Multistep Model Predictive Control Based on Artificial Neural Networks

ARTICLE *in* INDUSTRIAL & ENGINEERING CHEMISTRY RESEARCH · SEPTEMBER 2003

Impact Factor: 2.59 · DOI: 10.1021/ie020703k

7 AUTHORS, INCLUDING:

Shi-Shang Jang
National Tsing Hua University
73 PUBLICATIONS **634** CITATIONS
SEE PROFILE

Shyan-Shu Shieh
Chang Jung Christian University
32 PUBLICATIONS **375** CITATIONS
SEE PROFILE

Multistep Model Predictive Control Based on Artificial Neural Networks

Ji-Zheng Chu

Department of Automation, Beijing University of Chemical Technology, Beijing 100029, China

Po-Feng Tsai, Wen-Yen Tsai, Shi-Shang Jang,* Shyan-Shu Shieh,† Pin-Ho Lin,‡ and Shi-Jer Jiang§

Chemical Engineering Department, National Tsing-Hua University, Hsin-Chu 30043, Taiwan

This work studies the effect of different models on the performance of multistep model predictive control (MMPC) via simulation examples and bench- and pilot-scale experiments. The models used in the study are two common types of artificial neural networks (ANNs), namely, feedforward networks (FFNs) and external recurrent networks (ERNs). The steady-state offset of MMPC using FFN models is observed throughout simulation cases and experiments in case that prediction horizon is longer than the control horizon. This study further explains the FFN-induced offset phenomena mathematically. In the experimental part of this work, we compare the performances of MMPC using these two ANN models, conventional proportional−integral controllers and linear model predictive control in the dual-temperature control problems, which include a bench-scale ethanol and water distillation column and a pilot-scale *i*-butane and *n*-butane distillation column.

1. Introduction

The rapid development of computing technology makes it possible to pursue higher performance controllers. Many model-based control schemes have been proposed to incorporate a process model into a control system. In his review paper, Hussain[1] categorized them into three classes: predictive control, inverse model-based control, and adaptive control. Among them, predictive control is the most common scheme. Dutta and Rhinehart[2] also gave a brief comment on various model-based control schemes. The objective of this paper is to analyze the controllability of incorporating an output feedback model, such as feedforward neural networks (FFNs) and external recurrent networks (ERNs), into a multistep model predictive control (MMPC) system.

Linear model predictive control (LMPC) was almost simultaneously developed in both academic and industrial sectors.[3−5] At the very beginning, many pioneering investigators implemented horizon search instead of inverse model in the LMPC algorithm. The main reason is that the direct inverse of a "dynamic matrix" or a "heuristic model" is either not available or not necessary. In the case of horizon search, proper selection of the "prediction horizon" as well as the "input suppression horizon" may increase the stability of a control system.[6] Thus, in the area of MPC, it is essential to implement a prediction horizon into the future. We call a horizon-based MPC a multistep MPC (MMPC).

MPC demands a dynamic process model that is sufficiently accurate and can be high-speed computing, though the feedback mechanism of MPC tolerates some model mismatch to the plant. In the conventional LMPC, process outputs in the prediction horizon are expressed as a linear function of inputs and outputs, both of which are in the past time. Also, coefficients in the model function are determined from time series response data of the target process. Such a linear model is perfect for linear systems. It is a local approximation for nonlinear systems. For a highly nonlinear system, the region of acceptable approximation may become too small to be practically controllable.

In fact, developing a valid model for process dynamics is frequently the major work in the implementation of an advanced control system. Modeling cost normally accounts for over 75% of the expenditures in an advanced control project. Artificial neural networks (ANNs) as a process model for control purposes conceive the following superiority points as compared with other conventional modeling methods:[1,7,8]

(1) Models derived from first principles are usually difficult and/or costly to develop for processes that are not well understood or very complex. Additionally, to evaluate model parameters and to make models concise enough for online execution, assumptions and simplifications are inevitable and compromise model accuracy.

(2) ANNs provide a general approach for extracting process dynamics from input−output data only. Their learning ability makes them versatile and friendly for practical applications. In addition to their great power for approximating complex functionality, the compact form and high speed of information retrieval make ANNs very suitable for online use.

After reviewing 100 relevant papers on the subject of the application of ANNs to model-based control design, Hussain[1] concluded that real successful online applications are very rare.

Among all kinds of neural networks, FFNs and ERNs are the most commonly used in process control. Though MacMurray and Himmelblau[7] pointed out that ERNs

* To whom correspondence should be addressed. E-mail: ssjang@che.nthu.edu.tw.

† Present address: Department of Occupational Safety and Hygiene, Chang Jung University, Tainan, Taiwan.

‡ Present address: Department of Chemical Engineering, Nanya Institute of Technology, Taoyuan, Taiwan.

§ Present address: China Petroleum Corp., Chia-Yi, Taiwan.

10.1021/ie020703k CCC: $25.00
Published on Web 09/06/2003

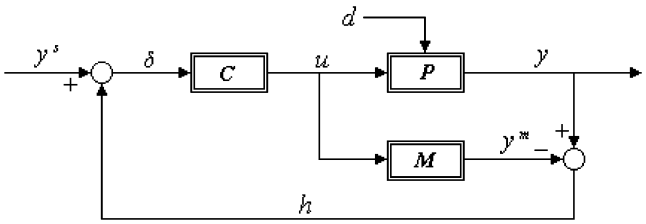

Figure 1. General architecture of MPC.

can forecast several steps ahead (prediction horizon) of a process and are more suitable for control purposes, we believe FFNs are not proper for MMPC, and this issue has not been thoroughly addressed before. Actually, quite a number of applications of FFNs in the framework of MPC have been reported.[1] In our experiments, MMPC using FFN models would result in offset when there is a long-term disturbance. On the contrary, MMPC using ERN models achieves zero offsets. In this paper, we further explain the reason causing such a difference between FFN and ERN models as incorporated into MMPC.

We first witnessed a nonzero offset existing in the MMPC using FFNs in simulation studies. To ensure our observations, we repeated the same control scheme in a bench-scale ethanol−water distillation column and a pilot-scale *i*-butane−*n*-butane distillation column. Both experiments involved the dual-temperature control problem. The major consideration in choosing two distillation columns as the process studied for the algorithm test is that such systems constitute a constrained, coupled, nonlinear, nonstationary process and have disparate dynamics.[2] They are complex enough to reveal superiority of a sophisticated scheme such as MMPC based upon ANN models. Another reason for the choice is that several reports[2,7,9,10] on the use of model-based control strategies used distillation columns as their application examples.

In the following context, the algorithm of MMPC is stated in section 2. In section 3, the FFNs and ERNs are introduced. In section 4, an analysis is presented to account for the offset of MMPC with FFN models. In section 5, simulation results are presented to verify the analysis on the offset problem. Section 6 is for the implementation of ANN-based MMPC and LMPC in a bench-scale distillation column of ethanol and water. The results demonstrate the advantage of nonlinear ANN model-based MMPC over LMPC. Section 7 is dedicated to the implementation of ANN-based MMPC and traditional proportional−integral (PI) control in a pilot-scale distillation column of *i*-butane and *n*-butane. The experimental results reveal superiority of ANN-based MMPC over PI control. The offset problem with MMPC using FFNs observed in our experiment is also included in sections 6 and 7. Conclusion remarks are given in section 8.

2. Principles of MMPC

MPC has a general architecture as shown in Figure 1, where double line blocks are used to emphasize the fact that the operators are nonlinear by following Frank's convention.[11,12] In this figure, vectors **u**, **y**, and **d** contain the manipulated variables (MVs), the controlled variables (CVs), and the disturbances, respectively, *P* is the process, *M* is a model of the process, and *C* is the controller, which is an optimizer minimizing the objective function below:

$$J = \sum_{i=1}^{M}\sum_{p=1}^{P}[y^s_{i,t+p} - (y^m_{i,t+p} + h_{i,t})]^2 + \sum_{j=1}^{N}\sum_{c=1}^{C} q_{j,c}\Delta u_{j,t+c}{}^2 \quad (1)$$

through searching for a set of increments for the MVs and subjected to constraints such as

$$P \geq C \quad (2)$$

$$u_{j,t+c} = u_{j,t+C} \qquad (c > C, j = 1, 2, ..., N) \quad (3)$$

$$|\Delta u_{j,t+c}| \leq \Delta u_{j,\max} \qquad (j = 1, 2, ..., N; c = 1, 2, ..., C) \quad (4)$$

$$u_{j,\min} \leq u_{j,t+c} \leq u_{j,\max} \qquad (j = 1, 2, ..., N) \quad (5)$$

$$y_{i,\min} \leq y^m_{i,t+p} \leq y_{i,\max} \qquad (i = 1, 2, ..., M; p = 1, 2, ..., P) \quad (6)$$

where M = number of controlled variables, N = number of manipulated variables, P = length of the prediction horizon (in time steps), C = length of the control horizon (in time steps), u_j = manipulated variable j, Δu_j = increment of manipulated variable j, defined as

$$\Delta u_{j,t+c} = u_{j,t+c} - u_{j,t+c-1} \qquad (j = 1, 2, ..., N; c = 1, 2, ..., C) \quad (7)$$

$\Delta u_{j,\max}$ = upper bound for the increment of manipulated variable j, $u_{j,\min}$ = lower bound of manipulated variable j, $u_{j,\max}$ = upper bound of manipulated variable j, y^m_i = predicted value for controlled variable i, as evaluated by model M, y^s_i = setpoint value for controlled variable i, t = step number of the current time, $q_{j,c}$ = weight for manipulated variable j in the cth increment in its control horizon, $h_{i,t}$ = difference between the measured and predicted values of controlled variable i, namely

$$h_{i,t} = y_{i,t} - y^m_{i,t} \quad (8)$$

and y_i = measured value for controlled variable i.

