CHAPTER 8

RECENT ADVANCES IN NEURAL NETWORK APPLICATIONS IN PROCESS CONTROL

U. Halici[1], **K. Leblebicioglu**[1], **C. Özgen**[1, 2], and **S. Tuncay**[1]
Computer Vision and Intelligent Systems Research Laboratory,
[1]Department of Electrical and Electronics Engineering,
[2]Department of Chemical Engineering
Middle East Technical University, 06531, Ankara, Turkey
{halici,kleb,cozgen}@metu.edu.tr, tuncay@ec.eee.metu.edu.tr

You must understand the process before you can control it and the simplest control system that will do the job is the best.

William L. Luyben (1990)

This chapter presents some novel applications of neural networks in process control. Four different approaches utilizing neural networks are presented as case studies of nonlinear chemical processes. It is concluded that the hybrid methods utilizing neural networks are very promising for the control of nonlinear and/or Multi-Input Multi-Output systems which can not be controlled successfully by conventional techniques.

1 Introduction

Classical control techniques such as Proportional Integral (PI) control or Proportional Integral Derivative (PID) control are successfully applied to the control of linear processes. Recently, linear Model Predictive Control (MPC) has also successfully been accomplished in the control of linear systems. However, about 90% of the chemical and biological processes are highly nonlinear and most of them are Multi-

© 2000 by CRC Press LLC

Input Multi-Output (MIMO). When the system is nonlinear and/or MIMO the above conventional techniques usually fail to control such systems. Nowadays, the systems used in industry require a high degree of autonomy and these techniques are not capable of achieving this [9].

The need to meet demanding control requirements in increasingly complex dynamical control systems under significant uncertainty makes the use of Neural Networks (NNs) in control systems very attractive. The main reasons behind this are their ability to learn to approximate functions and classify patterns and their potential for massively parallel hardware implementation. In other words, they are able to implement (both in software and hardware) many functions essential to controlling systems with a higher degree of autonomy.

Due to their ability to learn complex nonlinear functional relationships, neural networks (NNs) are utilized in control of nonlinear and/or MIMO processes. During the last decade, application of NNs in identification and control has been increased exponentially [24], [64].

The wide spread of application has been due to the following attractive features:

1. NNs have the ability to approximate arbitrary nonlinear functions;
2. They can be trained easily by using past data records from the system under study;
3. They are readily applicable to multivariable systems;
4. They do not require specification of structural relationship between input and output data.

This chapter contains four different approaches utilizing NNs for the control of nonlinear processes. Each of them is examined as a case study and tested on nonlinear chemical processes. While the first case study is utilizing NN in the usual way, the other three case studies are novel hybrid approaches.

In case study I, a simple NN control system having a neuro-estimator and a neuro-controller is developed to control a neutralization system, which shows a highly nonlinear characteristic. The system is tested for both set point tracking and disturbance rejection. The performance is compared with a conventional PID controller.

In case study II, a new structure, which incorporates NNs with the linear MPC to extend its capacity for adaptive control of nonlinear systems, is proposed. The developed controller is utilized in the control of a high-purity distillation column using an unsteady-state simulator. Its set point tracking and disturbance rejection capabilities are tested and compared with a linear MPC controller.

In case study III, an approach, which incorporates NNs with PI controllers, is presented. The main problem with the PI type controllers is the determination of proportional and integral constants for each operating (bias) point. The neural network is used to make an interpolation among the operating points of the process to be controlled and produce the related integral and proportional constants. The controller is tested in control of a binary batch distillation column.

In case study IV, a new method is proposed to control multi-input multi-output (MIMO) nonlinear systems optimally. An "optimal" rule-base is constructed, which is then learned and interpolated by a NN. This rule-based neuro-optimal controller is tested in the control of a steam-jacketed kettle.

The organization of the rest of this chapter is as follows: in the next two sections the concept of process control and use of NNs in process control are presented. The other next four sections are dedicated to case studies. The last section contains the remarks and future studies.

2 Process Control

In the development, design, and operation of process plants, the process engineers are involved with five basic concepts: state, equilibrium, conservation, rate, and control.

The identification of a system necessitates the definition thermo-dynamic **state** according to which all the properties of a system are fixed. Chemical, physical and biological systems can not be carried beyond the limits of thermodynamic **equilibrium**, which limits the possible ranges of chemical and physical conditions for the processes taking place in the system.

Conservation of mass, momentum and energy require that certain quantities be conserved in the process because of the mass, energy and momentum balances. The type and size specifications of process equipment of a system depend on the amounts of throughput and also on the **rates** at which physical, chemical and biological processes take place in the equipment. This concept is covered in the field of chemical and biological kinetics.

A process can be feasible both thermodynamically and kinetically but can still be inoperable because of poor operating performance. This can be a result of uncontrollability of the process or because of uneconomic conditions. Therefore, **control** of a system for a satisfactory operating performance, physically and economically, is as important for the design and operation of a process system as the concept of equilibrium and rate of processes [25].

Process control is the regulation of chemical, physical and biological processes to suppress the influence of external disturbances, to ensure the stability of the process and to optimize the performance of the process.

Some important features of process control can be listed as [25]:

- The study of process control necessitates first the study of **time-dependent** changes. The problems can not be formulated without a dynamic structure. The control of any process can only be studied by a detailed analysis of the unsteady-state behavior which can be obtained from the dynamic model of the process.
- Also, process control systems are **information-processing** systems. They receive information, digest it, act on it and generate information as signals.
- All process control systems are **integrated** systems of components, in which each component affects the overall performance of the system. Therefore, a global approach which considers the whole system and its environment as an entity is important.
- Most process control systems are **feedback** systems in which information generated by the system are processed again to regulate the behavior of the system.

- Finally, the **economical** concerns should be among the performance objectives of the process control system.

Process control systems in chemical, biological and physical process industries are characterized by constantly changing performance criteria, primarily because of the changes of the market demand. Also, these processes are highly nonlinear and can not be well modeled. Thus, the control has to be done to update the manipulated variables on-line to satisfy the changing performance criteria on the face of changing plant characteristics. Various control techniques based on different performance criteria and process representations are used to solve these problems.

During the operation of a plant, several requirements must be satisfied and can be considered as performance criteria. Some of them are listed below [58]:

1. Safety and environmental regulations,
2. Product specifications,
3. Operational constraints,
4. Economics.

These criteria must be translated to mathematical expressions in order to write a control law. They can further be classified as objectives (functions of variables to be optimized dynamically) and constraints (functions of variables to be kept within bounds).

Translation of performance criteria to mathematical expressions may require some assumptions. These assumptions are made not only to simplify the solution of the problem, but also to make the problem manageable for implementation in the existing hardware.

All controllers use a representation or a model of the process. Generally, in chemical and biological processes, models are nonlinear and also the model parameters are not well known. Thus there is always a mismatch between the model prediction and the actual process output. Additional reasons for the differences are due to changes in operating points and equipment.

Mismatches between a plant and its model result in unsatisfactory trading of the performance criteria. The tuning parameters can help the trade-off between the fast set point tracking and smooth manipulated variable response. It is always desirable to minimize the amount of on-line tuning by using a model of the process at the design stage that includes a description of the uncertainties.

Even if an uncertainty description is used, there is always a need for updating the model parameters on-line in an adaptive way. **Model Predictive Controllers,** MPC, are those controllers in which the control law is based on a process model [17]. MPC is a control scheme in which the controller determines a manipulated variable profile that optimizes some open-loop performance objective on a time interval extending from the current time to the current time plus a prediction horizon [15]. MPC is suitable for problems with a large number of manipulated and controlled variables, constraints imposed on both the manipulated and controlled variables, changing control objectives and/or equipment failure, and time delays. A model of the process is employed directly in the algorithm to predict the future process outputs.

Usually, in many process control problems, system models are not well defined; either they are missing or system parameters may vary with respect to time. NNs are convenient for obtaining input-output models of systems since they are able to mimic the behavior of the system after training them. Even if the NN model or identification may have mismatches with the plant at the beginning, it becomes better and better as the on-line training progresses. Furthermore, on-line training makes the NN model handle the time varying parameter changes in the plant, directly.

By training the NN to learn the “inverse model” of a plant it can be used as a “controller” for the plant. Also, NN controllers can be used in MPC structures both as estimator and/or controller parts.

Since chemical and biological processes are usually very complex, instead of using NN alone in control of these processes, using them together with conventional approaches such as PI or PID control techniques or recent techniques such as rule based expert systems or fuzzy logic, in a hybrid manner, improves the performance of the overall controller.

3 Use of Neural Networks in Control

In control systems applications, feedforward multi-layer NNs with supervised training are the most commonly used. A major property of these networks is that they are able to generate input-output maps that can approximate any function with a desired accuracy. NNs have been used in control systems mainly for system identification and control.

In **system identification**, to model the input-output behavior of a dynamical system, the network is trained using input-output data and network weights are adjusted usually using the backpropagation algorithm. The only assumption is that the nonlinear static map generated by the network can adequately represent the system's dynamical behavior in the ranges of interest for a particular application. NN should be provided information about the history of the system: previous inputs and outputs. How much information is required depends on the desired accuracy and the particular application.

When a multi-layer network is trained as a **controller**, either as an open-loop or closed loop, most of the issues are similar to the identification case. The basic difference is that the desired output of network, that is the appropriate control input to be fed to the plant, is not available but has to be induced from the known desired plant output. In order to achieve this, one uses either approximations based on a mathematical model of the plant (if available), or a NN model of the dynamics of the plant, or, even, of the dynamics of the inverse of the plant. NNs can be combined to both identify and control the plant, thus forming an adaptive control structure.

We will now introduce some basic ways in which NN training data can be obtained in tasks relevant to control [37]:

- **Copying from an existing controller**: If there is a controller capable of controlling a plant, then the information required to train a neural network can be obtained from it. The NN learns to copy the existing controller. One reason for copying an existing controller is that it may be a device that is impractical to use, such as a human expert. In some cases, only some finite input-output command pairs of a desired controller are known. Then an NN can be trained to emulate the desired controller by interpolating these input-output command pairs.

- **System Identification:** In the identification case, training data can be obtained by observing the input-output behavior of a plant. In more complex cases, the input to the model may consist of various delayed values of plant inputs and the network model may be a recursive one.

- **Identification of System Inverse:** In this scheme, input to the network is the output of the plant and the target output of the network is the plant input. Once the plant inverse NN is obtained, it is fed by the desired plant output and its output is then the desired control input to the plant. The major problem with inverse identification is that the plant's inverse is not always well defined.

- **Model Predictive Controller:** First a multi-layer network is trained to identify the plant's forward model, then another NN, i.e., the controller, uses the identifier as the plant's estimator in an MPC structure. This scheme has an advantage of being an adaptive controller, but it necessitates the computation of the Jacobian of the identifier NN.

There are many advanced networks for more complex system identification of control problems. The reader is referred to [3], [4], [37] for a list of references.

The system identification part is the backbone of almost all neurocontroller architectures so we will discuss this concept in more detail for SISO plants suggested in [38]. These models have been chosen for their generality as well as for their analytical tractability. The models of the four classes of plants can be described by the following nonlinear difference equations:

Model I:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) \quad + \mathrm{g}(u(k),..,u(k-m+1)) \tag{1}$$

Model II:

$$y_p(k+1) = \mathrm{f}(y_p(k),..,y_p(k-n+1)) + \sum_{i=0}^{m-1} \beta_i u(k-i) \qquad (2)$$

Model III:

$$y_p(k+1) = \mathrm{f}(y_p(k),..,y_p(k-n+1)) + \mathrm{g}(u(k),..,u(k-m+1)) \qquad (3)$$

Model IV:

$$y_p(k+1) = \mathrm{f}(y_p(k),..,y_p(k-n+1); u(k),..,u(k-m+1)) \qquad (4)$$

where $(u(k), y_p(k))$ represents the input-output pair of the plant at time k and f:$R^n \rightarrow R$, g:$R^m \rightarrow R$ are assumed to be differentiable functions of their arguments. It is further assumed that f and g can be approximated to any desired degree of accuracy on compact sets by multilayer NNs. Due to this assumption, any plant can be represented by a generalized NN model.

