
<th>tau_t</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>1.0</td>
<td>5.1195</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>2.0</td>
<td>5.2668</td>
</tr>
<tr>
<td>3</td>
<td>1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>5.0</td>
<td>5.7102</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>10.0</td>
<td>6.2460</td>
</tr>
<tr>
<td>5</td>
<td>1.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>20.0</td>
<td>6.8751</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>1.0</td>
<td>-5.0</td>
<td>1.0</td>
<td>5.0859</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No.</th>
<th>K_I</th>
<th>tau_l</th>
<th>K_T</th>
<th>tau_t</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>624</td>
<td>40.0</td>
<td>20.0</td>
<td>-40.0</td>
<td>10.0</td>
<td>5.3418</td>
</tr>
<tr>
<td>625</td>
<td>40.0</td>
<td>20.0</td>
<td>-40.0</td>
<td>20.0</td>
<td>6.5314</td>
</tr>
</tbody>
</table>

### 1st Selection

<table>
<thead>
<tr>
<th>K_I</th>
<th>tau_l</th>
<th>K_T</th>
<th>tau_t</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0</td>
<td>1.0</td>
<td>-10.0</td>
<td>1.0</td>
<td>2.9557</td>
</tr>
</tbody>
</table>

### 2nd Candidates

<table>
<thead>
<tr>
<th>K_I</th>
<th>tau_l</th>
<th>K_T</th>
<th>tau_t</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0</td>
<td>0.5</td>
<td>-5.0</td>
<td>0.5</td>
<td>3.0271</td>
</tr>
<tr>
<td>2</td>
<td>20.0</td>
<td>0.5</td>
<td>-5.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>20.0</td>
<td>0.5</td>
<td>-5.0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

---

**Fig. 47** Tuning of PI Controller Parameters
times in this study), four parameters are obtained with reasonable accuracy.

The tuning results for three set point changes are listed in Table 5. Note that the best tuned controller parameters differ significantly for the three different set point changes. This difference is caused by process interactions and the non-linearity of the system. In other words, these are fine tuning results that are optimized for each specified set point change. Therefore, a detuning will be required to cover the full operation range for practical use. However, this approach was not investigated in this study.

The comparison of process responses between "best-tuned" PI controllers and Network Model Predictive (NMP) controllers are shown in Figs. 48 - 50. The values of IAE and ICA (M = 30) are also listed in the bottom of figures. The process responses obtained by the NMP controller and the PI controller that is tuned specifically for each set point change are almost the same. Actually, the NMP controller shows better results with respect to IAE (Figs. 49 and 50). It should be emphasized that the same set of parameters for the NMP controller such as $\Delta F' = 0.02$ and $N_r = 2$ are used for three different set point changes, whereas three different controller tunings are used by the PI controllers.

If the PI parameters tuned for the specified set point change is used for other conditions, the process will not be
### Table 5  Tuning Results of PI Controller Parameters for the Step Change in Set Points

<table>
<thead>
<tr>
<th>Set Points Change</th>
<th>K₁</th>
<th>ta₁</th>
<th>K₉</th>
<th>ta₉</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.1</td>
<td>36</td>
<td>0.9</td>
<td>-14</td>
<td>1.9</td>
<td>2.9280</td>
</tr>
<tr>
<td>No.2</td>
<td>35</td>
<td>0.5</td>
<td>-4.0</td>
<td>1.4</td>
<td>1.6612</td>
</tr>
<tr>
<td>No.3</td>
<td>7.3</td>
<td>3.0</td>
<td>-33</td>
<td>1.2</td>
<td>1.7262</td>
</tr>
</tbody>
</table>

K₁ : Proportional Gain for Liquid Level Controller  
ta₁: Integral Time for Liquid Level Controller  
K₉ : Proportional Gain for Temperature Controller  
ta₉: Integral Time for Temperature Controller

(Set Point Change)  
No.1: hs':0.6→0.8, Ts':0.6→0.4  
No.2: hs':0.2→0.2, Ts':0.3→0.7  
No.3: hs':0.1→0.4, Ts':0.7→0.5