It is clear from Figure 1 and the objective function in eq 1 that MPC provides a feedforward action by including the term y^m, predicted effect of MVs u on the process through process model M, and h, predicted the effect of disturbance d on the process, as well as a feedback mechanism by minimizing the difference between the setpoint and the actual CVs. MPC possesses three attractive properties, namely, dual stability, perfect control, and zero offset under certain assumptions, as analyzed by Economou et al.[12] in the framework of the internal model control (IMC).

In this study, the Levenberg−Marquardt algorithm,[13] as recommended by Ramchandran and Rhinehart,[10] is adopted in searching for $\Delta u_{j,t+c}$ (j = 1, 2, ..., N; c = 1, 2, ..., C), which gives a minimum of J as defined in eq 1.

3. ANN Models

Figures 2 and 3 depict one FFN and one ERN, respectively. The difference between FFNs and ERNs is clear from these two figures. A FFN takes target values from outside the network (measured CVs) as part of its input and is therefore referred to as a *series*−

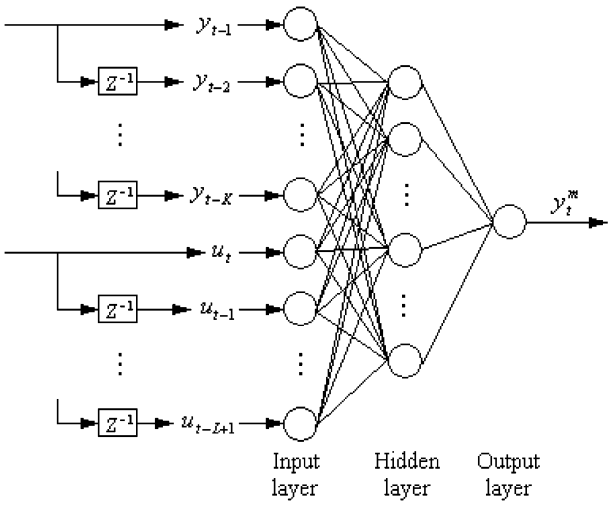

Figure 2. Structure of a FFN (there may be more than one hidden layer).

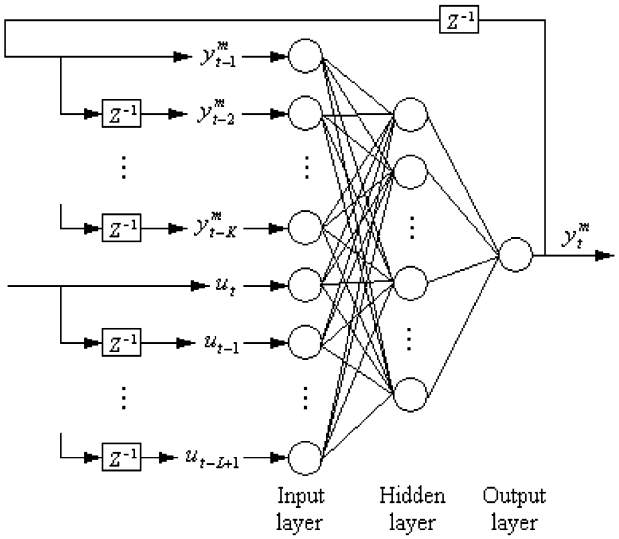

Figure 3. Structure of an ERN (there may be more than one hidden layer).

parallel identification model[7,14,15]

$$y_t = f(y_{t-1},...,y_{t-K};u_t,...,u_{t-L+1}) \tag{9}$$

where f is a nonlinear function of the network itself and K and L are the orders of the model with respect to the output and input. It should be noted that in prediction of the output variables two or more steps ahead of the current instant the predicted values of the FFN at previous instants are used as the network's input because no measured values are available ahead of the current instant. On the other hand, an ERN takes its own output (predicted CVs) as its partial input and is called a *parallel* identification model

$$y_t = f(y_{t-1}^m,...,y_{t-K}^m;u_t,...,u_{t-L+1}) \tag{10}$$

In all the experimental and simulation studies of this paper, neural networks are trained using the error back-propagation algorithm. Because there are no general rules for determining the structure (number of hidden layers, number of nodes in each hidden layer, etc.) of neural networks, structures of the FFN and ERN are fixed in a trial-and-error way in the training stage. To avoid overfitting, the model-validation technique used by Psichogios and Ungar[16] is adopted; namely, the performance of networks in predicting the testing data is checked after each epoch of learning, and the learning process is terminated if the prediction error on the testing data is increased by further training.

4. Offset of MMPC Using Series−Parallel Models

Ljung[14] has presented a thorough discussion of the advantages and disadvantages of parallel and series−parallel modeling approaches, and MacMurray and Himmelblau[7] have said that the parallel model is preferable in MMPC because model predictions more than one step ahead of the current instant may be required and feedback of the model output is therefore necessary. However, the inherent improperness of series−parallel models for MMPC is not fully recognized, and they are still adopted occasionally. Here, we present an analysis to account for the offset problem with MMPC based on series−parallel models.

For simplicity, we take a single-input single-output (SISO) system for analysis, and subscripts denoting variable indexes are also omitted. Without considering the penalty on the MV, MMPC in section 2 for a SISO process is reduced to solving the following system of equations in the sense of least-squares errors:

$$\begin{cases} y_{t+1}^s - (y_{t+1}^m + h_t) = 0 \\ y_{t+2}^s - (y_{t+2}^m + h_t) = 0 \\ ... \\ y_{t+P}^s - (y_{t+P}^m + h_t) = 0 \end{cases} \tag{11}$$

If our goal is to bring the system to a certain setpoint value y^s, namely,

$$y_{t+1}^s = ... = y_{t+P} = y^s \tag{12}$$

and let $E = y^s - h_t$, then the equation system of eq 11 can be rewritten as

$$y_{t+1}^m = y_{t+2}^m = ... = y_{t+P}^m = E \tag{13}$$

Therefore, the necessary and sufficient condition of zero offset is that there exist a set of values, $\{u_{t+1}, u_{t+2}, ..., u_{t+C}\}$, for the MV that fulfill the system of P equations as shown in eq 13 if the system is closed-loop stable.

For a parallel model shown in eq 10, its recurrent nature will eventually make all the P equations in eq 13 be fulfilled and ensure zero offset, although model mismatch may exist. The reason is that y_{t+1}^m, y_{t+2}^m, ..., and y_{t+P}^m will eventually and automatically become identical through recurrence (or iteration) if the system is stable. That means that all of the P equations of eq 13 are reduced to the single one:

$$f(y_\infty^m,...,y_\infty^m;u_\infty,...,u_\infty) = E \tag{14}$$

by approaching to the trivial solution $u_\infty = u_{t+1} = ... = u_{t+C}$.

On the other hand, equations of eq 13 are satisfied strongly conditionally if a series−parallel model is employed because y_{t+1}^m, y_{t+2}^m, ..., and y_{t+P}^m are never the same with each other, unless the model is perfect. This point is clear from the implementation of eq 13 for an

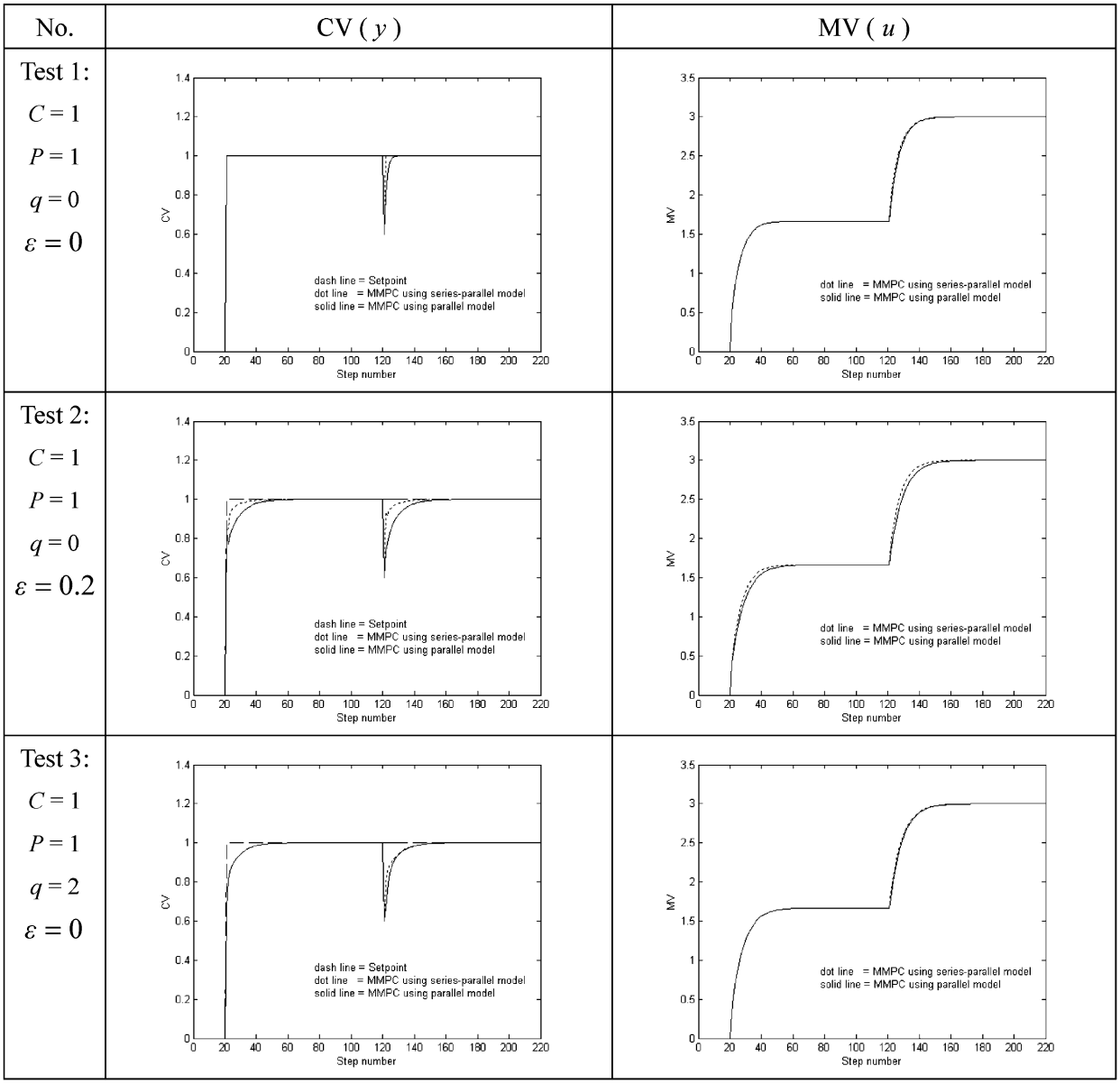

Figure 4. Performance of the linear system under MMPC ($C = 1$ and $P = 1$) with different MV penalties (q) and model mismatches (ϵ), subject to setpoint changes and disturbances.