To identify a plant, an identification model is chosen based on prior information concerning the class to which it belongs. For example assuming that the plant has a structure described by model III, we have two types of identifiers:

1. Parallel model: In this case, the structure of the identifier is identical to that of the plant with f and g replaced by the corresponding NNs, N_1 and N_2 respectively. This model is described by the equation

$$\hat{y}_p(k+1) = N_1(\hat{y}_p(k),..,\hat{y}_p(k-n+1)) + N_2(u(k),..,u(k-m+1)) \qquad (5)$$

2. Serial-parallel model: The model is described by the equation:

$$\hat{y}_p(k+1) = N_1(\hat{y}_p(k),..,\hat{y}_p(k-n+1)) + N_2(u(k),..,u(k-m+1)) \qquad (6)$$

When a plant is identified, a proper controller can be designed based on the identification model. When external disturbances and/or noise are not present in the system, it is reasonable to adjust the control and identification parameters simultaneously. However, when noise and/or disturbances are present, controller parameter updating should be carried out over a slower time scale to ensure robustness.

A number of applications of NNs to process control problems have been reported. A widely studied application involves a nonlinear-model predictive controller [5], [22], [23], [48], [49], [51], [65]. Piovoso *et al.* have compared NN to other modeling approaches for IMC, global linearization and generic model control and they have found that NNs give excellent performance in the case of severe process/model mismatch [46]. Seborg and co-workers have used radial basis function NN for nonlinear control and they have applied their approaches to simulated systems as well as an actual pH process [39], [47], [48], [49], [53]. They have found the NN based controllers to be superior to other methods in terms of their ease of design and their robustness. NNs are often viewed as black box estimators, where there is no attempt to interpret the model structure [61]. NNs have been used in nonlinear process identification [11], in IMC [13], [39], in adaptive control [7], [16], in tuning conventional PID controllers [63], and in both modeling and control of nonlinear systems [16]. The model adaptation of NN based nonlinear MPC has been studied in [29] and [30].

Narendra *et al.* explained how neural networks can be effectively used for identification and control of nonlinear dynamic systems, where an NN is trained by a backpropagation algorithm for adjustment of parameters [38]. They studied multilayer and recurrent NN in a unified configuration for modeling. Simulation studies on low order nonlinear dynamic systems showed that such modeling and control schemes are practically feasible and they proposed that the same methods can also be used successfully for the identification and control of multivariable systems of higher dimensions.

Bhat *et al.* discussed the use of multilayer NN trained by back-propagation algorithm for dynamic modeling and control of chemical processes [6]. They proposed two approaches for modeling and control of nonlinear systems. The first approach utilizes a trained NN model of the system in a model based control work frame and the second approach utilizes an inverse model of the plant extracted using NN in the internal model control structure. They implemented the first approach on a CSTR where pH is the controlled variable. Their results showed that NN is better in representing the nonlinear characteristics of the CSTR than classical convolution model and, also, the controller performance is superior to convolution model-based control.

Willis *et al.* discussed NN models from the process engineering point of view and explained some approaches for use of NN in modeling and control applications [65]. They considered some industrial applications whereby an NN is trained to characterize the behavior of the systems, namely industrial, continuous and fed-batch fermenters, and a commercial scale, industrial, high purity distillation column. They pointed out that NNs exhibit potential as soft sensors. They also explained a methodology for use of NN models in MPC structure to control nonlinear systems. The results of their simulation studies on a highly nonlinear exothermic reactor have indicated that although there are many questions to be answered about NN for optimum utilization (e.g., topology, training strategy, modeling strategy, etc.), NN are a promising and valuable tool for alleviating many current process engineering problems.

Nguyen *et al.* have presented a scheme for use of NNs to solve highly nonlinear control problems. In their scheme, an emulator, which is a multilayered NN, learns to identify the dynamic characteristics of the system [41]. The controller, which is another multi-layered NN, learns the control of the emulator. Then this controller is used in order to control the actual dynamic system. The learning process of the emulator and the controller continues during the control operation so as to improve the controller performance and to make an adaptive control.

Tan described a hybrid control scheme for set point change problems for nonlinear systems [59]. The essence of the scheme is to divide the control into two stages, namely, a coarse control stage and a fine control stage, and use different controllers to accomplish a specific control action at each stage. For the coarse stage, a modified multilayer NN with backpropagation training algorithm is used, which drives the system output into a predefined neighborhood of the set point. The controller then switches to the fine control stage at which a linearization of the system model is identified around the current set point, and is controlled with an appropriated PID controller. Simulation results have shown that there are some difficulties that can be faced in the development of such a hybrid control scheme, such as the criteria for the controller switching stages, and the possibility of abrupt changes in control input in the controller switching phase. The applicability of this control scheme to nonlinear control problems is discussed.

Dreager *et al.* have proposed a new nonlinear MPC algorithm for control of nonlinear systems [13]. For the prediction step, their algorithm utilizes a NN model for a nonlinear plant. They have applied this algorithm to a pH system control and also a level control system. They have compared the performance of their nonlinear MPC algorithm with that of a conventional PI controller on these two systems. Results have indicated that the proposed controller outperforms with respect to the PI controller.

Hamburg *et al.* examined various methods, especially NN, with respect to their use to detect nuclear material diversions, considering speed and accuracy [19]. The NN technique is enhanced with the use of a computer simulation program for creating the training data set. This simulation approach provided the opportunity of including outliers of various types in a data set for training the NN because an actual process data set used for training possibly might not have outliers. They compared the methods on their ability to identify outliers and reduce false alarms. These methods were tested on data sets of nuclear material balances with known removals. The results obtained by the NNs were quite encouraging.

Sablani *et al.* used NNs to predict the overall heat transfer coefficient and the fluid to particle heat transfer coefficient, associated with liquid particle mixtures, in cans subjected to end-over-end rotation [50]. These heat transfer coefficients were also predicted by means of a dimensionless correlation method on the same data set. The results showed that the predictive performance of the NN was far superior to that of dimensionless correlations.

Noriega and Wang presented a direct adaptive NN control strategy for unknown nonlinear systems [43]. They described the system under consideration as an unknown NARMA model, and a feedforward NN was used to learn the system. Taking NN as a neuro model of the system, control signals were directly obtained by minimizing either the instant difference or the cumulative differences between a set point and the output of the neuro model. They applied the method in flow rate control and successful results were obtained.

Since 1990, there are too many academic papers on NN controllers and applications in process control, though there are a few real applications. Nowadays advantages and disadvantages of NNs have been well understood. New studies, such as hybrid structures, are constructed in which NNs can appear in several stages emphasizing their advantages.

In the following sections four case studies are presented to show the applications of NNs in conjunction with other techniques for control of complex processes.

4 Case Study I: pH Control in Neutralization System

pH control problem is very important in many chemical and biological systems and especially in waste treatment plants. The neutralization process is very fast and occurs as a result of a simple reaction. However, from the control point of view it is a very difficult problem to handle because of its high nonlinearity due to the varying gain (in the range of 1 up to 10^6) and varying dynamics with respect to the operating point (see Figure 1). Introduction of NNs in modeling of processes for control purposes is very useful due to their flexibility in applications.

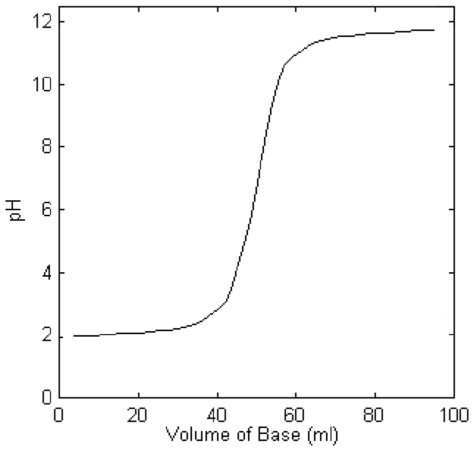

Figure 1. Titration curve of strong acid – strong base system.

In the literature, dynamic mathematical models of pH systems are available [18], [36]. Many control algorithms have been applied to pH control including adaptive, linear model-based, nonlinear internal model, and nonlinear generic model [10], [21], [45], [54], [55].

In this section, a control system having a neuro estimator and a neuro controller is presented and it is used in the control of a pH neutralization system [40].

4.1 Neutralization System

The neutralization system is a highly nonlinear one, whose nonlinearity is reflected in the S shape of the titration curve given in Figure 1. The stirred tank neutralization system that we considered is shown in Figure 2. It has a feed which is composed of one component (acid) and a titrating stream (base). For simplicity, perfect mixing is assumed and the level is kept constant.

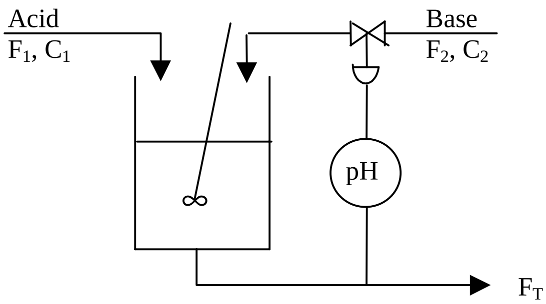

Figure 2. Scheme for the pH process for nonlinear neutralization system.

The material balance can easily be written as [35]

$$V \frac{dC_H}{dt} = F_2 C_2 + F_1 C_1 - (F_1 + F_2)\, C_H \tag{7}$$

Assuming the neutralization reaction is very fast, the equilibrium equation can be written as follows [66]:

$$\frac{K_w}{C_H} + C_1 = C_H + C_2 \tag{8}$$

Using Equations (7) and (8) the change of hydrogen ion concentration can be written as

$$\frac{dC}{dt} = \frac{-C_H^2\left[F_2C_2\text{-}(F_1 + F_2)(\dfrac{K_w}{C_H}\text{-}C_H) - F_1C_1\right]}{V(K_w + C_H^2)} \tag{9}$$

and

$$\mathrm{pH} = -\log(C_H) \tag{10}$$

where

C_1 = concentration of acid (M)
C_2 = concentration of base (M)
F_1 = flow rate of acid (lt/min)
F_2 = flow rate of base (lt/min)
C_H = concentration of hydrogen ion (M)
V = volume of tank (lt)
K_w = water dissociation constant = 1×10^{-14}

Nominal values are
C_{1s} = 0.01 M;
C_{2s} = 0.01 M;
F_{1s} = 0.3 lt/min;
V = 3 lt

C_H is the process state variable while F_2 is selected as the manipulated variable.

4.2 Neural Network Control of the Neutralization System

The structure of the NN controller system is shown in Figure 3. The controller system has a NN controller and a NN estimator.

The estimator is trained by taking the error between the desired plant output and the estimator output. On the other hand the controller is trained by taking the error between the estimator output and a reference point. So the controller assumes that the estimator output matches the plant output.

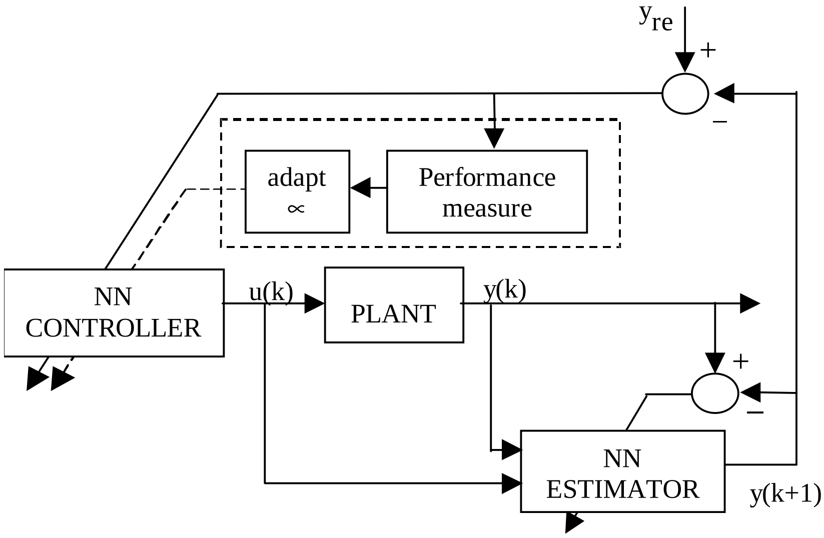

Figure 3. NN controller system used for the neutralization system.