(Pairing)  
Controlled Variable → Manipulated Variable  
Liquid Level (h') → Hot Stream Flow rate (Ph')  
Temperature (T') → Cold Stream Flow rate (Fc')
Fig. 48 Comparison of Process Responses between "Best-Tuned" PI Controller and NMP Controller (1): (ΔF' = 0.02, Nr = 2)

Step Change in Set Points
(hs': 0.6 → 0.8, Ts': 0.6 → 0.4)

IAE: PI Controller → 2.9280
     NMP Controller → 2.9877

ICA: PI Controller → 2.3901
     NMP Controller → 2.4646
Fig. 49 Comparison of Process Responses between "Best-Tuned" PI Controller and NMP Controller (2): ($\Delta F' = 0.02$, $N_r = 2$)

Step Change in Set Points
($h_s': 0.2 \rightarrow 0.2$, $T_s': 0.3 \rightarrow 0.7$)

IAE: PI Controller $\rightarrow 1.6612$
   NMP Controller $\rightarrow 1.5239$

ICA: PI Controller $\rightarrow 2.2462$
   NMP Controller $\rightarrow 3.1930$
Fig. 50  Comparison of Process Responses between "Best-Tuned" PI Controller and NMP Controller (3): ($\Delta F'=0.02$, $N_r=2$)

Step Change in Set Points
($h_s'=0.1 \rightarrow 0.4$, $T_s'=0.7 \rightarrow 0.5$)

IAE: PI Controller→1.7262
NMP Controller→1.6910

ICA: PI Controller→2.6665
NMP Controller→3.5128
controlled satisfactorily. This is illustrated in Figs. 51 (Set Point Change-2) and 52 (Set Point Change-3). In both figures, PI controllers whose parameters are tuned for Set Point Change-1 are used. Due to improper controller tuning, the process becomes unstable resulting in a cyclic behavior in liquid level or temperature response, and an on-off manner in the flow rates.

iii) Use of Less-Trained Network

Up to this point, the well-trained network model is used for process response simulations and control. The model is obtained by training the network up to the learning depth of TSE=0.7% with the data set TDS-2000. However, it might be difficult to achieve such a precise model in every situation. It is, therefore, of interest to examine the feasibility of using a less-trained network model instead of the well-trained one for control purposes.

Simulation results with a less-trained network model are shown in Figs. 53 and 54. The model is trained up to TSE=1.0 % with the data set TDS-900. As mentioned earlier, this model shows a notable prediction error in the region of above 0.8 or below 0.2. As shown in Fig. 53, the model error causes 5 % offset in temperature if no predictor correction is made in the NMP controller. Nevertheless, this size of error is still "correctable" by the NMP controller with predictor correction as shown in Fig. 54; the liquid level and temperature
Fig. 51  Response to the Step Change in Set Points by Ill-Tuned PI Controllers (1):
(Kl=36, tau_l=0.9, K_t=-14, tau_t=1.9)

Set Point Change:
h: 0.2 → 0.2, T_s: 0.3 → 0.7

IAE = 2.4936, ICA = 23.037
Fig. 52  Response to the Step Change in Set Points by Ill-Tuned PI Controllers (2):
(K1=36, taul=0.9, Kt=-14, taut=1.9)
Set Point Change:
hs':0.1 → 0.4, Ts':0.7 → 0.5
IAE = 2.5180, ICA = 22.8275
Fig. 53  Response to the Step Change in Set Points by NMP Controller (ΔF′=0.02, Nr =2) with less trained Network Model
- No Predictor Correction
  (hs′:0.6 → 0.8, Ts′:0.6 → 0.4)

IAE = 3.9415, ICA = 2.9446
Fig. 54  Response to the Step Change in Set Points by NMP Controller ($AF' = 0.02$, $Nr = 2$) with less trained Network Model (2):
- with Predictor Correction
  ($hs: 0.6 \rightarrow 0.8$, $Ts: 0.6 \rightarrow 0.4$)

$\text{IAE} = 3.0069$, $\text{ICA} = 3.0246$
smoothly follow the step changes in set point and show no offsets. It takes only 12 more steps to reach the new set point of liquid level compared with the result shown in Fig. 40, obtained by a better trained NMP controller. Note that the IAE values for both controllers are very comparable: 3.0069 for the less-trained and 2.9877 for the better-trained controllers, respectively.

b. Response to the Ramp Change in Set Points

To avoid unnecessary process upsets, it is a common practice to ramp the set points from one value to another, rather than making an abrupt step change. The predictive controller is favorable for such situations. If the slope of ramp change is constant, information about the future set point (one step or several steps) is available. Using this information about set point changes in the future, the predictive controller can choose the best action to follow the desired set point changes. Since the action of a PI controller is feedback in nature and is always determined by the present error, any future information of set point changes will not help to select a better controller action. In other words, the action of a PI controller is one step behind from that of the predictive controller for the ramp change.