example with prediction horizon $P = 5$, control horizon $C = 2$, and the model's order with respect to the CVs and MVs being $K = 3$ and $L = 3$.

$$\begin{cases} y^m_{t+1} = f(y_t, y_{t-1}, y_{t-2}; u_{t+1}, u_t, u_{t-1}) \\ y^m_{t+2} = f(y^m_{t+1}, y_t, y_{t-1}; u_{t+2}, u_{t+1}, u_t) \\ y^m_{t+3} = f(y^m_{t+2}, y^m_{t+1}, y_t; u_{t+2}, u_{t+2}, u_{t+1}) \\ y^m_{t+4} = f(y^m_{t+3}, y^m_{t+2}, y^m_{t+1}; u_{t+2}, u_{t+2}, u_{t+2}) \\ y^m_{t+5} = f(y^m_{t+4}, y^m_{t+3}, y^m_{t+2}; u_{t+2}, u_{t+2}, u_{t+2}) \end{cases} \quad (15)$$

For this example, we are to solve three unknown $\{u_t, u_{t+1}, u_{t+2}\}$ from five different equations, and the solution can only be in the sense of least-squares errors, equations of eq 13 cannot be satisfied strictly, and offset will surely happen, unless a perfect model is used. Because model mismatch is usually inevitable, offset can be generally anticipated for MMPC using series−parallel models, and such a MMPC will perform poorly in disturbance rejection. From the viewpoint of the least-squares error solution of eq 13, we can also expect larger offset for a greater difference between prediction and control horizons, $P - C$, and for greater penalty on MVs.

5. Two Simulation Examples

5.1. A Simple Linear System. To show the difference between parallel and series−parallel models as they are incorporated in MMPC, we first examine a simple linear process:

$$G_u = \frac{2 - 1.7z^{-1}}{1 - 0.5z^{-1}} \quad (16)$$

$$G_d = 1 \quad (17)$$

or in the discrete form

$$y_t = 0.5y_{t-1} + 2u_t - 1.7u_{t-1} + d \quad (18)$$

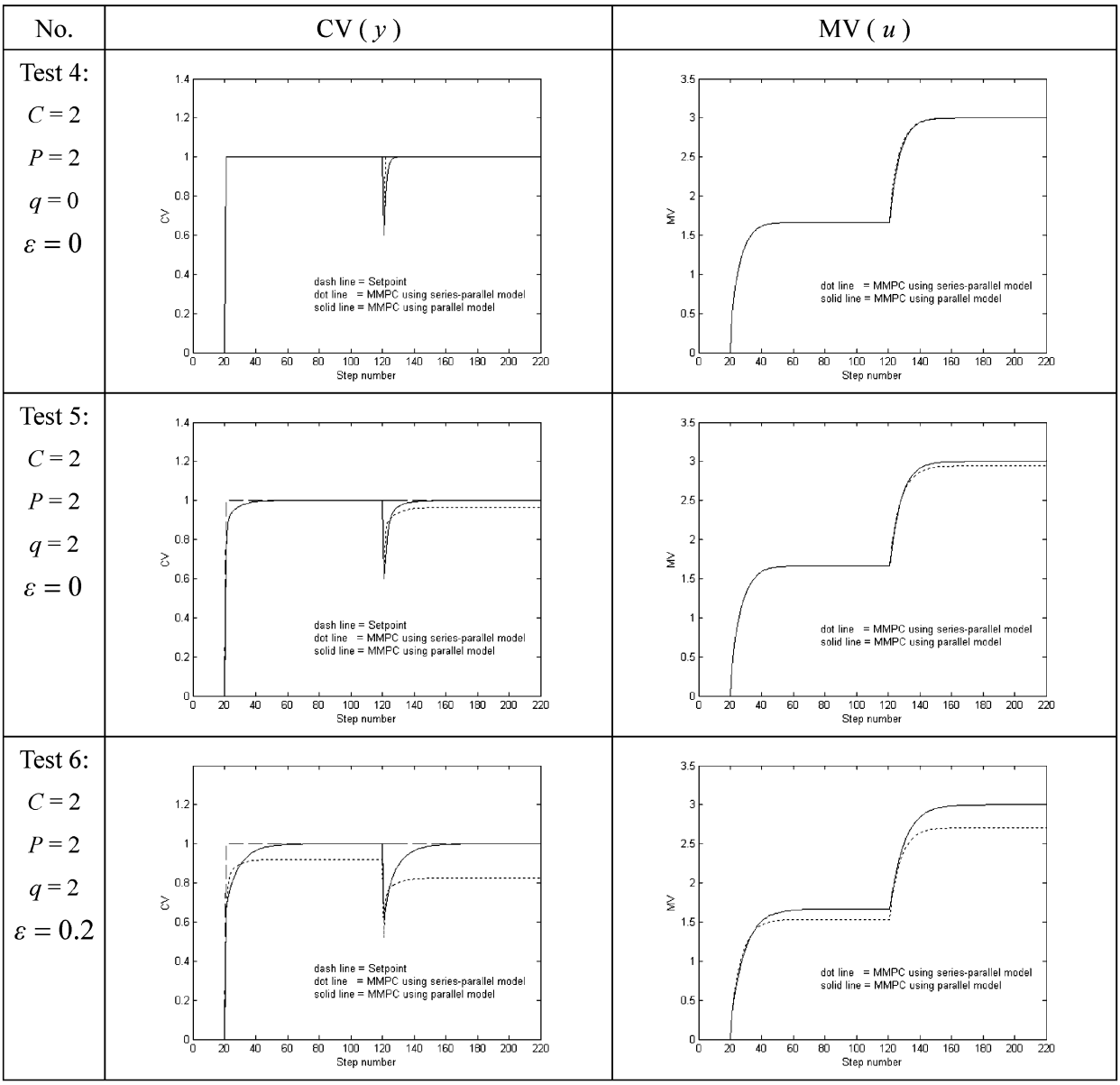

Figure 5. Performance of the linear system under MMPC ($C = 2$ and $P = 2$) with different MV penalties (q) and model mismatches (ϵ), subject to setpoint changes and disturbances.

with initial conditions

$$\begin{cases} y_t = y_{t-1} = 0 \\ u_t = u_{t-1} = 0 \\ d = 0 \end{cases} \tag{19}$$

The series−parallel and parallel models used are

$$y_t^m = (1+\epsilon)0.5y_{t-1} + (1+\epsilon)2.0u_t - 1.7u_{t-1} \tag{20}$$

$$y_t^m = (1+\epsilon)0.5y_{t-1}^m + (1+\epsilon)2.0u_t - 1.7u_{t-1} \tag{21}$$

where ϵ is used to modulate the model mismatch. If $\epsilon = 0$, eqs 20 and 21 are perfect models. Obviously, the orders of both parallel and series−parallel models are the same, namely, $K = 1$ and $L = 2$ with respect to the CVs and MVs, respectively.

Figures 4−6 show the behavior of this linear system under MMPC with different prediction and control horizons and using the two models of different mismatch degrees. To test the real nature of MMPC with different models and different parameter settings, relatively loose constraints are used here through inequalities (4) and (5) with $\Delta u_{\max} = 1$, $u_{\min} = 0$, and $u_{\max} = 10$. In all of the parallel runs, the setpoint is changed from 0 to 1 at the 20th step and disturbance $d = -0.4$ is introduced at the 120th step. Comparison between the test results in these figures justifies our conclusion that MMPC using a series−parallel model, eq 20 here, presents offset unless special conditions are fulfilled (see the discussion below), whereas MMPC using a parallel model, eq 21 here, is always free of offset.