The neural estimator is a multilayer feedforward NN with 10 neurons in the input layer, 20 in the hidden layer and one in the output layer. The values of the initial weights are chosen randomly between –0.1 and 0.1. The backpropagation algorithm is used to train the network. The value of learning rate α is decided by using 1-dimensional search. The input vector for the neuro estimator is chosen as:

$$\mathbf{x}_k = \left[y(k)\ y(k-1)\,..\,y(k-m);\, u(k)\ u(k-1)\,..\,u(k-m)\right] \tag{11}$$

and the output is $y_{\text{estimated}}(k+1)$.

After training the neural estimator the controller starts and the window data for the estimator are updated. The neural controller is also a multilayer feedforward NN with 10 neurons in the input layer, 10 in the hidden layer and one in the output layer. The values of the initial weights are chosen randomly between –0.1 and 0.1. The input of the neuro controller is

$$\mathbf{x}_k = \left[y(k)\ y(k-1)\,..\,y(k-m);\, u(k)\ u(k)\,..\,u(k-m)\right] \tag{12}$$

and the output is $u(k+1)$.

Again the backpropagation algorithm is used for training the neuro controller; however, learning rate α is chosen as a function of the square of the error, P_k. At sampling time, k, α is calculated as a function of the P_k, according to the set of linguistic rules:

If P_k is LARGE then α is 0.1.
If P_k is MEDIUM then α is 0.01.
If P_k is SMALL then α is 0.001.

The linguistic variables for P_k can be chosen as fuzzy sets, but here they are divided arbitrarily into regions as

LARGE = [25 – 16]
MEDIUM = [16 – 09]
SMALL = [09 – 00]

4.3 Results

The NN control system described in Section 4.2 is used for control of the neutralization system. Also a PID controller is designed for comparison. These controllers are compared for set point tracking and disturbance rejection cases.

In set point tracking the initial steady state point in pH is taken as 2.0 and a change of 5.0 is considered to reach a neutral point of pH = 7.0.

In disturbance rejection the system is considered to be at the neutral point at the start as pH of 7.0 and then a –20% load change is given to the flow rate of acid at t = 25 min and a +20% change is given to the concentration of the base solution at t = 100 min to test the performance of the controllers.

4.3.1 Conventional PID Controller Performance

Tuning of the PID controller is done with Ziegler-Nichols rules [36], [52]. The responses of the system for set point tracking and disturbance rejection are given in Figures 4 and 5. It is seen that the conventional PID controller has failed to control the neutralization system.

4.3.2 NN Controller Performance

The output of the neural estimator in comparison with the actual plant output is shown in Figure 6 for different inputs. The responses of the NN controller for set point tracking and disturbance rejection are given in Figures 7 and 8.

As can be seen in Figure 7 despite the oscillations seen in the first 40 minutes the NNC brings the system to set point and is better than a conventional PID. It is seen from Figure 8 that NNC works better for disturbance rejection compared to set point tracking.

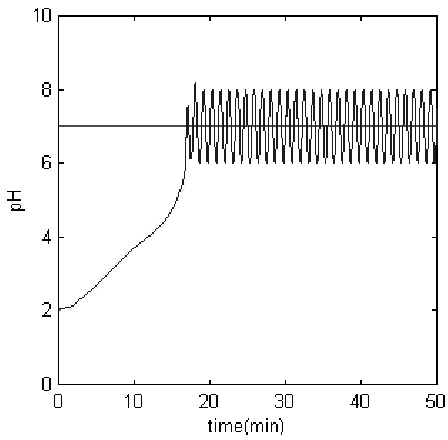

Figure 4. Set point tracking by PID Controller.

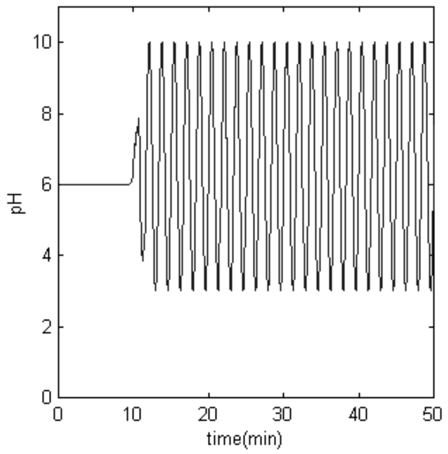

Figure 5. Disturbance rejection by PID Controller.

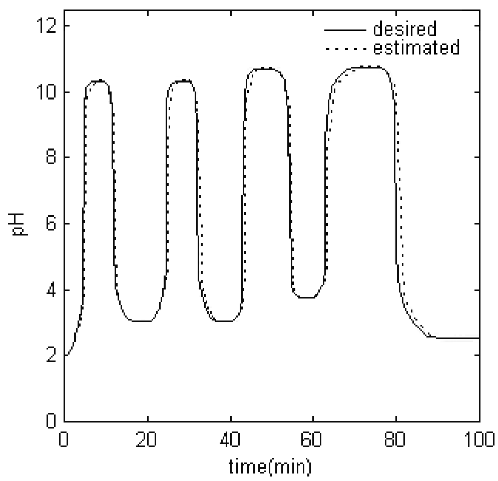

Figure 6. Neural estimator output and actual output for different inputs.

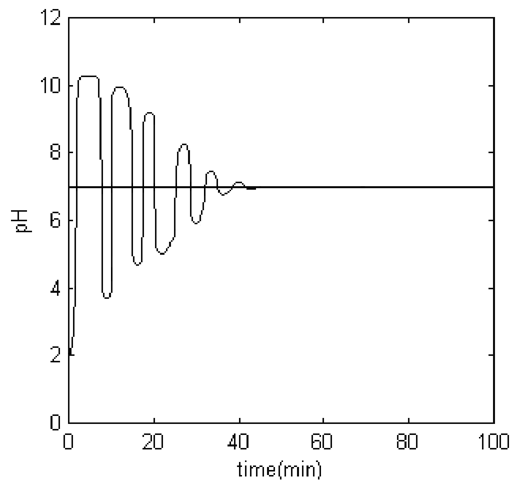

Figure 7. Set point tracking by NN Controller.

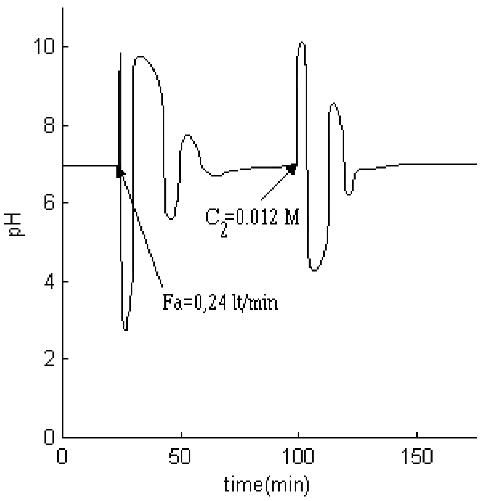

Figure 8. Disturbance rejection by NN Controller.

5 Case Study II: Adaptive Nonlinear-Model Predictive Control Using Neural Networks for Control of High Purity Industrial Distillation Column

In recent years, considerable interest has been devoted to a special class of model based control techniques referred to as Model Predictive Control (MPC) [15], [17]. The basic idea behind MPC algorithm is to use a process model to decide how to adjust the available manipulated variables, in response to disturbances and changing production goals. Control design methods based on the MPC concept have gained high popularity due to their ability to yield high performance control systems. The distinctive feature of the MPC technique is to predict the future behavior of the process outputs based on a non-parametric model, namely, impulse response or discrete convolution model. These can be directly and easily obtained from samples of input-output data without assuming a model structure. Therefore, the MPC technique is especially useful for processes exhibiting unusual dynamic behavior [13].

MPC technique is based on a linear model and, therefore, it is not very well suited for the control of nonlinear systems. Because of this, there have been numerous efforts to extend the linear MPC technique for the control of nonlinear systems [8], [33].

In this work, a new Adaptive Nonlinear-Model Predictive Controller (AN-MPC) utilizing a NN in the MPC work frame is proposed for the adaptive control of nonlinear SISO systems. This technique is used in the control of top-product composition of a distillation column as an application [26], [27].

5.1 Multicomponent High Purity Distillation Column

The performance of the proposed controller is tested on an industrial multi-component high-purity distillation column using an unsteady-state simulation program. The simulation used represents the distillation column in the catalytic alkylation section of the styrene monomer plant of Yarımca Petroleum Refinery, in Izmit, Turkey. In this case study, instead of obtaining the off-line training data from the actual system, the simulator is used because of practical reasons. Since it is a high purity distillation column, it exhibits highly nonlinear characteristics. The unsteady-state simulation package, which is named as DAL, is developed by Alkaya, in 1991 [2].

The distillation column, which has 52 valve trays, was designed to separate Ethyl-Benzene (EB) from a mixture of Ethyl-Benzene, Methyl-Ethyl-Benzene and Di-Ethyl-Benzene having a mole fraction 0.951, 0.012 and 0.037 respectively with a desired top product composition of 0.998. In the process, the top product composition of Ethyl-Benzene is controlled by manipulating the reflux rate as shown in Figure 9.

5.2 Adaptive Nonlinear-Model Predictive Controller Using Neural Networks

5.2.1 Linear Model Predictive Controller

Linear MPC technique may utilize an impulse response model as shown in equation (13) to predict the future behavior of the controlled output as a function of the respective manipulated variable.

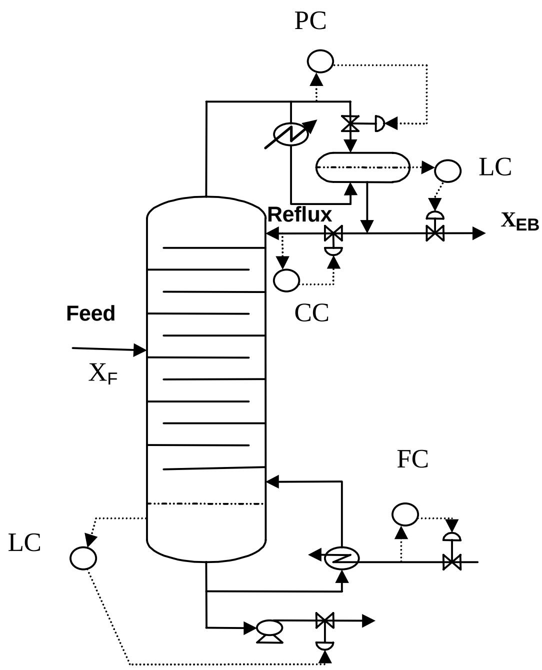

Figure 9. Distillation column.

$$\hat{C}_{n+1} = C_n + \sum_{i=1}^{T} h_i \Delta m_{n+1-i} \tag{13}$$

where

$\hat{C}_{n+1}$	represents the predicted value of the output for the n+1[th] sampling,
C_n	represents the actual value of the output at n^{th} sampling,
h_i's	represent the impulse response coefficients relating the controlled output to step changes in manipulated variable,
T	represents the MPC model horizon, which determines the number of impulse response coefficients,
Δm_i's	represent the implemented step changes in manipulated variable along model horizon prior to $n+1^{th}$ sampling.

Defining r_{n+1} as the set point of the output for the $n+1^{th}$ sampling, the linear MPC law based on equation (13) is formulated as follows:

$$\Delta m_n = (r_{n+1} - C_n - \sum_{i=2}^{T} h_i \Delta m_{n+1-i}) / h_i \tag{14}$$

In equation (14), Δm_n, which is the value of the step change in manipulated variable at n^{th} sampling, is computed to bring the predicted response to set point at n+1$^{\text{th}}$ sampling.