Simulation results of the ramp change in set points by PI and NMP controllers are shown in Figs. 55 ~ 57. The
Fig. 55 Comparison of Process Responses between Tuned PI Controller and NMP Controller (1): ($\Delta F'=0.02$, $Nr=2$)

Ramp Change in Set Points (20 steps):
hs': 0.6 → 0.8, Ts': 0.6 → 0.4

$IAE$: PI Controller $\rightarrow$ 0.3062
NMP Controller $\rightarrow$ 0.2159

$ICA$: PI Controller $\rightarrow$ 8.0179
NMP Controller $\rightarrow$ 3.4046
Fig. 56  Comparison of Process Responses between Tuned PI Controller and NMP Controller (2):
\((\Delta F' = 0.02, \text{ Nr } = 2)\)

Ramp Change in Set Points (8 steps):
hs': 0.2 → 0.2, Ts': 0.3 → 0.7

IAE: PI Controller→ 0.5471
     NMP Controller→ 0.1024

ICA: PI Controller→ 8.2625
     NMP Controller→ 2.1930
Fig. 57  Comparison of Process Responses between
Tuned PI Controller and NMP Controller (3):
($\Delta F' = 0.02, N_r = 2$)

Ramp Set Point Change (8 steps):
hs’: 0.1 → 0.4, Ts’: 0.7 → 0.5

IAE: PI Controller → 1.0604
    NMP Controller → 0.1446

ICA: PI Controller → 7.1727
    NMP Controller → 3.4128
comparison of IAE and ICA between two controllers is also listed in the figures. In the simulations, PI controller parameters are set at the same values shown in Table 5 (for the step change in set points). Therefore, there might be a slight deviation from the best tuning. Parameters of the NMP controller are also set at the same values as the step change cases. In all figures, the process responses by the NMP controller exactly follow the ramp change in the set points. Whereas, the responses by PI controllers are one step behind the desired ramp change in the set points. The slight error in temperature during $t' = 18 \sim 25$ in Fig. 55 is caused by reaching the limit in one of the flow rates. During this period, the cold stream flow rate is forced to keep its maximum value of unity. If more steps, for example 30 steps instead of 20 steps, are allowed for the ramp change in the set points, the response by NMP controller will exactly follow the desired set point changes.

3. Results of the Regulator Problem
   a. Disturbance Rejection

   In this section, the closed-loop system behavior upon disturbance changes, referred as the regulator problem, is considered. In other words, the process is to be regulated at a constant set point for the entire time interval.

   For the mixing tank problem, possible disturbances may come from the inlet stream temperature or outlet flow rate.
In this study, the latter is chosen as the source of disturbance. The effect of a step change in the proportional constant $\alpha$ is examined. Recall that the outlet flow rate is given as $F_{\text{out}} = \alpha \sqrt{h}$ and $\alpha$ is determined by the characteristics of the outlet valve. Therefore, a step change in $\alpha$ can be considered as a change in valve opening. If the step change in $\alpha$ from 1.0 (original value) to 2.0 is caused by a valve opening action, the sum of the two inlet flow rates must be doubled to maintain the same liquid level. Also, the ratio of two inlet flow rates must remain constant to maintain the same temperature. Consequently, a change in $\alpha$ will have an effect not only on the liquid level but also on the temperature.

b. Response to the Disturbance Change

Results of the process responses to step changes in $\alpha$ are shown in Figs. 58 and 59. In these simulations, both set points of liquid level and temperature are kept constant at 0.6 and the disturbance is initiated at $t' = 3$. Fig. 58 shows the response to a step change in $\alpha$ from 1.0 to 1.5, and Fig. 59 corresponds to a change in $\alpha$ from 1.0 to 0.5.