From Figure 4, it is interesting to note that MMPC of single-step-ahead prediction and control horizons, whether using the series−parallel model or the parallel model, with penalty on MV increments or not, is free of offset despite model/plant mismatch. MMPC of equal prediction and control horizons and using the series−parallel model does not present offset if the model is perfect or no penalty on MV is used, which is clear from

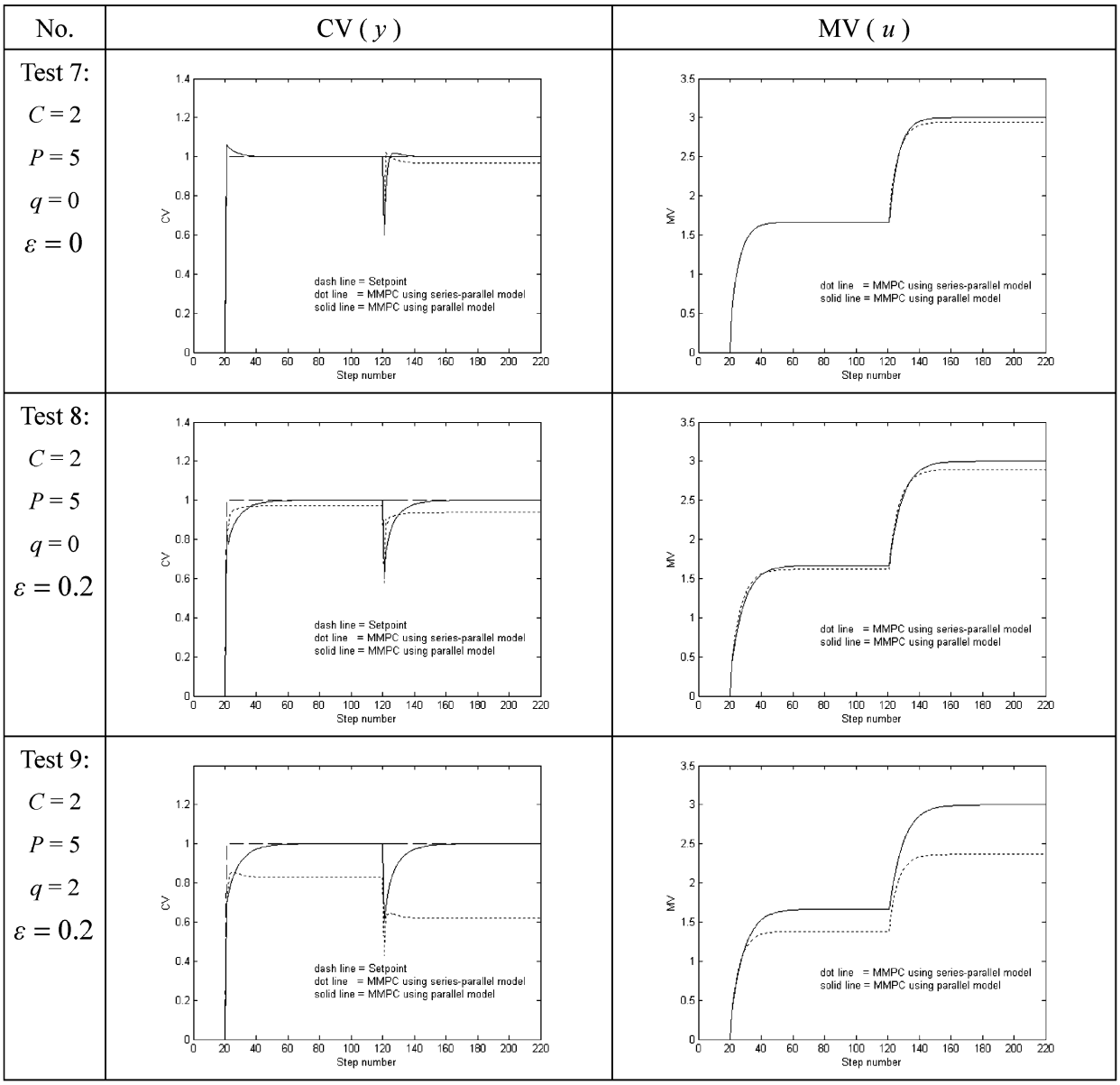

Figure 6. Performance of the linear system under MMPC ($C = 2$ and $P = 5$) with different MV penalties (q) and model mismatches (ϵ), subject to setpoint changes and disturbances.

Figure 5. As for the tests shown in Figure 6, where the prediction horizon is greater than the control horizon, MMPC based on the series−parallel model can be offset free only if the model is perfect. Penalties on MV will increase the amount of offset.

5.2. Continuously Stirred Tank Reactor (CSTR). To verify our previous analysis on the offset of MMPC with a FFN model, a kind of series−parallel model, we have performed a simulation of the CSTR once worked with by several researchers[12,16] and the author.[17] In this process a first-order nonisothermal reversible reaction A ↔ R occurs. The CV is the product concentration R_0, and the MV is the temperature of the inlet stream T_i. This process is described by three first-order nonlinear differential equations, and its setup details are referred to elsewhere.[12] With this process, Psichogios and Ungar[16] had their study on IMC and MMPC.

The same neural network as described by Psichogios and Ungar[16] is used in this study, and it has one input layer, two hidden layers, and one output layer. The input layer has six nodes, half of the nodes are for MV (T_i) at instants t, $t - 1$, and $t - 2$ and the other half for the CV (R_0) at instants $t - 1$, $t - 2$, and $t - 3$, with t being the current time instant. Each hidden layer has eight nodes, and a tangent sigmoid transfer function is used in each node. The output layer has one node with a pure linear transfer function, and the output is the CV at instant t. Functionally, the network can be expressed as

$$R_{0,t}^{m} = f(R_{0,t-1}, R_{0,t-2}, R_{0,t-3}; T_t, T_{t-1}, T_{t-2}) \qquad (22)$$

For easy comparison, we follow Psichogios and Ungar[16] strictly in building datasets for training and testing and in training and testing procedures.

In implemention of MMPC, the trained neural networks of eq 22 are used in two different ways, namely, as a series−parallel model and as a parallel model, and the same neural network is therefore distinguished according to running modes and is called the FFNand ERN models. The same constraints as those stated by Psichogios and Ungar[16] are used through inequalities

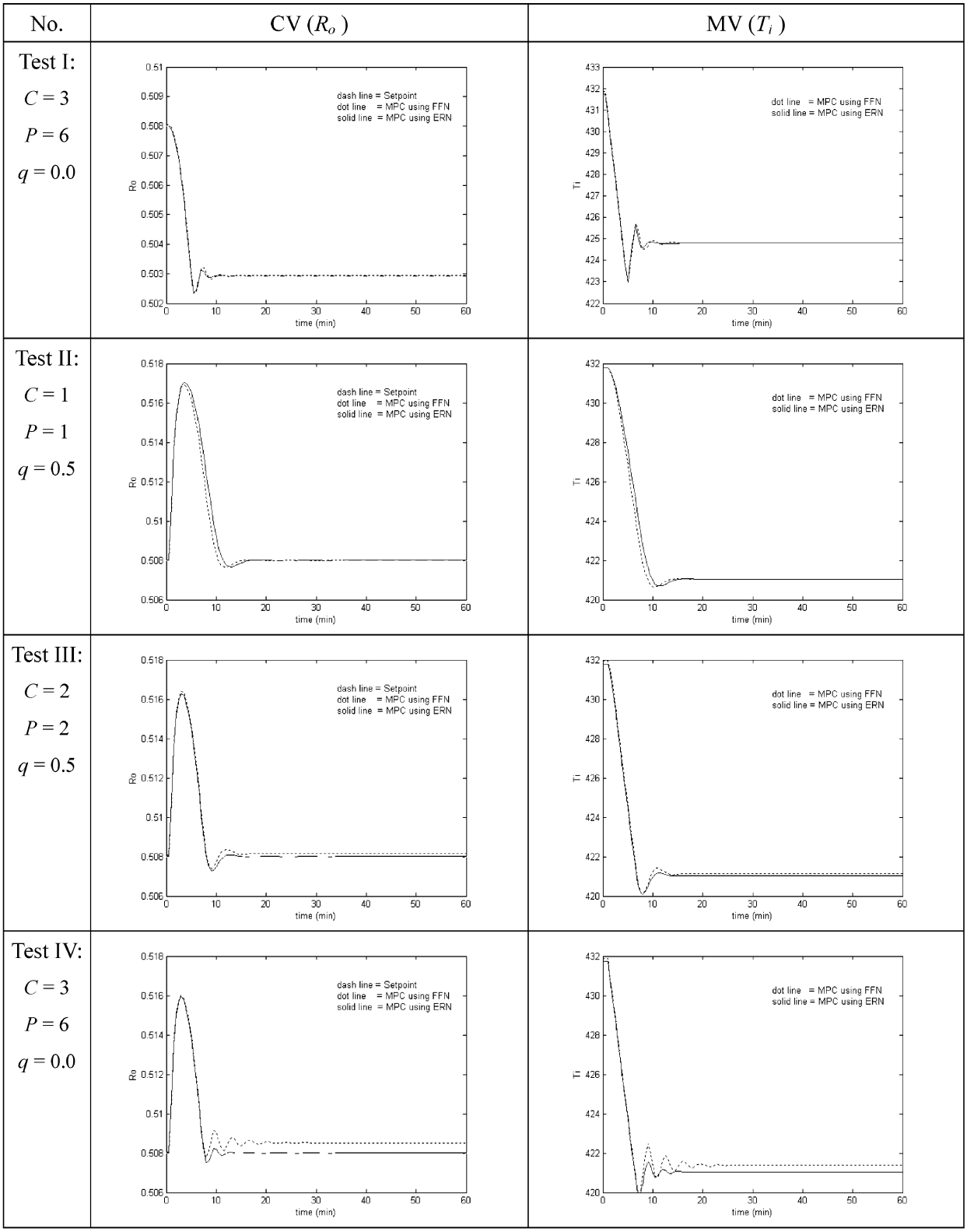

Figure 7. Performance of the CSTR under MMPC using FFN and ERN models, subjected to setpoint changes (test I) and disturbances (tests II−IV).

(4) and (5) with $\Delta u_{max} = 1$, $u_{min} = 420$, and $u_{max} = 435$. The CSTR under MMPC using FFN and ERN models are subjected to a 1% decrease in the setpoint value in test I and to a disturbance (+2% increase in the inlet feed concentration of reagent A) in tests II−IV, and the test results are shown in Figure 7. For a setpoint change in test I, both FFN and ERN model the target process almost perfectly, and MMPC with both models is virtually offset free, as anticipated. In test II, because only a one-step-ahead prediction horizon is used, MMPC using both FFN and ERN models achieves zero offset in the face of disturbance or model mismatch, just as argued in section 4. However, MMPC using the FFN model presents offset in tests III and IV, a clear contrast to its ERN counterpart, because disturbance means model mismatch and multistep prediction is employed.