5.2.2 Nonlinear-Model Predictive Controller

While the impulse response coefficients, h_i, obtained for a linear system at an operating point can be successfully used for other points, they can only be used for a nonlinear system by local linearization. Thus, there will always be a deviation between the predicted values of the output and the actual system output in nonlinear systems. Therefore, in such systems, this deviation may result in poor control performance when equation (14) is used directly.

However, if the modeling error that comes out at n+1$^{\text{th}}$ sampling is estimated somehow, then the linear MPC law can be re-formulated to obtain nonlinear MPC law as given below:

$$\Delta m_n^* = (r_{n+1} - P_{n+1} - C_n - \sum_{i=2}^{T} h_i \Delta m_{n+1-i}^*) / h_i \tag{15}$$

where Δm^*'s are the step changes in manipulated variable, and P_{n+1} is the deviation at n+1$^{\text{th}}$ sampling as defined below:

$$P_{n+1} = C_{n+1} - \hat{C}_{n+1} \tag{16}$$

5.2.3 Adaptive Nonlinear-Model Predictive Controller via Neural Networks

A Nonlinear-Model Predictive Controller (NMPC) based on equation (15) can be used to control a nonlinear process, unless there is a change in the process conditions. However, if the system parameters change during control operation, then the process model must be adapted to reflect the changes through the use of the estimator for P_{n+1}.

In the NMPC structure, the process is represented by a combination of a linear model and a NN model. The NN used in the NMPC provides an estimate for the deviation between the predicted value of the output computed via linear model and actual nonlinear system output, at a given sampling time. The adaptation of the process model is achieved by updating the NN model via on-line training using the real-time data obtained from the process. Therefore, by continuously training the NN for changes in process dynamics, the NMPC can be used as an Adaptive Nonlinear-Model Predictive Controller (AN-MPC) without any further modification. The resulting AN-MPC structure is shown in Figure 10.

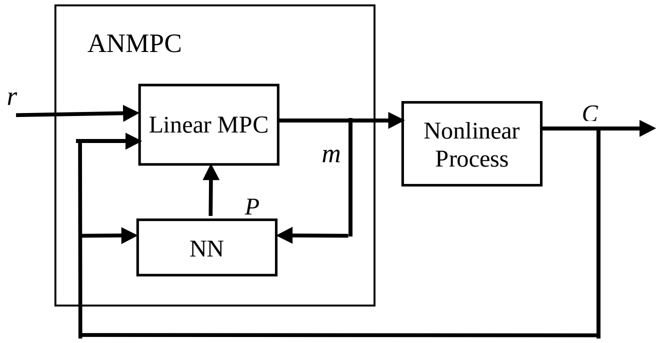

Figure 10. AN-MPC Structure using NN.

In this control structure, the NN is trained at each sampling time, with the present and T previous values of system input-output data with respect to the deviation. Trained NN is used to estimate deviation between the predicted and the actual value of the output. Consequently, AN-MPC computes the value of the manipulated variable, which should be implemented at the present sampling time using equation (15). The NN used in this study is a multi-layer feed-forward NN as shown in Figure 11.

Input vector to NN, $\mathrm{U}^k \in \mathrm{R}^{2T+1}$, is composed of two sub-input vectors: present and T past values of output and input of the nonlinear system.

$$\mathbf{U}^k = \left[C_n, C_{n-1}, .., C_{n-T}, m^*_n, m^*_{n-1}, .., m^*_{n-T}\right]^{\mathrm{T}} \tag{17}$$

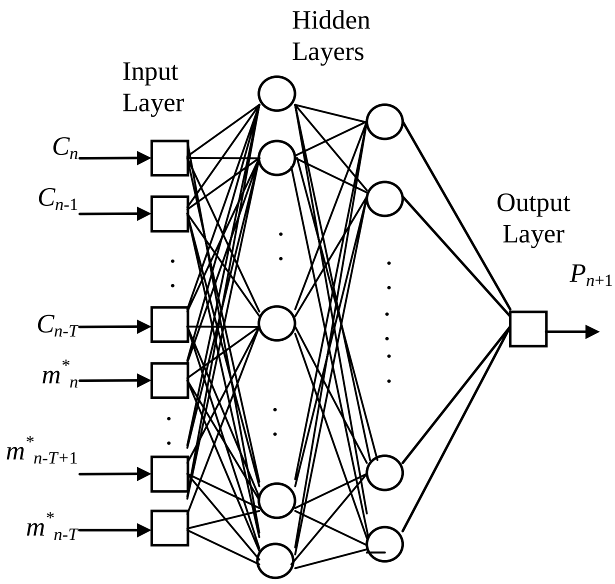

Figure 11. Multilayer NN used in AN-MPC.

As explained before, the NN is trained such that its output vector, which has a single element P_{n+1}, is the deviation of the nonlinear model from its linear MPC model for next sampling.

Since the deviation P_{n+1} is a function of present and past values of the process input and output, these two components of the input vector are shifted in a forward direction at each sampling.

The training of NN is done using backpropagation algorithm. Two types of training strategies, off-line training and on-line training, are used in this particular application. In off-line training the NN is trained to obtain the deviation around an initial operating point prior to control operation. The data required for this are obtained by utilizing a step response experiment where K consecutive step inputs are applied to the system, in the open-loop. That is, the system output resulting from K consecutive step inputs (one step change at each sampling) is observed and compared with the MPC prediction, at each sampling. The difference among them constitutes the deviation for each sampling.

In on-line training NN is continuously trained to obtain the deviation using on-line data for adaptive control purposes using AN-MPC. Thus at each sampling, actual output is observed and compared with its predicted value to compute the deviation. Then, this input-output and deviation data obtained from the system are used to train the NN, at each sampling.

5.3 Identification

The first step in the development phase of the AN-MPC for the distillation column is identification, where impulse response coefficients representing the relationship between reflux rate and EB composition at the top at a sampling period Δt are determined. Consequently, when a unit step input is given to the reflux rate, the top product EB mole fraction changes from 0.9988 to 0.9995 within 7.06 hours which is the response time of the process (Figure 12).

Therefore, settling time of the process is found to be 6.5 hours. Since, the settling time is too large, the model horizon, T, is chosen as 50. From this, sampling period is calculated as 0.13 hours (7.8 minutes) and impulse response coefficients are determined as given in Table 1.

Having determined the impulse response coefficients and MPC model horizon, T, the discrete convolution model (Equation 13) relating top product EB mole fraction to reflux rate is found, where C and Δm stand for top product EB mole fraction and step change in reflux rate, respectively.

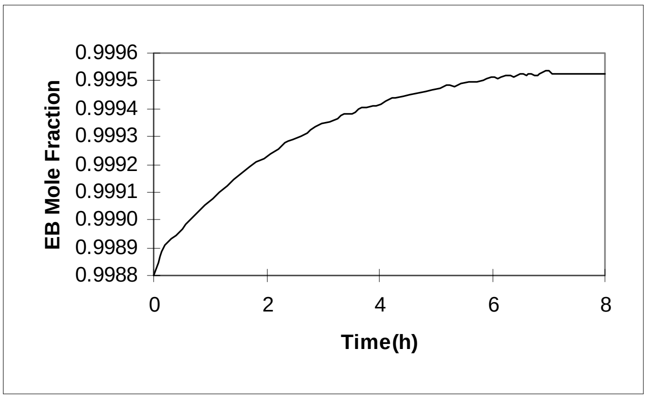

Figure 12. Unit Step Response.

Table 1. Impulse Response Coefficients.

i	$h_I \times 10^5$	i	$h_I \times 10^5$	i	$h_I \times 10^5$	i	$h_I \times 10^5$	i	$h_I \times 10^5$
1	2.7	11	2.8	21	1.2	31	0.6	41	–0.2
2	3.3	12	1.8	22	1.9	32	1.6	42	0.8
3	2.4	13	2.4	23	1.4	33	0.6	43	0.8
4	2.4	14	2.0	24	0.4	34	0.5	44	–0.2
5	3.4	15	1.0	25	1.4	35	0.6	45	0.8
6	2.6	16	2.0	26	1.4	36	0.6	46	0.8
7	2.6	17	1.6	27	0.4	37	0.6	47	–0.2
8	2.6	18	2.2	28	1.4	38	0.5	48	0.8
9	2.3	19	1.2	29	0.8	39	0.8	49	–0.2
10	1.8	20	1.2	30	0.6	40	0.8	50	0.8

5.4 Development of the Neural Network Model

The second step in the development of AN-MPC is the development of a NN model representing the deviation of the linear MPC model from the actual (nonlinear) system through off-line training. This is accomplished in three steps: 1. Obtaining the training data for the NN by utilizing an open-loop step response experiment; 2. Determination of a suitable NN architecture by following a trial and error procedure; 3. Off-line training of NN by using the data obtained in the first step. This enables the NN model to operate satisfactorily at the start. Otherwise, the initial modeling uncertainty for the NN can be too large and the system may become unstable at the beginning of the control operation.

The off-line training data for NN model are obtained through a step response experiment where 50 arbitrary consecutive step changes are introduced to manipulated variables as shown in Table 2, and the response is observed as shown in Figure 13. At each sampling time, by using the linear model of equation (13), the system response, and equation (16), deviation of linear model predictions from the actual output is calculated. Then, using these data the training vectors for the NN are created.

A 10^{-8} order of magnitude error in P_{n+1} results in a 10^{-3} order of magnitude change in the control input, which is acceptable for this application. Therefore training of the NN is terminated when the error in training is less than or equals to 1×10^{-8}. Since the model horizon, T, is chosen as 50, the number of nodes in the input layer is 102. By following a trial-error procedure a suitable NN architecture satisfying the training-stop criteria is determined as a three-layered feed-forward NN, having 104 and 50 nodes in the first and second hidden layers with sigmoid type activation functions, and an output node with an identity type activation function.

Table 2. Step Changes in Reflux Rate.

t (h)	Δm (lbmol/h)	t (h)	Δm (lbmol/h)	t (h)	Δm (lbmol/h)
0.00	2.10	2.21	0.21	4.42	2.10
0.13	2.20	2.34	0.19	4.55	2.20
0.26	–0.20	2.47	0.78	4.68	4.40
0.39	–0.14	2.6	1.78	4.81	–1.10
0.52	–6.00	2.73	1.90	4.94	–2.20
0.65	–4.50	2.86	1.30	5.07	–3.20
0.78	2.10	2.99	1.40	5.20	–0.19
0.91	2.20	3.12	–0.17	5.33	–0.21
1.04	4.40	3.25	–0.19	5.46	–3.45
1.17	–1.10	3.38	–0.70	5.59	–2.78
1.30	–2.20	3.51	–0.50	5.72	–0.19
1.43	–3.20	3.64	2.10	5.85	0.21
1.56	–0.19	3.77	2.20	5.98	0.19
1.69	–0.21	3.90	–0.20	6.11	0.78
1.82	–3.45	4.03	–0.14	6.24	1.80
1.95	–2.78	4.16	–6.00	6.37	1.90
2.08	–0.19	4.29	–4.50		

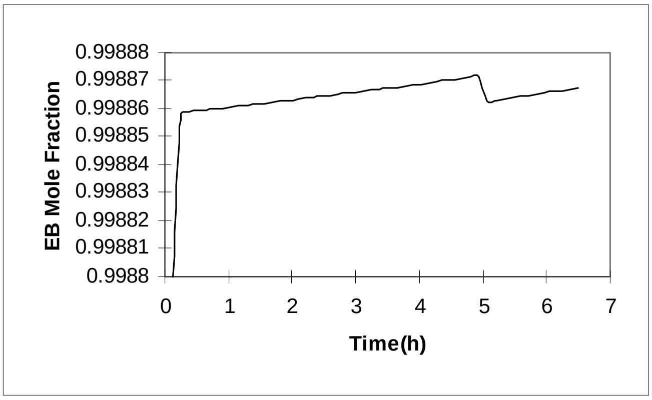

Figure 13. Response of the Distillation Column to changes given in Table 2.