For the case of $\alpha = 1.0 \rightarrow 1.5$ (Fig. 58), the NMP controller shows an excellent response to a step change in disturbance. In response to an increase of outlet flow rate, inlet flow rates increase immediately to maintain the liquid level and temperature, and a new steady-state point is
Fig. 58  Process Response to the Disturbance by NMP Controller: $\alpha=1.0 \to 1.5$
($hs'=0.6, Ts'=0.6$)

IAE = 0.1357, ICA = 1.2809
Fig. 59  Process Response to the Disturbance by NMP Controller (2): $\alpha = 1.0 \rightarrow 0.5$
(hs' = 0.6, Ts' = 0.6)

IAE = 0.2309, ICA = 14.141
established by $t' = 16$. However, for the case of $\alpha = 1.0 \rightarrow 0.5$ (Fig. 59), both liquid level and temperature show a cyclic behavior around a new steady-state value. This is caused by the miss selection of two inlet flow rates, which also show a cyclic behavior. If appropriate inlet flow rates are selected, the cyclic behavior will be avoided.

The network model that is used in the controller has no ability to consider the change in $\alpha$, because the network model was constructed without any information about the outlet flow rate. Therefore, the change in $\alpha$ produce a significant prediction error in process responses. Although the error correction part of the controller tries to compensate for this error, it will not work correctly for large error. (Although the results are not shown here, the error correction term also shows a cyclic behavior in the case of $\alpha = 1.0 \rightarrow 0.5$.)

To solve this problem, two approaches that take the controller action into account are examined (in addition to the maximum flow rate of 1.0 and the minimum flow rate of 0.0). The first approach is to add a new constraint to the controller action by limiting the maximum change of flow rate per step, $\Delta F'_{\text{max}}$. Such limitations are commonly used in industrial processes to avoid undesirable process upsets, and the limit values are usually decided by experience or from mechanical aspects. In this study, $\Delta F'_{\text{max}} = 0.2$ is used for both hot and cold stream flow rates.
The second approach is to include the controller action in the flow rate selection scheme. If the performance of the controller action is rated in terms of ICA, the following objective function can be obtained.

\[ f = (IAE) + \lambda \times (ICA) \quad (9) \]

where \( f \) : objective function
\( \lambda \) : controller action parameter

IAE and ICA in equation (9) are used in the recursive simulation to search for the best combination of two flow rates. Therefore, both values are calculated for the period of \( t' = k \sim k+Nr \) (\( Nr = 2 \) in this work), not for the period of \( t' = 0 \sim 30 \). The parameter \( \lambda \) is adjusted by the priority of IAE or ICA, and \( \lambda = 0.03 \), which shows later to be a reasonable choice, was selected for this study.

The process responses to the disturbance with the two approaches incorporated are shown in Figs. 60 and 61. For the case of \( \alpha = 1.0 \rightarrow 1.5 \) (Fig. 60), there is almost no difference in the process response or the controller action compared to the results with no constraint (Fig. 58). The process response by NMP controller is still excellent. On the other hand, there is a significant improvement for the case of \( \alpha = 1.0 \rightarrow 0.5 \) (Fig. 61) with the new approach. Although the deviation from the set point during \( t' = 3 \sim 10 \) is larger than
Fig. 60  Process Response to the Disturbance by NMP Controller with Constraints of Controller Action: $\alpha = 1.0 \rightarrow 1.5$
($h_s' = 0.6, T_s' = 0.6$)

$IAE = 0.1541$, $ICA = 1.1009$
Fig. 61 Process Response to the Disturbance by NMP Controller with Constraints of Controller Action (2) : $\alpha = 1.0 \rightarrow 0.5$  
($h_s' = 0.6$, $T_s' = 0.6$) 

$IAE = 0.2271$, $ICA = 2.2809$
that of the case with no constraint, there is no cyclic behavior after that period. In summary, by adding some constraints on the controller action, excellent disturbance rejection can be achieved.

The effect of $\lambda$ on the performance of controller is summarized in Fig. 62 in terms of ICA and ICA ($M = 30$). In all cases, $\Delta F_{\text{max}}$ is limited to 0.2. (Without any constraints (Fig. 59), the value of IAE and ICA become 0.2309 and 14.141, respectively.) As can be seen, $\lambda = 0.03$ is considered as the best choice for this disturbance change.