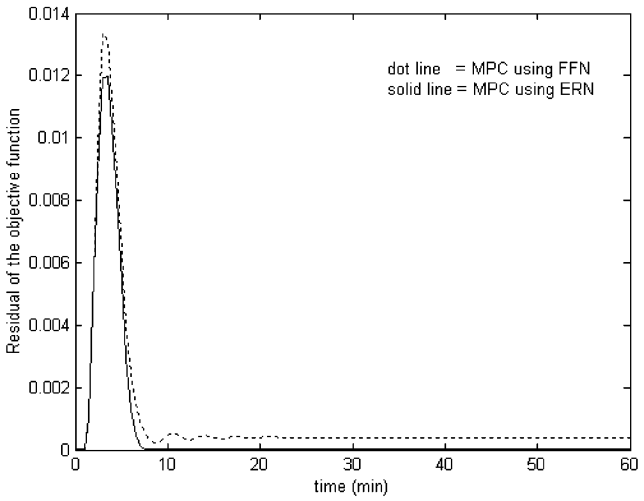

Figure 8. Residual of the objective function J of eq 1 for test IV with the CSTR under MMPC using FFN and ERN models.

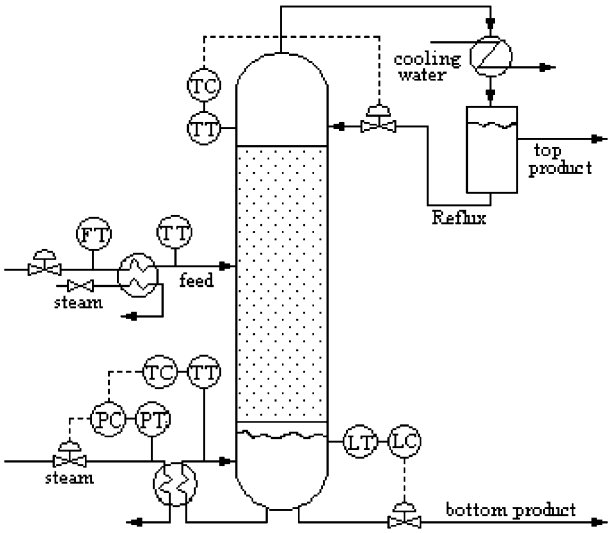

Figure 9. Configuration of the ethanol and water distillation column.

Table 1. Parameters of the Ethanol−Water Column

pressure	1 atm
inner diameter	0.1 m
packing height	1.1 m
stripping section height	0.9 m
packing porosity	0.968
feed rate	400 mL/min
feed composition	0.30 mole fraction of ethanol
reboiler holdup	0.0257 m^3
bottom holdup	0.0263 m^3
reflux drum holdup	0.0135 m^3

Additionally, the objective function (J) of eq 1, for test IV, is drawn in Figure 8, which says that offset presents wherever J is not small enough.

By the way, we also repeated the test of Psichogios and Ungar, where −2% disturbance in the inlet feed concentration of reagent A was exerted for 20 min, and got the same result as shown in their paper.[16] For such a "big" negative disturbance, MMPC with both FFN and ERN drove the system to its physical limit and exhibited offset during the disturbance. The system had an offset-free recovery to its setpoint after the disturbance was removed, as expected, because the ERN and FFN models were virtually perfect in that case.

Table 2. Optimum Parameters of PI Controllers for the Ethanol−Water Column

	Top Temperature		
gain	−42	integral time, s	70
	Bottom Temperature		
gain	13.5	inner loop gain	1.2
integral time, s	192.9		

Table 3. Structural Parameters of the FFNs and ERNs

no. of hidden layers	1
no. of hidden nodes	3
activation function for hidden nodes	hyperbolic tangent
no. of output nodes	1
transfer function for the output node	linear
entries in input data set 1	$u_1(t-1)$, ..., $u_1(t-18)$
entries in input data set 2	$u_2(t-1)$, ..., $u_2(t-18)$
entries in input data set 3	$y(t)$, ..., $y(t-6)$

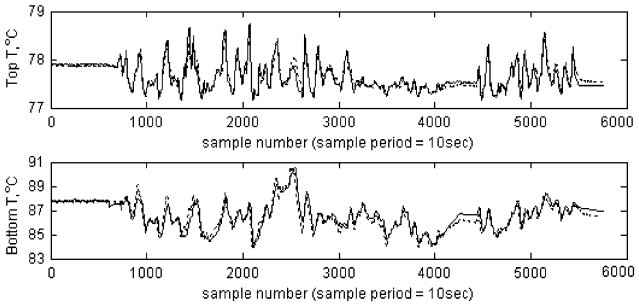

Figure 10. Testing data: top and bottom temperatures of the ethanol−water column, where solid line = measured and dotted line = predicted by the ERN.

6. Experiment on an Ethanol and Water Distillation Column

This test is carried out on a bench-scale distillation column for ethanol and water mixture, which is depicted in Figure 9 together with the PI controllers. Table 1 lists the structural and operational parameters of the column. It should be noted that the isotropic mixture (78.15 °C, 0.8943 mole fraction of ethanol, at 1 atm) is likely to form at the top of the column.

From measured step response curves, the four classic transfer functions for this column are derived as

$$G_{11} = \frac{y_1(s)}{u_1(s)} = \frac{-0.0194e^{-30s}}{68.75s + 1} \tag{23}$$

$$G_{12} = \frac{y_1(s)}{u_2(s)} = \frac{0.004e^{-166.25s}}{13.75s + 1} \tag{24}$$

$$G_{21} = \frac{y_2(s)}{u_1(s)} = \frac{-0.2607e^{-96.25s}}{518.75s + 1} \tag{25}$$

$$G_{22} = \frac{y_2(s)}{u_2(s)} = \frac{0.1427e^{-93.75s}}{410s + 1} \tag{26}$$

where u_1 = reflux valve opening (%), u_2 = reboiler heating steam pressure (kPa), y_1 = top temperature (°C), and y_2 = bottom temperature (°C).

The relative gain array (RGA) of Bristol[18] is calculated as

$$\begin{bmatrix} 1.6043 & -0.6043 \\ -0.6043 & 1.6043 \end{bmatrix} \tag{27}$$

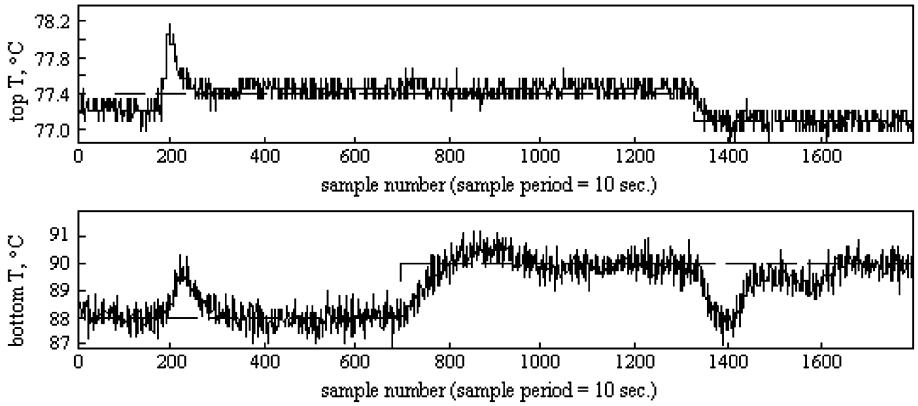

Figure 11. Transient process of the ethanol−water column under PI control, in response to step changes in the setpoints of the top and bottom temperatures, where solid line = measured and dashed line = setpoint.

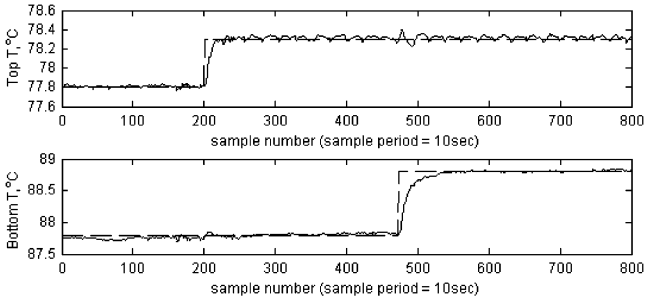

Figure 12. Top and bottom temperatures of the ethanol−water column subjected to two setpoint changes, under the MMPC (P = 20 and C = 1) using ERN models, where solid line = measured and dashed line = setpoint.

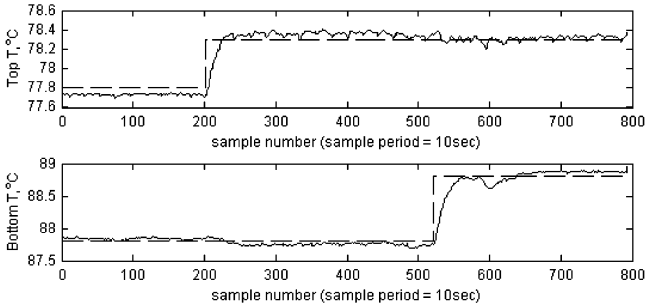

Figure 13. Top and bottom temperatures of the ethanol−water column subjected to two setpoint changes, under the MMPC (P = 20 and C = 1) using FFN models, where solid line = measured and dashed line = setpoint.

Optimal parameters for the two PI controllers are determined from eqs 22−26 by the BLT approach[19] and are listed in Table 2.