5.5 Control Application

After obtaining the linear MPC model, model horizon and NN model, which represents the deviation of linear MPC model from the actual system, these two models are combined in an MPC workframe. AN-MPC is obtained, in which on-line training of NN is maintained continuously to adapt the controller for changes in process operating conditions. The AN-MPC is tested for its set point tracking and disturbance rejection capabilities. In order to test the performance of the AN-MPC and compare it with that of linear MPC for disturbance rejection capability, the Ethyl-Benzene (EB) mole fraction in feed composition was decreased by 3% (from the steady-state value of 0.9513 to 0.9228), keeping relative mole fractions of Di-Ethyl-Benzene and Methly-Ethyl-Benzene constant. The open-loop response of the process, for this –3% disturbance in the feed composition, is given in Figure 14.

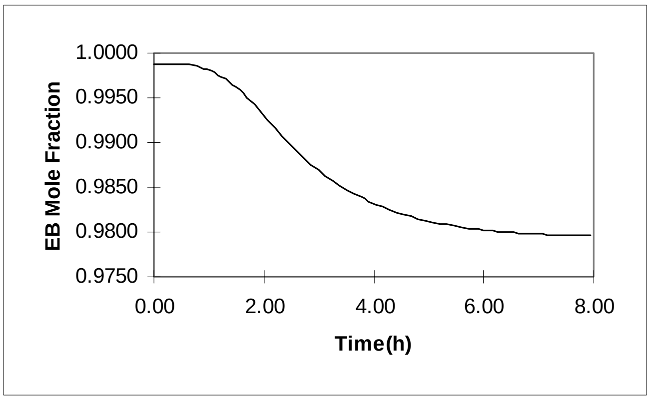

Figure 14. Open-loop Response.

The closed-loop response of the process with the linear MPC and the corresponding control inputs are given in Figures 15 and 16, respectively. The closed-loop response of the process with AN-MPC and corresponding control inputs are given in Figures 17 and 18, respectively.

As it can be seen from Figure 14, when –3% disturbance is introduced to EB mole fraction in feed, the EB mole fraction in top product changes from 0.9988 to 0.9797 within 7 hours. When the linear MPC is used to control the system, the controlled response shows some deviation from set point (Figure 15) and control input is very oscillatory (on-off type) changing between zero reflux and total reflux. Obviously, such behavior of the reflux rate for a distillation column is not practically acceptable. Whereas, when the AN-MPC is used to control the system, the controlled response, as shown in Figure 17, shows little deviation from the set point and, in this case, it matches the set point after 5 hours. Furthermore, the control input (Figure 18) exhibit much smoother behavior than that of linear MPC and they change within reasonable limits.

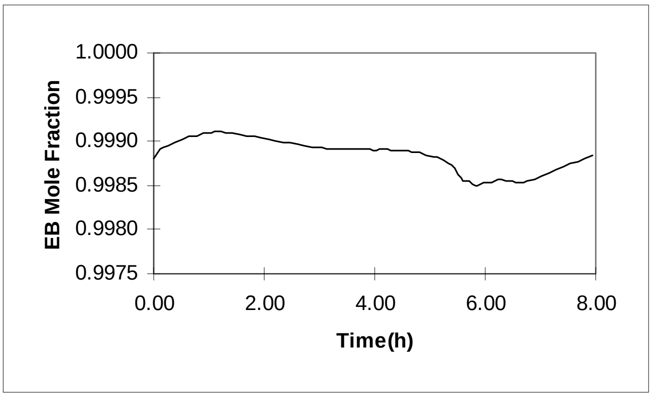

Figure 15. Closed-loop response of the distillation column, which is under control of linear MPC, to a –3% step change in EB Feed Composition.

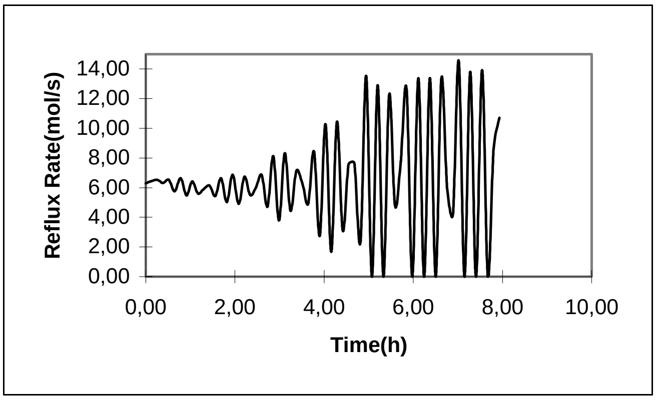

Figure 16. Control actions (reflux rate changes) of linear MPC for a –3% step change in EB feed composition.

In order to test performance of the AN-MPC and to compare it to that of linear MPC for set point tracking capability, the set point is changed in the EB mole fraction from 0.9988 to 0.9900. For this change, the closed-loop response and respective control actions of linear MPC are as shown in Figures 19 and 20, and the closed-loop response and respective control actions with AN-MPC are shown in Figures 21 and 22 respectively.

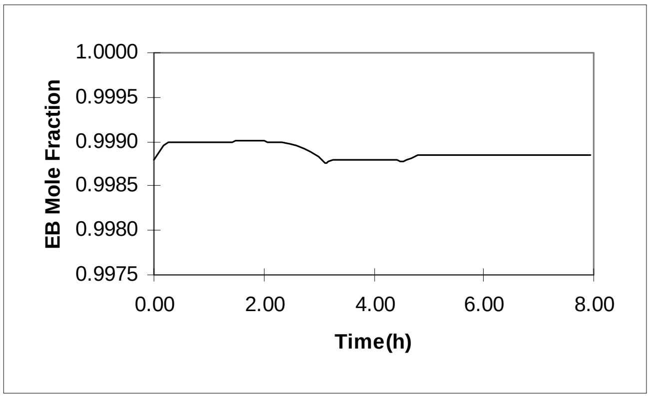

Figure 17. Closed-loop response of the distillation column, which is under control of AN-MPC, to a –3% step change in EB feed composition.

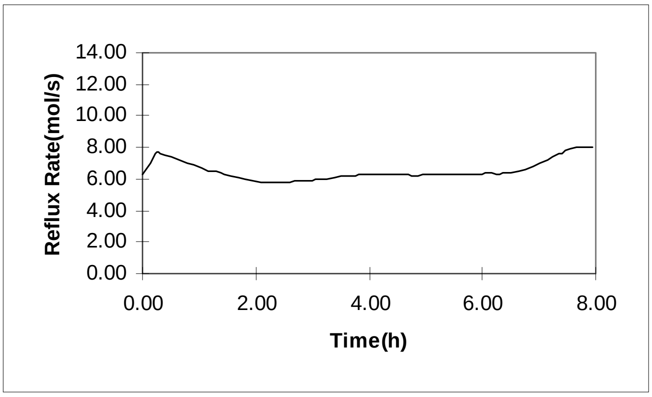

Figure 18. Control actions (reflux rate changes) of AN-MPC for a –3% step change in EB feed composition mole fraction.

As can be seen from the Figures 19 and 20, the controlled output using linear MPC is oscillatory and does not match with the new set point and the respective control input shows high oscillations. However, control input of AN-MPC is quite smooth and exhibits very small oscillations compared to that of linear MPC. Furthermore, the output controlled with the AN-MPC matches the set point within 6 hours with a very small oscillation compared to that of linear MPC as it can be observed in Figures 21 and 22.

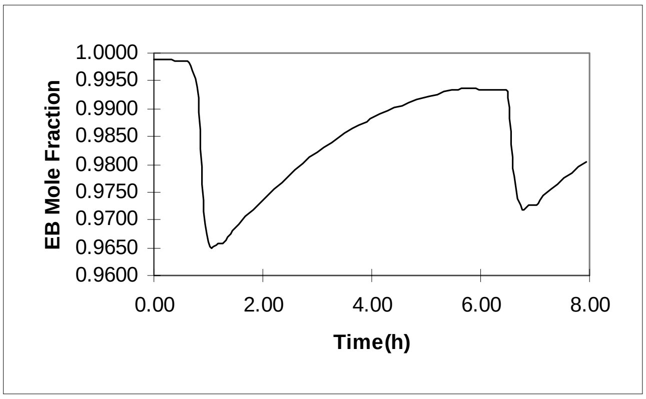

Figure 19. Closed-loop response of the distillation column, with linear MPC, to a set point change of –0.0088 in EB.

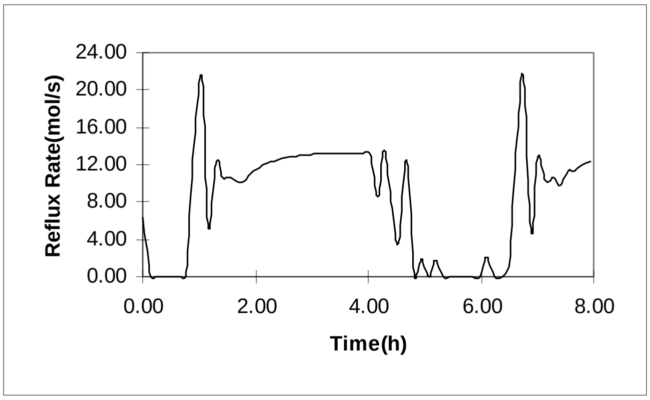

Figure 20. Control actions (reflux rate changes) of linear MPC for a set point change of –0.0088 in EB mole fraction.

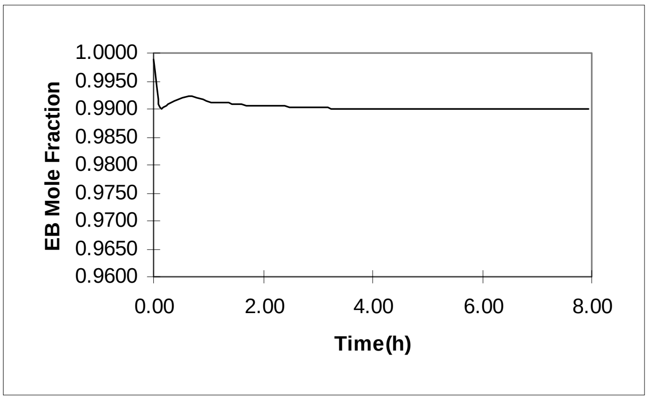

Figure 21. Closed-loop response of the distillation column, with AN-MPC, to a set point change of –0.0088 in EB mole fraction.

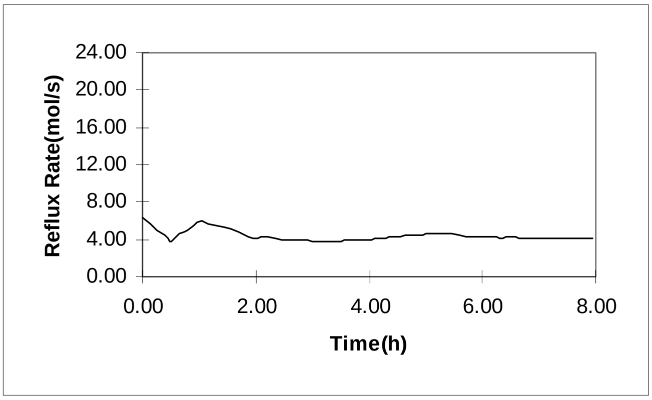

Figure 22. Control actions (reflux rate changes) of AN-MPC for a set point change of –0.0088 in EB mole fraction.

6 Case Study III: PI Controller for a Batch Distillation Column with Neural Network Coefficient Estimator

The main problem with the conventional PI type controllers is the determination of proportional and integral coefficients for each operating (bias) point. In this section, a control method in which a NN is incorporated as an online parameter estimator for the PI-type controller is proposed and used in the control of a binary batch distillation column [60].