Process responses to the disturbance at another liquid level and temperature are shown in Figs. 63 and 64. In these figures, set points of liquid level and temperature are kept constant at 0.2 and 0.3, respectively. Both constraints of controller action are used in the simulations. In response to the increase (Fig. 63) or decrease (Fig. 64) of outlet flow rate, two inlet flow rates are immediately adjusted to maintain the liquid level and temperature. Furthermore, there is no cyclic behavior around a new steady-state point.

Comparison of the performance between the "best-tuned" PI controller and NMF controller (with constraints of controller action) is summarized in Table 7. The PI parameters used in the simulations are listed in Table 6, which are tuned in the same manner as that of the servo problems. As can be seen, the performance of PI controllers is better than that of a NMF controller. However, this is not
Fig. 62  Effect of Controller Action Parameter on IAE and ICA for the disturbance of $\alpha:1.0 \rightarrow 0.5$  ($hs'=0.6$, $Ts'=0.6$)
Fig. 63 Process Response to the Disturbance by NMP Controller with Constraint of Controller Action (1): $\alpha=1.0 \rightarrow 1.5$
($hs'=0.2$, $Ts'=0.3$)

$I_{AE} = 0.0714$, $I_{CA} = 0.9270$
Fig. 64 Process Response to the Disturbance by NMP Controller with Constraints of Controller Action (2): $a=1.0 \rightarrow 0.5$ ($hs'=0.2, Ts'=0.3$)

$IAE = 0.1589, ICA = 0.7070$
Table 6  Tununig Results of PI Controller
Parameters for the Disturbance Change

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>Kt</th>
<th>taul</th>
<th>Kt</th>
<th>taut</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha: 1.0 \rightarrow 0.5$</td>
<td>12</td>
<td>1.5</td>
<td>-27</td>
<td>0.8</td>
<td>0.1514</td>
</tr>
<tr>
<td>$\alpha: 1.0 \rightarrow 1.5$</td>
<td>4.9</td>
<td>1.9</td>
<td>-38</td>
<td>2.2</td>
<td>0.0505</td>
</tr>
</tbody>
</table>

(Set Point: hs' = 0.6, Ts' = 0.6)

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>Kt</th>
<th>taul</th>
<th>Kt</th>
<th>taut</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha: 1.0 \rightarrow 0.5$</td>
<td>5.2</td>
<td>1.6</td>
<td>-35</td>
<td>1.1</td>
<td>0.0408</td>
</tr>
<tr>
<td>$\alpha: 1.0 \rightarrow 1.5$</td>
<td>4.9</td>
<td>1.9</td>
<td>-38</td>
<td>2.2</td>
<td>0.0505</td>
</tr>
</tbody>
</table>

(Set Point: hs' = 0.2, Ts' = 0.3)

Table 7  Comparison of Process Responses to the Disturbance Change between Best-Tuned PI Controller and NMP Controller

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>IAE (PI)</th>
<th>IAE (NMP)</th>
<th>ICA (PI)</th>
<th>ICA (NMP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha: 1.0 \rightarrow 0.5$</td>
<td>0.0666</td>
<td>0.2271</td>
<td>1.4835</td>
<td>2.2809</td>
</tr>
<tr>
<td>$\alpha: 1.0 \rightarrow 1.5$</td>
<td>0.1514</td>
<td>0.1541</td>
<td>1.1395</td>
<td>1.1009</td>
</tr>
</tbody>
</table>

(Set Point: hs' = 0.6, Ts' = 0.6)

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>IAE (PI)</th>
<th>IAE (NMP)</th>
<th>ICA (PI)</th>
<th>ICA (NMP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha: 1.0 \rightarrow 0.5$</td>
<td>0.0505</td>
<td>0.1589</td>
<td>0.8928</td>
<td>0.7070</td>
</tr>
<tr>
<td>$\alpha: 1.0 \rightarrow 1.5$</td>
<td>0.0408</td>
<td>0.0714</td>
<td>1.1550</td>
<td>0.9270</td>
</tr>
</tbody>
</table>

(Set Point: hs' = 0.2, Ts' = 0.3)
a fair comparison because the PI controller parameters are obtained by the tuning only for the specified disturbance of outlet flow rate. The controller parameters are quite different from those tuned for set point changes (Table 5). It must be noted that these parameter settings cannot be commonly used for the disturbance of temperature or the servo problem.