The test includes (1) process identification with ANNs, (2) PI control, (3) comparison of the performance of MMPC using FFN and ERN models, and (4) comparison of the performance of the ERN-based MMPC and LMPC. In implementation of the FFN- and ERN-based MMPC, no penalty on the MVs (u_1 and u_2) is used, namely, all q's in eq 1 are zero. The control horizon (C) for both MVs is 1, whereas the prediction horizon (P) is examined at different values in the test. Constraints on minimization include 35% $\le u_1 \le$ 95%, $|\Delta u_1| \le$ 6%, 55 kPa $\le u_2 \le$ 145 kPa, and $|u_2| \le$ 12 kPa. LMPC used here is the same as the direct matrix control (DMC), and its algorithm is referred to in a standard textbook of Marlin.[20] We use term LMPC instead of DMC here because DMC is a trademark and has special technical meanings.

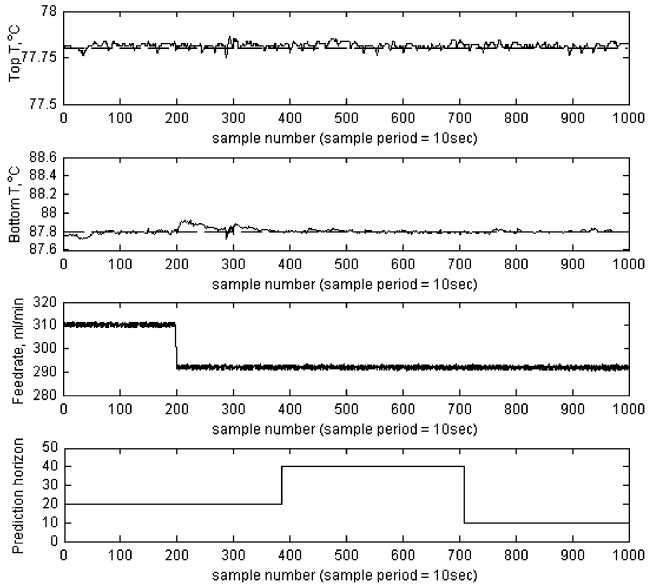

Figure 14. Top and bottom temperatures of the ethanol−water column subjected to a feed rate decrease from 311 to 290 mL/min at the 200th sample instant, under the MMPC (C = 1 and P = 20, 40, and 10) using ERN models, where solid line = measured and dashed line = setpoint.

6.1. Training of ANN Models. The training and testing datasets are obtained by recording the CVs, with the MVs varying randomly. For neural network models to better distinguish the effects of MVs on CVs, MVs are changed in three combination patterns, namely, random changes of both u_1 and u_2, and random changes of either u_1 or u_2, with one of them being constant. The structural parameters determined for the final two FFNs and two ERNs are listed in Table 3. Figure 10 presents the testing results from the two best-trained ERNs, and it is clear that the predicted curves by the two ERNs deviate observably from the measured ones. The testing results with the two FFNs are not shown here because the FFNs are much more accurate and the predicted curves almost coincide with the measured ones.

6.2. PI Control. Figure 11 shows the performance of the PI controllers, which are a record of the transient process of the CVs in response to step changes in the setpoints of both the top and the bottom temperatures. It was witnessed that, although the PI controllers were excellent when only one loop was closed (the experimental results of one closed loop are referred to else-

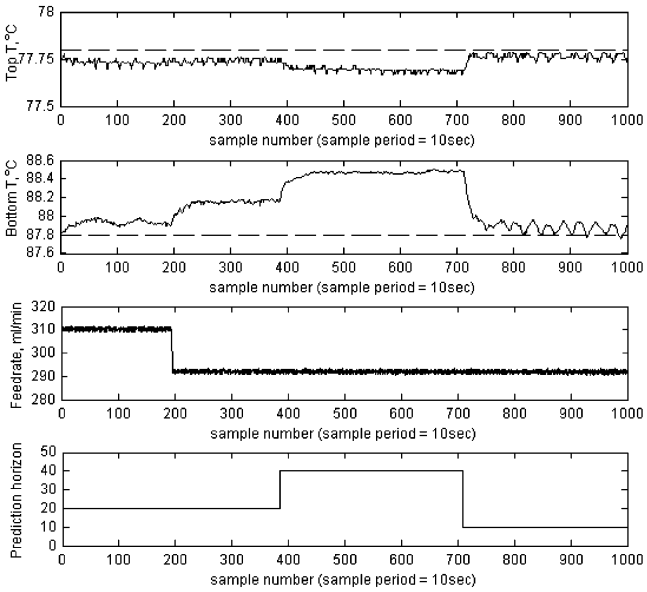

Figure 15. Top and bottom temperatures of the ethanol−water column subjected to a feed rate decrease from 311 to 290 mL/min at the 200th sample instant, under the MMPC ($C = 1$ and $P = 20$, 40, and 10) using FFN models, where solid line = measured and dashed line = setpoint.

where[21]), they were poor when both loops were closed because there was interaction between the top and bottom temperature loops, as is clearly indicated by the RGA in eq 27.

6.3. MMPC Using FFNs and ERNs. MMPC using the above FFNs and ERNs is used to control the ethanol and water column. Figure 12 depicts the transient process of the column controlled by MMPC using two ERNs as the models and subjected to setpoint changes. Figure 13 is the FFN counterpart of Figure 12. From the comparison of Figures 12 and 13, it is clear that MMPC using FFN models presents offset, a clear contrast to the offset-free behavior of MMPC using ERNs. The offset is not very gross in these setpoint changes because the FFNs are trained around the operating point and therefore have small mismatch. Figures 14 and 15 show the disturbance rejection performance of MMPC using both ERN and FFN models. Neural network models become more inaccurate with the presence of disturbance, and FFN-based MMPC presents a serious offset (Figure 15), whereas MMPC based on ERNs rejects the disturbance completely and has no offset (Figure 14). Just as noted in the first example of simulation in section 5, the offset with the FFN-based MMPC, as is clearly shown in Figure 15, becomes larger for a bigger prediction horizon (P), which is logical from our analysis in section 4.

6.4. Performance of ERN-Based MMPC versus LMPC. Figures 16−19 compare MMPC using ERN models and LMPC as they are used in the ethanol−water column subjected to setpoint changes. It is clear from the comparison between Figures 16 and 17 and between Figures 18 and 19 that the ERN-based MMPC is much superior to the LMPC in tracking setpoint

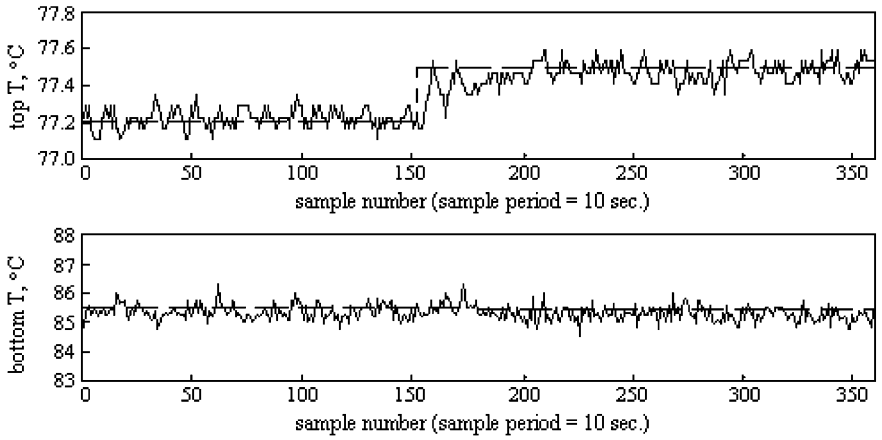

Figure 16. Top and bottom temperatures of the ethanol−water column subjected to a setpoint change in the top temperature, under the MMPC ($P = 12$ and $C = 1$) using ERN models, where solid line = measured and dashed line = setpoint.

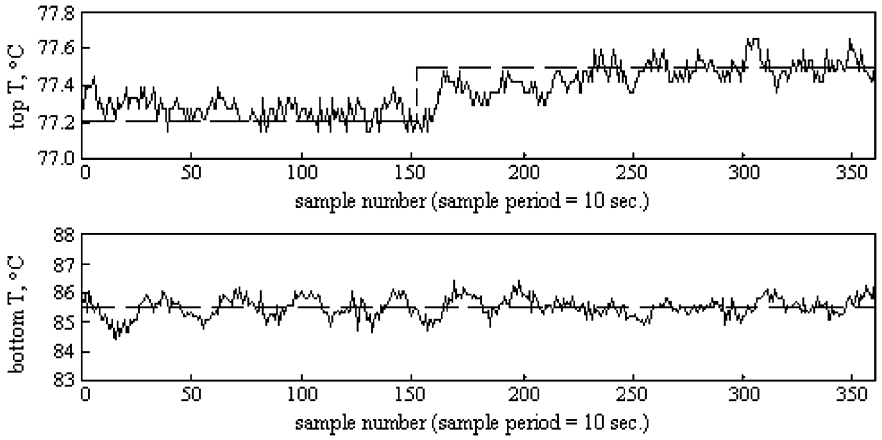

Figure 17. Top and bottom temperatures of the ethanol−water column subjected to a setpoint change in the top temperature, under the LMPC ($P = 12$ and $C = 1$), where solid line = measured and dashed line = setpoint.