6.1 Binary Batch Distillation Column

Batch distillation is an important unit operation where small quantities of high technology/high value added chemicals and bio-chemicals are to be separated. The other separation unit, which is widely used in the chemical industry, is the continuous distillation column. Unlike batch distillation, the mixture, which is separated, is continuously supplied to the column in the continuous distillation case. The most outstanding feature of batch distillation is its flexibility. This flexibility allows one to deal with uncertainties in feed stock or product specification. The operation of a batch distillation column can be described as three periods: start up, production and shutdown periods. The column usually runs under total reflux in the start up period until it reaches the steady state where the distillate composition reaches the desired product purity [12], [35].

We will consider a basic separation system as depicted in Figure 23. This column is used to separate two components in the liquid mixture by taking advantage of the boiling points; that is, the component with the lower boiling point will tend to vaporize more readily and therefore can be selectively collected in the vapor boiled off from the liquid [35].

The basic requirement of the simulation to be developed is to compute the overhead or distillate composition (condenser product) as a function of time. If we consider a binary mixture, the lighter component will have a higher composition in the distillate than in the kettle (bottoms). However as the total amount of binary is reduced due to continued withdrawal of the distillate, the concentration in the light component in the distillate will decrease and get to an eventually low level. This decrease in the more volatile component concentration while inevitable can be delayed by increasing the reflux ratio during the distillation at the expense of the distillate (product) flow rate.

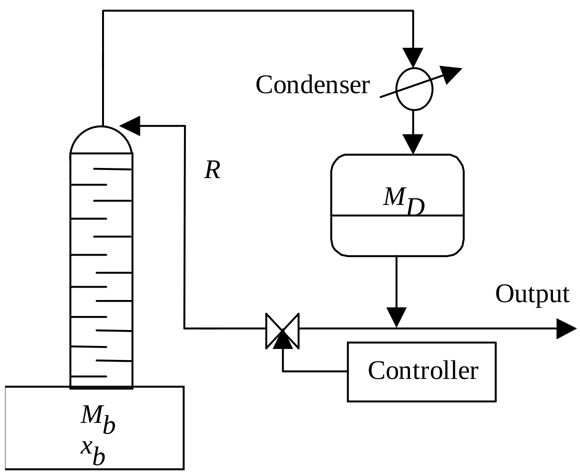

Figure 23. Binary batch distillation column with composition controller.

Dynamic simulation of the batch distillation column and investigation of an automatic control system for distillate composition have been done in the study by using the assumptions [35]:

1. Reflux drum and tray holdups are constant;

2. Binary system with constant volatility;
3. Equimolar overflow;
4. Vapor-liquid equilibrium is attained in each tray;
5. Vapor holdup is negligible when compared with liquid holdup.

The variables for the column model are:

H = Liquid holdup (mole)
G = Vapor holdup (mole)
L = Liquid flow rate (mole/sec)
R = Reflux rate (mole/sec)
M_b = Kettle holdup (mole)
C = Condenser holdup (mole)
t = Time (sec)

At liquid phase, total molar balance for the plate i is given by

$$\frac{dH_i}{dt} = L_{i+1} - L_i \tag{18}$$

Since H_i is assumed to be constant, thus $dH_i/dt = 0$, we conclude that $L_{i+1} = L_i = L_{i-1} = L_1 = R$. In nth tray the vapor phase total molar balance gives

$$\frac{dGi}{dt} = V_{i+1} - V_i \tag{19}$$

Since the vapor holdup is assumed to be constant, $V_{i+1} = V_i = V_{i-1} = ... = V_1 = V$. Total molar balance for the kettle gives

$$\frac{dM_b}{dt} = R - V$$

$$M_b(0)=M_b^0 \tag{20}$$

Since $dM_b/dt \neq 0$ the total amount of liquid in the kettle changes significantly with time. The component balance for the kettle, trays, and condenser gives:

Kettle:

$$\frac{d\left(M_b x_b\right)}{dt} = Rx_1 - Vy_b \tag{21}$$

$M_b(0)\, x_b(0) = M_b^0\, x_b^0$

Plate 1:

$$H\frac{dx_1}{dt} = Rx_2 + Vy_b - Rx_1 - Vy_1 \tag{22}$$

$x_1(0) = x_b^0$

where H is the constant liquid holdup on each of the trays. The initial liquid composition on each of the trays is taken as the initial kettle composition x_b^0, which would occur if the column is initially charged with a single liquid.

Plate 2:

$$H\frac{dx_2}{dt} = Rx_3 + Vy_1 - Rx_2 - Vy_2 \tag{23}$$

$x_2(0) = x_b^0$

Plate i:

$$H\frac{dx_i}{dt} = Rx_{i+1} + Vy_{i-1} - Rx_i - Vy_i \tag{24}$$

$x_i(0) = x_b^0$

Plate N:

$$H\frac{dx_N}{dt} = Rx_d + Vy_{N-1} - Rx_N - Vy_N \tag{25}$$

$x_N(0) = x_b^0$

Condenser:

$$C\frac{dx_d}{dt} = Vy_n - Rx_d \tag{26}$$

$x_d(0) = x_b^0$

The vapor phase concentrations are calculated from the simple vapor liquid equilibrium relation based on the relative volatility α (Raoult's Law).

$$y_i = \frac{\alpha \, x_i}{1 + (\alpha - 1) \, x_i} \tag{27}$$

All the equations given above are used to compute the bottoms holdup $M_b(t)$, and still composition $x_b(t)$, the plate compositions $x_1(t)$, $x_2(t)$, ... , $x_i(t)$, ... , $x_N(t)$ and the distillate composition $x_d(t)$ [35]. In this study, there are 13 trays in the batch distillation column.

There are two basic operation methods for the batch distillation column. The first one is constant reflux rate and variable product composition. The second one is variable reflux and constant product composition of the key component (top product in this case). In this study the aim is to achieve desired constant product composition. Therefore, reflux ratio, R, should be changed during the batch distillation operation. In literature, different methods have been applied to the distillate control problem. The majority of these efforts tried to solve the problem by using optimal control techniques. In these studies, Pontryagin's maximum principle was used in order to maximize the distillate composition [57].

If the column at a bias point is uncontrolled, then the distillate composition $x_d(t)$ would drop off substantially after some time, where it should remain at the relatively high value to give a product of the required purity. In order to remedy this situation we will add to the model the equations for an automatic control system.

Basically a feedback configuration is considered for the control purpose. The term 'feedback' comes from the way in which such a controller works. The variable to be controlled, in this case the distillate composition $x_d(t)$, is sensed (measured) and then compared with the desired value, the set point x_{dset}, to form an error $e(t)$.

$$e(t) = x_{dset} - x_d(t) \tag{28}$$

An ideal controller would keep the error at zero $e(t) = 0$, for which the distillate composition would equal the set point $x_{dset} = x_d(t)$. However, a real controller can not achieve this ideal performance, and it is attempted to design a controller that comes as close as possible to this ideal.

Once the controller generates the error, it is used to modify the manipulated variable within the system to be controlled. In this case the manipulated variable is the reflux rate, R. The manipulation of R will be done according to the controller equation,

$$R = R_{ss} + K_c \left(e + \frac{1}{T_I}\int_0^t e\,dt\right) \tag{29}$$

where

R_{ss} = steady state reflux
K_c = controller gain
T_I = controller integral time

This equation describes the action of a proportional integral controller; the first term, $K_c e$, is the proportional part and the second term,

$$\left(K_c / T_I\right)\int_0^t e\,dt \tag{30}$$

is the integral part. We will now consider briefly how each of these sections contribute to the controller of the reflux rate, R. If we had an ideal situation in which the error is always zero, the controller equation simply reduces to $R = R_{ss}$ and the reflux rate would be equal to the steady state value; that is, the batch distillation column would be operating in such a way that the distillate composition is always equal to set point (from error equation).

This situation could never be achieved in practice since the batch distillation column operates in an unsteady state so that $x_d(t)$ is always changing with time and, therefore, does not remain at the set point x_{dset}. If the distillation column has some error $e(t) \neq 0$, the proportional term will change the reflux rate, R, according to the equation $R = R_{ss} + K_c e$. If the error is positive $x_{dset} > x_d\,(t)$, so that the distillate composition is too low, the proportional control term will increase R, which is the correct action in order to increase $x_d(t)$. On the other hand if the error is negative corresponding to $x_{dset} < x_d(t)$, the reflux rate, R, is reduced by the proportional control term, which again is the correct action to reduce $x_d(t)$. Thus the proportional control action always moves the reflux rate, R, in the right direction to bring $x_d(t)$ closer to the set point x_{dset}. The integral control action removes the offset or steady state error. However, it may lead to an oscillatory response of slowly decreasing amplitude or even increasing amplitude, both of which are undesirable [44]. In the rest of the discussion proportional constant K_c is denoted as K_p and the integral term K_c/T is denoted as K_I. In the following section, the role of the NN in the control method of the batch distillation column is explained. Furthermore, several simulation results are also given.

6.2 PI Controller with Neural Network as a Parameter Estimator

The main problem with the PI type controller is the determination of proportional and integral constants (K_p, K_I) for each operating (bias) point. In order to solve this problem, a NN parameter estimator is incorporated into PI control method as shown in Figure 24.

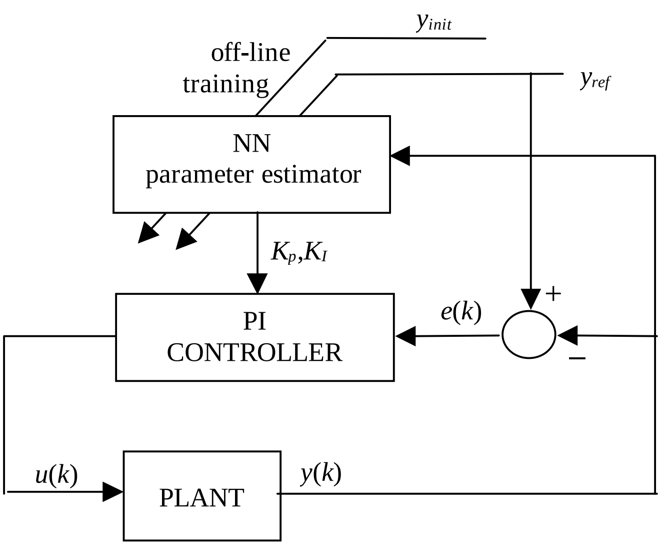

Figure 24. The structure of the PI controller with NN parameter estimator.

The NN shown in Figure 25 is trained for parameter estimation. Actually, the aim of the neural network is to make an interpolation among the operating points of the distillation column and produce the related integral and proportional constants. Hence, a training pair for the neural network is in the form of ([$y_{initial}$, y_{ref}], [K_I, K_P]). The initial and desired bias points actually refer to the initial material concentration and the desired material concentration at the top tray. After training, the neural network can be used as an online parameter estimator for the PI-type controller. As an alternative point of view, the bias points can be seen as the antecedent and the corresponding integral and proportional constants can be seen as the consequent part of an If-Then rule. In this case NN performs an interpolation in the rule space of the system.

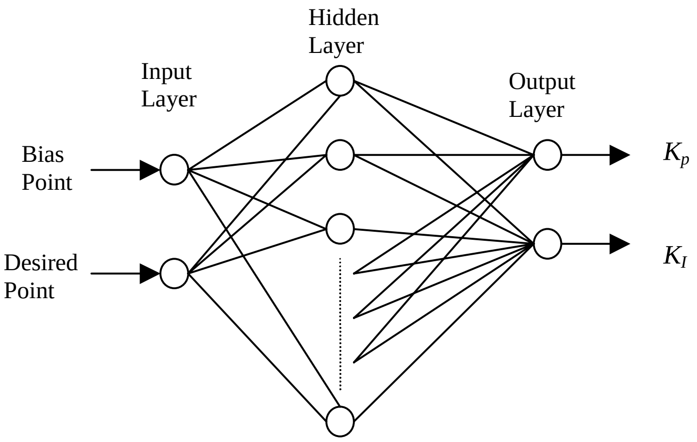

Figure 25. Neural network for parameter estimation.