4. Noise Effect on the Controller Performance

In process control, noise can arise from a number of sources: the measurement device, electrical equipment, or the process itself. In this section, the performance of a NMP controller is examined with the presence of white noise in the measurements.

Although the noise effect should be considered in the procedure of model construction, the same model constructed in chapter III, which is based on the noise-free data, is used here. This approach is acceptable because a BP network usually acts as a least-square filter to white noise [Klimasauskas, 1989]. Although white noise generates inconsistent mapping relationship, if enough of these relationships are included in the training data set, then the noisy portion cancels and the useful information will remain. A network can screen a considerable amount of white noise at the stage of model construction.

The noise considered in this study is illustrated in
Fig. 65 Effect of the Gaussian Noise ($\sigma = 2\%$) on the Steady-State Operation (Open-Loop: No Controller Action)
- Steady-State point: ($h_s' = 0.2$, $T_s' = 0.3$)
Fig. 65. Gaussian noise with standard deviation, $\sigma$, of 2% is simply added to the actual process response. The noisy signal is given to the controller as the measurement of the real process. If the process is operated under the open-loop steady-state condition, the noise has no effect on the actual process response. However, the noise causes an undesirable effect on the process response under closed-loop condition. This is illustrated in Figs. 66 and 67. Response to the step change in set points is considered, and tuned PI controllers are used here. As can be seen in the noise-free condition (Fig. 66), the process quickly responds to the step change in set points, and smoothly reaches the new set points with smooth controller actions. On the other hand, with only 2% of Gaussian noise, the actual process response as well as the feed-back signal (solid and broken line in lower panel of Fig. 67) show random fluctuations. If appropriate flow rates are chosen under the open-loop condition, no fluctuation will occur at least in the actual response. The controller action also shows random fluctuations with large frequency and amplitude. These flow rate fluctuations will not be acceptable for the final control devices such as control valves or pumps.

Process responses to the same set point change by the NMP controller are shown in Figs. 68 and 69. The former result is obtained with no constraint of controller actions, namely $\Delta F'_{\text{max}} = 1.0$ and $\lambda = 0.0$. Whereas, the latter result is
Fig. 66  Response to the Step Change in Set Points by "Best-Tuned" PI Controller with Noise Free Condition

\( (hs':0.1 \rightarrow 0.4, Ts':0.7 \rightarrow 0.5) \)

\( IAE = 1.7262, ICA = 2.6665 \)
Fig. 67  Response to the Step Change in Set Points by "Best-Tuned" PI Controller with the Gaussian Noise ($\sigma = 2\%$) 
($hs': 0.1 \rightarrow 0.4$, $Ts': 0.7 \rightarrow 0.5$)
IAE = 2.9229, ICA = 22.380
Fig. 68  Response to the Step Change in Set Points by NMP Controller
- with the Gaussian Noise ($\sigma = 2 \%$)
- No Constraints of Controller Action
  ($hs':0.1 \rightarrow 0.4$, $Ts':0.7 \rightarrow 0.5$)
IAE = 2.2735, ICA = 29.653
Fig. 69 Response to the Step Change of Set Points by NMP Controller with the Gaussian Noise ($\sigma = 2\%$) with Constraints of Controller Action ($hs':0.1 \rightarrow 0.4$, $Ts':0.7 \rightarrow 0.5$) $IAE = 3.0446$, $ICA = 8.7128$
obtained with the same constraints of controller actions as that used for the regulator problem, namely $\Delta F'_{\text{max}} = 0.2$ and $\lambda = 0.03$. The result with no constraint on the controller action (Fig. 68) shows similar result to that of the "best" tuned PI controller. Process responses as well as the controller actions show random fluctuations, which will not be acceptable for most of the actual situations. The result with $\Delta F'_{\text{max}} = 0.2$ and $\alpha = 0.03$ (Fig. 69) shows more favorable response and smoother controller action. Although IAE ($t' = 0$ - 30) increases from 2.274 to 3.045, ICA decreases considerably from 29.65 to 8.713.