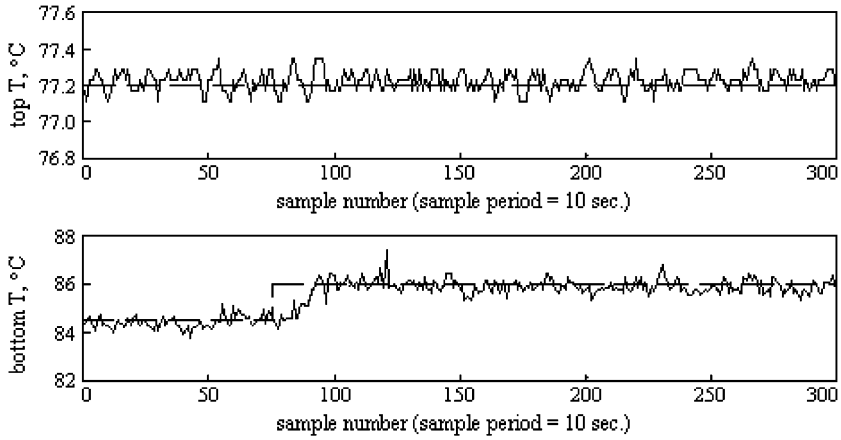

Figure 18. Top and bottom temperatures of the ethanol−water column subjected to a setpoint change in the bottom temperature, under the MMPC ($P = 12$ and $C = 1$) using ERN models, where solid line = measured and dashed line = setpoint.

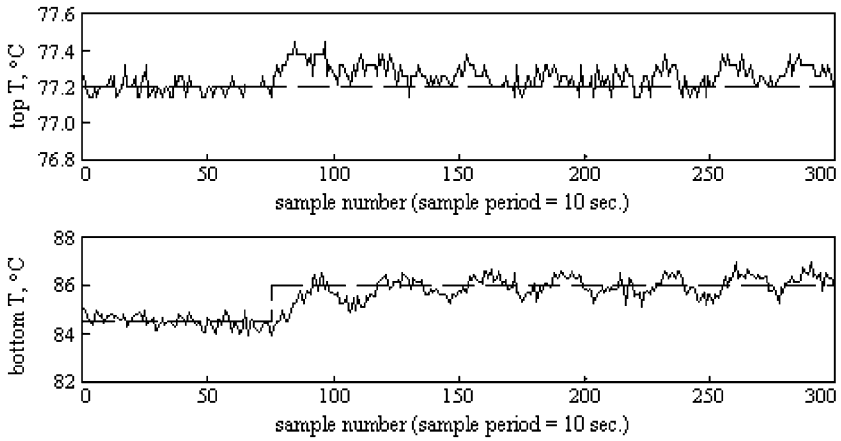

Figure 19. Top and bottom temperatures of the ethanol−water column subjected to a setpoint change in the bottom temperature, under the LMPC ($P = 12$ and $C = 1$), where solid line = measured and dashed line = setpoint.

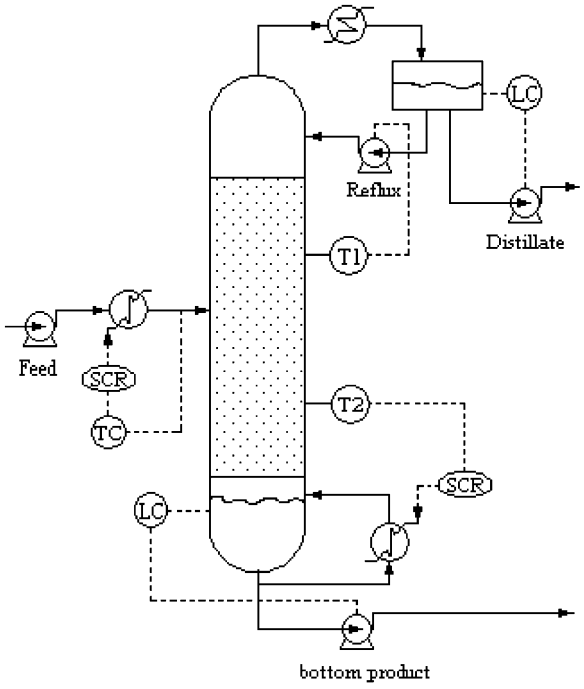

Figure 20. Configuration of the *i*-butane and *n*-butane distillation column.

changes for this ethanol−water distillation column where strong nonlinearity is present because the operation is in the neighborhood of an isotropic mixture at the top of the column.

Table 4. Parameters of the *i*-Butane and *n*-Butane Column

inner diameter	5 cm
packing height	180 cm
reflux drum	
inner diameter	5 cm
height	20 cm
feed composition	
i-butane	75 wt %
n-butane	25 wt %
feed flow rate	60 mL/min

Table 5. Optimized Parameters of PI Controllers for the *i*-Butane and *n*-Butane Column

Top Temperature at the 3rd Plate	
gain, °C/% reflux pump speed	−12
integral time, s	120
Bottom Temperature at 12th Plate	
gain, °C/% reboiler power	2.5
integral time, s	250

7. Experiment on an *i*-Butane and *n*-Butane Distillation Column

This test is performed on a pilot-scale column for *i*-butane and *n*-butane, as sketched in Figure 20. This pilot-scale column is packed with wire mesh packing, and its parameters are shown in Table 4. The reboiler is heated by electricity. Four metering pumps are included to control various flow rates. J-type thermocouples are adopted in temperature measurement. The separation capability of the column was about 18 theoretical plates. Temperatures at approximate loca-

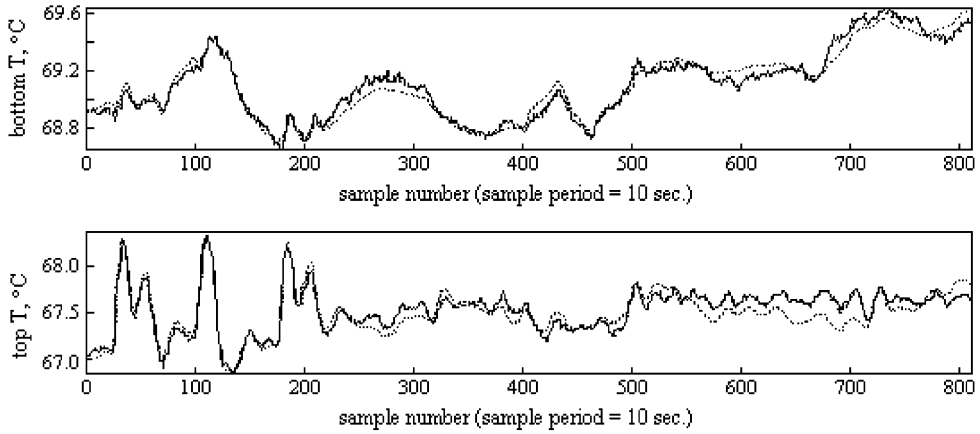

Figure 21. Testing data: top and bottom temperatures of the *i*-butane and *n*-butane column, where solid line = measured and dotted line = predicted by the ERN.

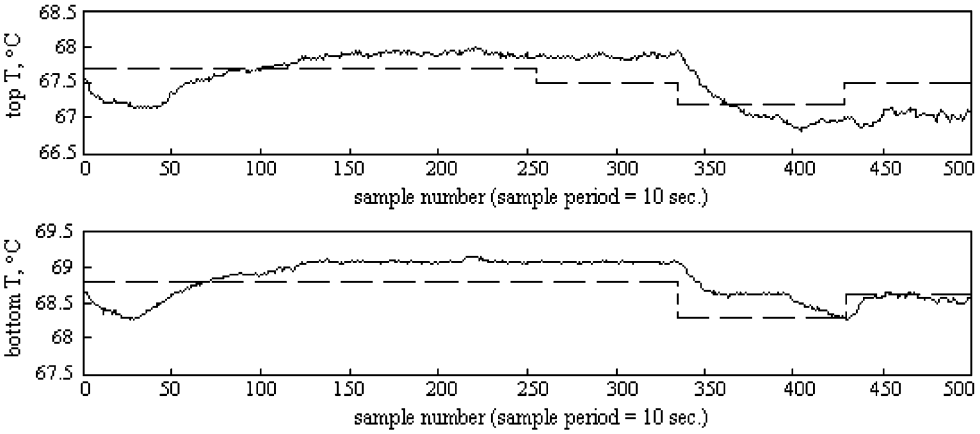

Figure 22. Transient process of the top and bottom temperatures of the *i*-butane and *n*-butane distillation column under the FFN-based MMPC, in response to step changes in the setpoints of the two CVs, where solid line = measured and dashed line = setpoint.

tions of the 3rd and the 12th theoretical plates, referred to as top and bottom temperatures, respectively, are the CVs. The control scheme with PI controllers is also shown in Figure 20 for this dual-temperature control problem.

For this pilot-scale column, the four classic transfer functions describing local behavior are derived from step response curves:

$$G_{11} = \frac{y_1(s)}{u_1(s)} = \frac{-0.14e^{-120s}}{1200s + 1} \tag{28}$$

$$G_{12} = \frac{y_1(s)}{u_2(s)} = \frac{0.6e^{-220s}}{840s + 1} \tag{29}$$

$$G_{21} = \frac{y_2(s)}{u_1(s)} = \frac{-0.04e^{-80s}}{700s + 1} \tag{30}$$

$$G_{22} = \frac{y_2(s)}{u_2(s)} = \frac{0.4e^{-70s}}{400s + 1} \tag{31}$$

where u_1 = reflux pump's speed, u_2 = reboiler heating power, y_1 = temperature at the height of the 3rd theoretical plate (top temperature), and y_2 = temperature at the height of the 12th theoretical plate (bottom temperature).

The RGA is evaluated as

$$\begin{bmatrix} 1.75 & -0.75 \\ -0.75 & 1.75 \end{bmatrix} \tag{32}$$

The BLT approach[20] is used to determine the optimum parameters for the two PI controllers from eqs 28−31, and the parameters are listed in Table 5.

The content of the test on this column is in three parts: (1) process identification with FFNs and ERNs, (2) control with MMPC using FFN models (the FFN-based MMPC), and (3) comparison of the MMPC using ERN models (the ERN-based MMPC) with the optimum PI controllers. In implemention of the FFN- and ERN-based MMPC, no penalty is used on the MVs, the control and prediction horizons are $C = 1$ and $P = 15$, and the following two constraints are included: $6\% \le u_1 \le 46\%$ and $23\% \le u_2 \le 29\%$ in the minimization.