6.3 Results

In this study, the number of the training pairs is 20 and the training algorithm is the standard backpropagation algorithm. After 50 epochs the mean-square error was reduced to 0.0001. Figure 26 shows a simulation result produced by NN and PI control. The initial concentration for the distillate is 0.5 and the desired concentration is 0.9. NN produced the proportional and integral constants as 8 and 12. It can be seen from the graph that the produced constants yield a satisfactory result. The steady state error is approximately 0.5%.

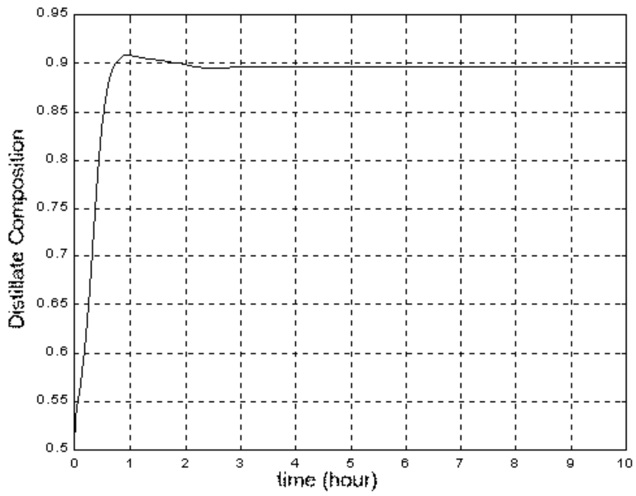

Figure 26. Distillate composition, x_d vs time.

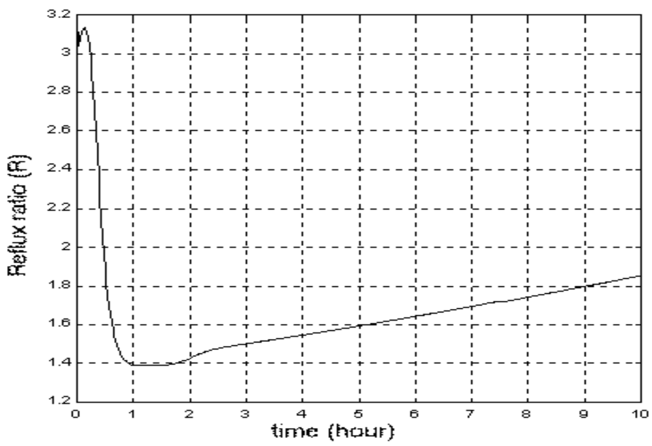

Figure 27. Reflux ratio versus time.

In Figure 27, the corresponding reflux change is given. At the beginning of the operation, reflux ratio decreases in order to increase the distillate composition. After distillate composition reaches its steady state, reflux ratio increases in order to fix the distillate ratio to set point value. It should be noted that, since batch distillation column is used in this study, the material in the still decreases with time. However, the simulation duration is not sufficiently long to observe the fall in the composition in our cases.

Since the amount of maximum overshoot is small, the relative stability of the system is quite good. Figure 28 shows the other two simulations. In the upper part of the Figure 28a, initial distillate composition is 0.5 and the desired (final) composition is 0.8. NN produced the proportional and integral constants as 30 and 20 respectively. In this case, the steady state error is zero; so the PI controller with estimated parameters worked better than the case as shown in Figure 26. The lower part of the Figure 28b shows the simulation results with the initial composition 0.5 and final composition 0.85. It can be seen that desired composition is achieved by the PI control with the help of the NN.

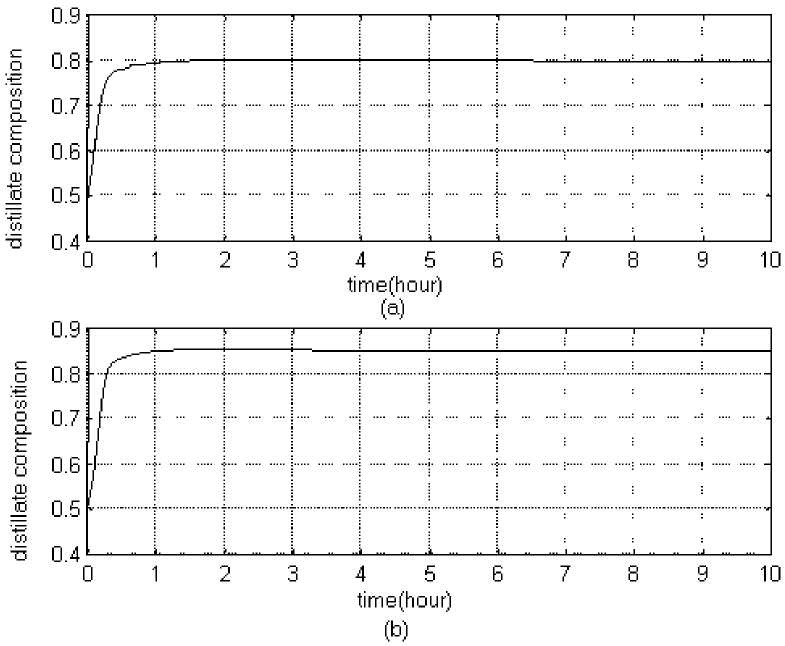

Figure 28. Distillate composition versus time.

In this study, controller parameters are tuned experimentally to achieve fast rise time and small steady-state error and they are used in training the NN. However, some conventional techniques such as the Ziegler-Nichols method can be used for tuning.

7 Case Study IV: A Rule-based Neuro-Optimal Controller for Steam-Jacketed Kettle

In this section, a new method is proposed for the optimal control of multi-input multi-output (MIMO) systems. The method is based on a rule-base derived optimally, which is then interpolated by neural networks.

The design of controllers for MIMO systems has always been a hard problem even for the linear ones [56]. The only prevailing idea used in the control of linear MIMO system is decoupling, if possible at all. During the last decade there have been serious attacks on this problem by methods that are especially constructed to control nonlinear plants, such as neuro-control and sliding mode control techniques [1], [34], [42], [56], [62]. Most of these techniques are quite complicated and possibly work for a particular case only.

The fuzzy control techniques had limited application in MIMO systems control mainly because of the facts that the derivation of rules is not easy (usually not available) and the number of rules is usually large, depending on the number of outputs and states.

Ours is a new attempt to this unsettled problem using a rule-base combined with neural networks. On the other hand there are interesting details and generalizations which will be discussed in the following sections.

7.1 Analysis of the Kettle

The steam-jacketed kettle system has a wide application area in industry. It is especially used in chemical processes. The dynamic response and control of the steam-jacketed kettle shown in Figure 29 are to be considered in this study. The system consists of a kettle through which water flows at a variable rate, w_i kg/min. The inlet water, whose flow rate may vary with time, is at temperature T_i = 5°C. The kettle water, which is well agitated, is heated by steam condensing in the jacket at temperature T_V. This is a three-input two-output system.

Flow rate of inlet water, flow rate of outlet water and flow rate of steam are the control inputs of our system. Temperature and the mass of the water inside the kettle are the outputs [12].

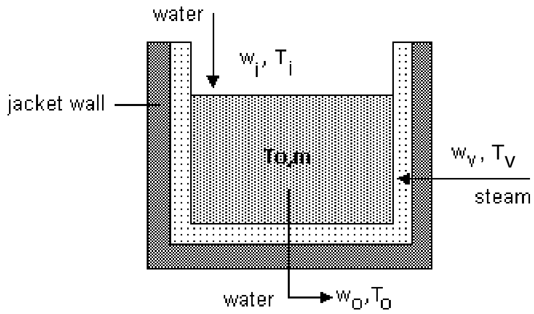

Figure 29. The kettle.

The following assumptions are made for the kettle [12]:

1. The heat loss to the atmosphere is negligible;
2. The thermal capacity of the kettle wall, which separates steam from water, is negligible compared with that of water in the kettle;
3. The thermal capacity of the outer jacket wall, adjacent to the surroundings, is finite, and the temperature of this jacket wall is uniform and equal to the steam temperature at any instant;
4. The kettle water is sufficiently agitated to result in a uniform temperature;
5. Specific internal energy of steam in the jacket, U_v, is assumed to be constant;
6. The flow of heat from the steam to the water in the kettle is described by the expression

$$q = U(T_v - T_o)$$

where
- q = flow rate of heat, J/(min)(m^2)
- U = overall heat transfer coefficient, J/(min)(m^2)(°C)
- T_v = steam temperature, °C
- T_o = water temperature, °C.

The mathematical model of the system can be obtained by making an energy balance on the water side and on the steam side. The symbols used throughout this analysis are defined as follows:

T_I = temperature of inlet water, °C
T_o = temperature of outlet water, °C
w_I = flow rate of inlet water, kg/min
w_o = flow rate of outlet water, kg/min
w_v = flow rate of steam, kg/min
w_c = flow rate of condensate from kettle, kg/min
m = mass of water inside the kettle, kg
m_1 = mass of jacket wall, kg
V = volume of the jacket steam space, m^3
C = heat capacity of water, J/(kg)(°C)
C_1 = heat capacity of metal in jacket wall, J/(kg)(°C)
A = cross sectional area for heat exchange, m^2
t = time, min
H_v = specific enthalpy of steam entering, J/kg
H_c = specific enthalpy of steam leaving, J/kg
U_v = specific internal energy of steam in jacket, J/kg
ρ_v = density of steam in jacket, kg/m^3

Energy balance and mass balance equations for the water and steam side can be written as [12]:

$$mC\frac{dT_o}{dt} = w_i CT_i - w_o CT_o + UA(T_v - T_o) \tag{31}$$

$$\frac{dm}{dt} = w_i - w_o \tag{32}$$

$$m_1 C_1 \frac{dT_v}{dt} = w_v (H_v - H_c) - (U_v - H_c)V\frac{d\rho_v}{dt} - UA(T_v - T_o) \tag{33}$$

$$V\frac{d\rho_v}{dt} = w_v - w_o \tag{34}$$

As can be seen from the equations, the system is a nonlinear one. The state, input and output vectors are:

$$
\mathbf{x} = \begin{bmatrix} T_o \\ m \\ T_v \\ \rho_v \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} w_i \\ w_v \\ w_o \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} T_o \\ m \end{bmatrix} \tag{35}
$$

7.2 A Rule-Based Neuro-Optimal Controller for Nonlinear MIMO Systems

7.2.1 MIMO Systems

It is assumed that a MIMO plant is given with a known mathematical model as shown below

$$
\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\
\mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t))
\end{aligned} \tag{36}
$$

where $\mathbf{x}(t)$, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathrm{R}^n$, $\mathbf{u}(t) \in \mathrm{R}^m$ and $\mathbf{y}(t)$, $\mathbf{g}(\mathbf{x}(t)) \in \mathrm{R}^p$. The system output $\mathbf{y}(t)$ is supposed to track a reference signal $\mathbf{y}_d(t) \in \mathrm{R}^p$.

7.2.2 Rule Derivation

The controller is developed using a rule-base in which the rules are developed by making use of the mathematical model of the plant in an optimal sense. That is, since a model is available, by partitioning the state-space and the output-space and defining a representative for each partition, one can determine the control signals (i.e., rules) optimally, using a suitably chosen cost function.