In summary, the constraints of controller action used in the regulator problem is also effective on the noise problem. Needless to say, input data filter such as exponential or moving average filter can be used to improve the controller performance.

5. Discussion

In this chapter, a model predictive control system was developed based on the BP network model constructed in chapter III. This network model predictive (NMP) controller shows an excellent performance not only for the servo problem but also for the regulator problem. However, disturbances or noises sometimes cause undesirable cyclic behavior or fluctuations in process responses as well as controller actions. This behavior is similar to that of too fine tuned
PI controllers. In such situations, some constraints of controller action are useful to stabilize the system as shown in the simulations presented in section 3 and 4. If appropriate constraints are chosen, the performance of controller will not be deteriorated even in a noise-free or disturbance-free situation. Comparison of three controllers' performance for servo problem (no disturbance) in noise-free situation is shown in Table 8: the added constraints on the controller action does not affect much the value of IAE or ICA.

Although NMP controller proposed in this study shows excellent results, there are many possibilities to improve its performance. For example, It will be better in a noisy condition to use the moving average for the prediction error correction. Also, the flow rate selection by case studies can be replaced by a reverse mapping model. These studies may be considered in future efforts.
Table 8  Comparison of Process Responses to the Step Change in Set Points between Best-Tuned PI Controller and NMP Controller

Comparison of Integral Absolute Error (IAE)

<table>
<thead>
<tr>
<th>Set Point Change</th>
<th>PI Cont.</th>
<th>NMP Cont.</th>
<th>NMP Cont-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.9280</td>
<td>2.9877</td>
<td>3.3103</td>
</tr>
<tr>
<td>2</td>
<td>1.6612</td>
<td>1.5239</td>
<td>1.8519</td>
</tr>
<tr>
<td>3</td>
<td>1.7262</td>
<td>1.6910</td>
<td>2.4929</td>
</tr>
</tbody>
</table>

Comparison of Integral Controller Action (ICA)

<table>
<thead>
<tr>
<th>Set Point Change</th>
<th>PI Cont.</th>
<th>NMP Cont.</th>
<th>NMP Cont-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.3901</td>
<td>2.4646</td>
<td>2.3646</td>
</tr>
<tr>
<td>2</td>
<td>2.2462</td>
<td>3.1930</td>
<td>2.9730</td>
</tr>
<tr>
<td>3</td>
<td>2.6665</td>
<td>3.5128</td>
<td>3.3928</td>
</tr>
</tbody>
</table>

Set Point Change 1: hs'=0.6→0.8, Ts'=0.6→0.4
Set Point Change 2: hs'=0.2→0.2, Ts'=0.3→0.7
Set Point Change 3: hs'=0.1→0.4, Ts'=0.7→0.5

PI Cont. : Proportional-Integral Controller
NMP Cont. : Network Model Predictive Controller
NMP Cont-R: Network Model Predictive Controller with Constraint of Controller Action
V Conclusion

Modeling and control of non-linear MIMO system (a two-input, two-output mixing tank problem) were performed using a back-propagation network. The network model that was constructed from the pseudo-random binary data shows an excellent capability in both one-step ahead prediction and recursive prediction. The model construction process is very simple so that the time consuming nature of the modeling task can be minimized. Although the open-loop data was used in the current study to minimize the effort of data gathering or screening, the closed-loop data can be used for model construction [Broussard, 1990].

Based on the constructed model, a model predictive control system was developed. Although the controller is relatively simple in terms of concept and computation, it shows excellent control performances in both servo and regulator problem. Since the network model is non-linear, the controller is capable to cover the entire range of operating condition without local adjustment of parameters. Although the model prediction error sometimes causes a cyclic behavior of process responses, it can be stabilized with some constraints of controller action. The constraints of controller action are also effective on white noise.

There are still many issues that need to be addressed. Stability and convergence of the controller must be studied
in detail. The effects of time delay also have to be considered in many cases. However, the use of network for modeling and control of MIMO system appears to be very promising with its ability to treat non-linearity and process interactions.
REFERENCES


Birky, G. J. and T. McAvoy, "A Neural Net to Learn the Design of Distillation Controls", Department of Chemical Engineering, University of Maryland, Collegepark, MD 20742, 1988.