7.1. Training of the FFN and ERN Models. Following the same procedure as that in subsection 6.1, training and testing datasets are established and two FFNs and two ERNs are trained to model the top and bottom temperatures, respectively. The structural parameters of the four networks are the same as those listed in Table 3, and Figure 21 compares the values predicted by the trained ERNs with the measured ones in the testing dataset for the two CVs, top and bottom temperatures. As in the ethanol−water column of the last section, the FFNs are much more accurate, and the predicted curves coincide with the measured ones.

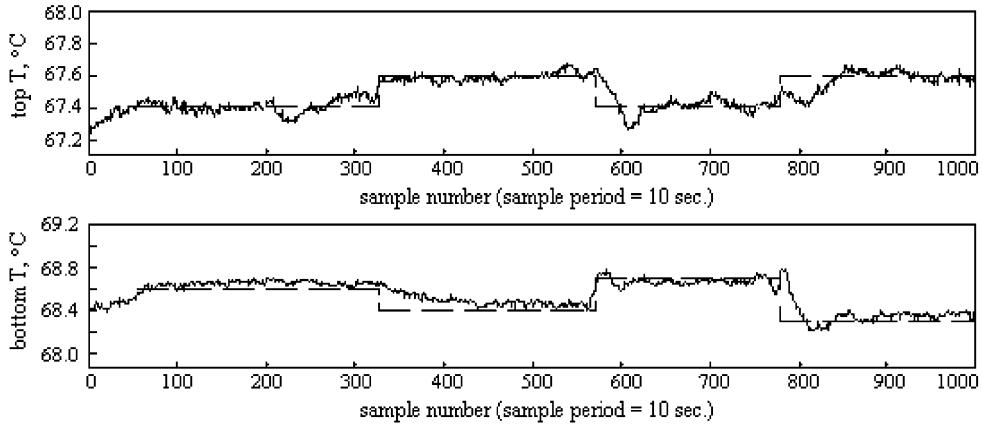

Figure 23. Top and bottom temperatures of the *i*-butane and *n*-butane column under ERN-based MMPC, in response to step changes in the setpoints of the CVs, where solid line = measured and dashed line = setpoint.

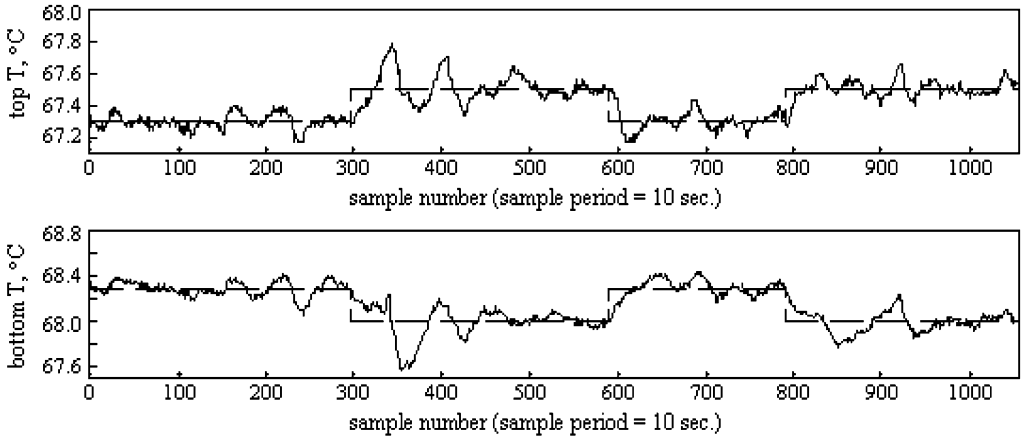

Figure 24. Top and bottom temperatures of the *i*-butane and *n*-butane column under PI control, in response to step changes in the setpoints of the CVs, where solid line = measured and dashed line = setpoint.

7.2. Performance of the FFN-Based MMPC. The FFN-based MMPC is tested on the *i*-butane and *n*-butane column by introducing a series of step changes in the setpoints of two CVs, namely, the top and bottom temperatures. From the result shown in Figure 22, it is clear that offsets are present with both CVs. With the analysis on the offset of MMPC using series−parallel models (here, the FFN models) in section 4, we are not surprised at such offsets because a long-term disturbance, afternoon sunshine on the reflux drum, existed on the experimental days.

7.3. Performance of the ERN-Based MMPC versus PI Control. The performance of the ERN-based MMPC is examined in this test. For comparison, PI controllers with parameters in Table 5, and as implemented in Figure 20, are also used to control the column in a parallel run. Figures 23 and 24 are the result of control with the ERN-based MMPC and the PI controllers. The transient process of the CVs in response to step changes in the setpoints of the top and bottom temperatures shows clear superiority of the ERN-based MMPC over conventional PI control in this dual-temperature control problem where interaction exists, as is indicated by the RGA in eq 32.

8. Conclusion

Through the analysis, simulation examples, and experiments on a bench-scale ethanol−water distillation column and on a pilot-scale column of *i*-butane and *n*-butane, we draw the following conclusions:

(1) Though FFNs perform better (or even much better) than ERNs in fitting training and testing data, which are measured at limited conditions, the presence of CVs in the input of FFNs, and all kinds of series−parallel models, makes such networks improper for MMPC. Because model mismatch is inevitable (especially when an unmeasured long-term disturbance is present), MMPC using FFNs produces offset. This defect of FFNs is analyzed and is demonstrated with simulation examples and with experimental data in this paper.

(2) The superiority of MMPC using nonlinear ANN models over conventional PI control and over LMPC is experimentally testified for the dual-temperature control problem in the two distillation columns. The advantage of MMPC comes from its capability of decoupling interactions between different control loops and from ANNs' capability of capturing the nonlinear dynamics of the processes.

(3) The success of ERNs as models in MMPC strongly supports the viewpoint of Rhinehart and co-workers,[2,10] who said that "it is gain prediction, more-so than state prediction, that makes model-based control effective." This conclusion is evident from the training results in Figures 10 and 21, where ERNs do have a correct correlation of the process gain, although errors exist in the predicted state.

Acknowledgment

The authors are thankful for the financial support for this work from National Science Council, Republic of China, through Grant NSC90-2622-E007-003.

Literature Cited

(1) Hussain, M. A. Review of the application of neural networks in chemical process control: simulation and online implementation. *Artif. Intell. Eng.* **1999**, *13*, 55−68.

(2) Dutta, P.; Rhinehart, R. R. Application of neural network control to distillation and an experimental comparison with other advanced controllers. *ISA Trans.* **1999**, *38*, 251−278.

(3) Cutler, C. R.; Ramaker, R. L. Dynamic matrix control: A computer control algorithm. AIChE 86th National Meeting, Houston, TX, 1979.

(4) Richalet, J.; Rault, A.; Rapon, J. Model Predictive Heuristic Control: Application to industrial process. *Automatica* **1978**, *14*, 413−428.

(5) Brosilow, C. B. The structure and design of Smith predictor from the viewpoint of inferential control. Proceedings of the Joint Automatic Control Conference, Denver, CO, June 17, 1979.

(6) Garcia, C. E.; Morari, M. Internal model control—1. A unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev.* **1982**, *21*, 308−323.

(7) MacMurray, J. C.; Himmelblau, D. M. Modeling and control of a packed distillation column using artificial neural networks. *Comput. Chem. Eng.* **1995**, *19*, 1077−1088.

(8) Bhat, N.; McAvoy, T. Use of neural nets for dynamic modeling and control of chemical process. *Comput. Chem. Eng.* **1990**, *14*, 573−583.

(9) Shaw, A. M.; Doyle, F. J., III; Schwaber, J. S. A dynamic neural network approach to nonlinear process modeling. *Comput. Chem. Eng.* **1997**, *21*, 371−385.

(10) Ramchandran, S.; Rhinehart, R. R. A very simple structure for neural network control of distillation. *J. Process Control* **1995**, *5*, 115−128.

(11) Frank, P. M. *Entwurf von Regelhreisen mit Vorgeschirebenem Berhalten*; G. Braun Verlag: Karlsruhe, Germany, 1974.

(12) Economou, C. G.; Morari, M.; Palsson, B. O. Internal model control. 5. Extension to nonlinear systems. *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 403−411.

(13) Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431−441.

(14) Ljung, L. *System Identification: Theory for the User*; Prentice-Hall: Eaglewood Cliffs, NJ, 1987.

(15) Narendra, K. S.; Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Network* **1990**, *1*, 4−27.

(16) Psichogios, D. C.; Ungar, L. H. Direct and indirect model based control using artificial neural networks. *Ind. Eng. Chem. Res.* **1991**, *30*, 2564−2573.

(17) Jang, S. S.; Joseph, B.; Muaki, H. Control of constrained multivariable nonlinear process using a two-phase approach. *Ind. Eng. Chem. Res.* **1987**, *26*, 2106−2114.

(18) Bristol, E. On a new measure of interaction for multivariable process control. *IEEE Trans. Autom. Control* **1966**, *AC-11*, 133−134.

(19) Luyben, W. L. *Process Modeling, Simulation, and Control for Chemical Engineers*, 2nd ed.; International Editions; McGraw-Hill: London, 1990.

(20) Marlin, T. E. *Process Control: Designing Processes and Control Systems for Dynamic Performance*; International Editions; McGraw-Hill, London, 1995.

(21) Tsai, W.-Y. Artificial Neural Network Model Predictive Control on Packed Distillation Columns. Master's Thesis, National Tsing-Hua University, Hsin-Chu, Taiwan, 2001.

Received for review September 11, 2002
Revised manuscript received July 2, 2003
Accepted August 5, 2003

IE020703K