Suppose that each component of the state vector has N_i, $i = 1, 2, \ldots, n$ regions and the output vector has O_k, $k = 1, 2, \ldots, p$ components. Then there is a total of $(\prod_{i=1}^{n} N_i)(\prod_{k=1}^{p} O_k)$ rules to be derived. If the system state is initially at the ith partition (the representative of which is $\mathbf{x}_i$) and the system's initial and desired states are at partitions O_v and O_k (their representatives are y_v and y_k, respectively), the associated rule can be found optimally by solving the optimal control problem of minimizing the cost function in time interval $[0, t_f]$

$$J(\mathbf{u}) = \frac{1}{2}\left(\mathbf{y}(t_f) - \mathbf{y}_k\right)^T \mathbf{H}\left(\mathbf{y}(t_f) - \mathbf{y}_k\right) + \frac{1}{2}\int_0^{t_f}\left(\mathbf{y}(t) - \mathbf{y}_d(t)\right)^T \mathbf{Q}\left(\mathbf{y}(t) - \mathbf{y}_d(t)\right)dt + \frac{1}{2}\int_0^{t_f}\mathbf{u}(t)^{\mathrm{T}}\mathbf{R}\,\mathbf{u}(t) \tag{37}$$

subject to the state equation

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{x}(0) &= \mathbf{x}_i \\ \mathbf{y}(t) &= \mathbf{g}\left(\mathbf{x}(t)\right) \end{aligned} \tag{38}$$

Usually **H**, **Q** and **R** are diagonal matrices with suitably chosen diagonal entries. The vector function $\mathbf{y}_d(t)$ can be taken as any smooth function with

$$\mathbf{y}_d(0) = \mathbf{y}_v \;,\; \mathbf{y}_d(t_f) = \mathbf{y}_k$$

$$\dot{\mathbf{y}}_d(0) = \dot{\mathbf{y}}_d(t_f) = 0 \tag{39}$$

Furthermore, the constraints on $\mathbf{u}(t)$, that is, $|u_i(t)| \le B_i$, $i = 1, 2, \ldots, m$, can easily be incorporated in our steepest descent like optimal control problem solver [20]. What is supposed to be done is implicitly an interpolation in the function space of optimal controls. Here, it is assumed that the mapping between the given initial-final partitions and the associated optimal control functions is continuous. Therefore, if the number of the partitions is sufficiently high, the approximation error in constructing the optimal control function by a semi-infinite neural network, to be explained in the next section, will be quite small.

7.2.3 Neural Network

In order to be able to generate the control inputs so that the system output trajectory follows an optimal path between arbitrarily specified initial and final output states, one has to train a multilayer perceptron-like neural network [20]. This neural network should accept present output $\mathbf{y}(t_o)$ and desired output $\mathbf{y}(t_f)$ as its inputs and should generate the optimal control signal $\mathbf{u}(t)$ to accomplish the task. The structure of the controller utilizing NN is shown in Figure 30.

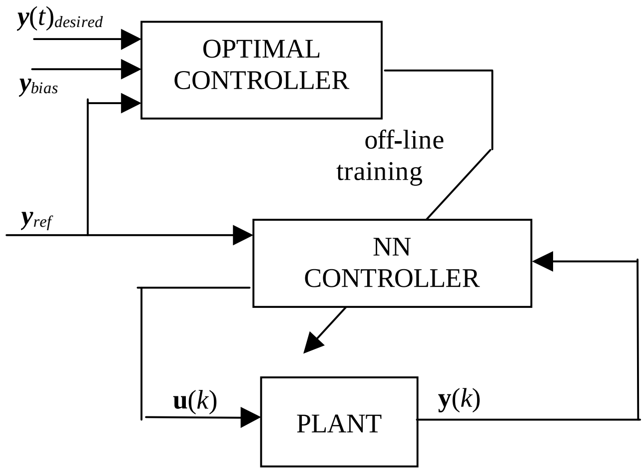

Figure 30. The structure of the rule-based neuro-optimal controller.

For training, input signals produced by optimal control and initial and final points of outputs should be used. It is interesting to note that, at least theoretically, the neural network is a semi-infinite dimensional one [31], [32] in the sense that it is a mapping between the finite dimensional input space and the infinite dimensional output space (i.e., control functions).

The output neurons produce discrete values of input function in $[t_o, t_f]$ interval. Therefore, the neural network can produce the samples of the control signal.

For example, if the number of outputs is n for a single input system, then $\mathbf{y}(t_o)$, $\mathbf{y}(t_f)$ are n-dimensional vectors as

$$\mathbf{y}(t_o)=[\, y_1(t_o)\;\; y_2(t_o)\; ...\; y_n(t_o)]$$
$$\mathbf{y}(t_f)=[\, y_1(t_f)\;\; y_2(t_f)\; ...\; y_n(t_f)]$$

Furthermore, if $[t_o, t_f]$ interval is divided into m parts with sampling period T, a typical training pair is in the form of

$$(\,[\, y_1(t_o)\; y_2(t_o)\; ...\; y_n(t_o)\; ...\; y_1(t_f)\; y_2(t_f)\; \ldots\; y_n(t_f)\,],\; [\, u(0)\; u(T)\; ...\; u(mT)])$$

where the $[u(0)\;\; u(T)\;\; \ldots\;\; u(mT)]$ is the discrete input vector, which moves the system from $\mathbf{y}(t_o)$ to $\mathbf{y}(t_f)$ and is produced by the optimal control. After a training operation, the neural network responds immediately and acts as a real-time controller. In fact, the neural network produces the optimal control vector for the control horizon [$t_{present}$, t_{future}] at $t_{present}$. The control horizon t_{future}-$t_{present}$ is much larger than the sampling duration. As mentioned already, the mapping between the input-output space and optimal control functions is assumed to be continuous. The data (i.e., the optimal control functions obtained by solving the optimal control problem) represent evaluations of this mapping at particular instants. So, the problem of conflicting rules does not exist.

7.3 Results

In our simulation, the output temperature range is chosen as [5°C, 75°C] and the mass range in the tank is chosen as [10kg, 20kg]. There is no need to partition the rest of the states because these are related with the temperature of the steam entering the jacket. Since the temperature of the steam entering is constant, single partition is enough for these states. We divide temperature range into seven regions and mass range into two regions. Therefore, we get 7×7×2×2 = 196 rules from optimal control and we use these 196 rules in order to train our neural networks. Since we have three inputs, three neural networks are constructed, each of them has four inputs, two hidden layers having 100 and 50 neurons and an output layer consisting of 25 neurons. The training algorithm is the backpropagation algorithm having a momentum term. After training, neural networks work as a real time controller. For example, with the initial values for outlet water temperature and mass of the water as [10kg, 5°C] and reference inputs as [15kg, 42°C], our results obtained by on-line NN controllers are given in Figures 31-35. For comparison, the results obtained by the optimal control are also shown in these figures. In Figure 36, water temperature in the kettle, which is controlled by a neural network in real time, is given together with the desired trajectory.

According to Figure 36, neuro-controller performance is satisfactory when compared with the optimal controller performance. After the training stage, the neural network can be used as an online controller. In addition, the output of the neural network can be considered as a function, because it estimates the control functions between two sampling (measurement) intervals. Secondly, the control functions are the optimal ones because the training pairs of the neural network consist of control functions produced by solving the associated optimal control problems.

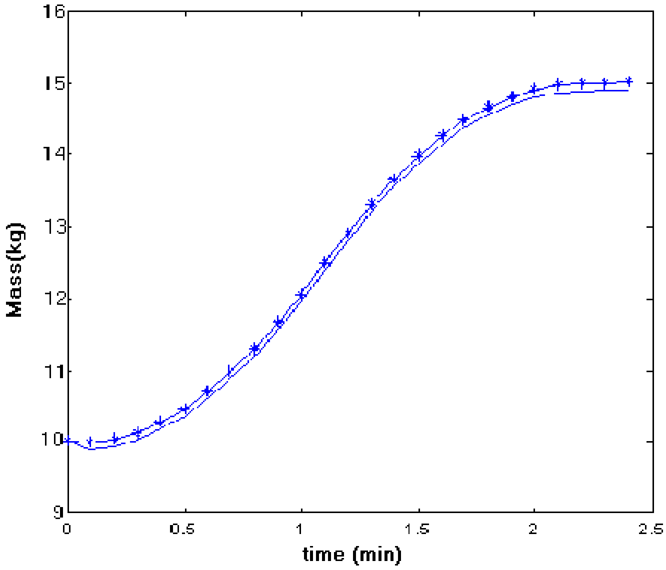

Figure 31. Trajectories for output 1: mass of the water inside the kettle.
* Desired trajectory, - Trajectory from neuro-controller.

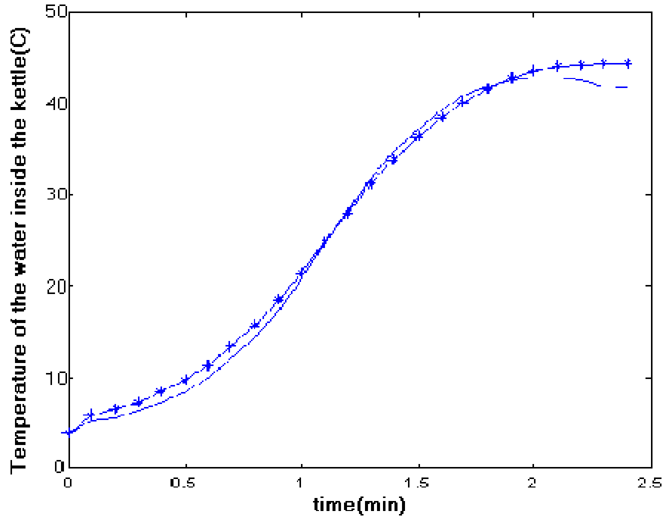

Figure 32. Trajectories for output 2: temperature of the water inside the kettle.
* Desired trajectory, - Trajectory from neuro-controller.

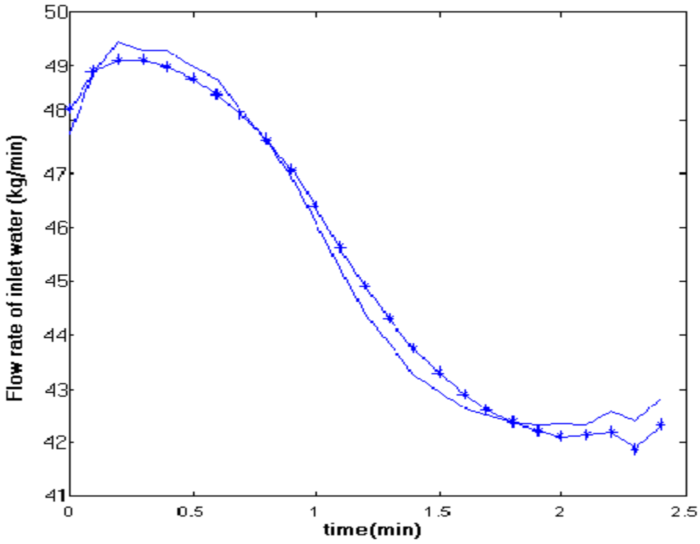

Figure 33. Controlled input 1: flow rate of inlet water.
* Output from optimal control, - Output from neural network.

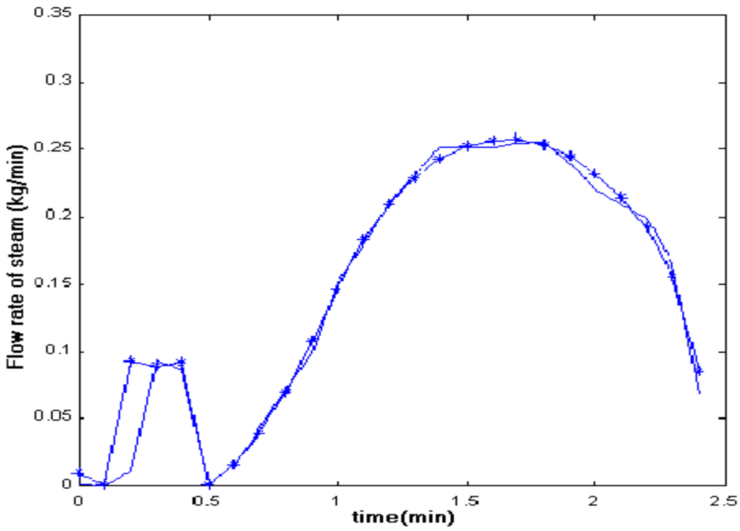

Figure 34. Controlled input 2: flow rate of steam.
* Output from optimal control, - Output from neural network.

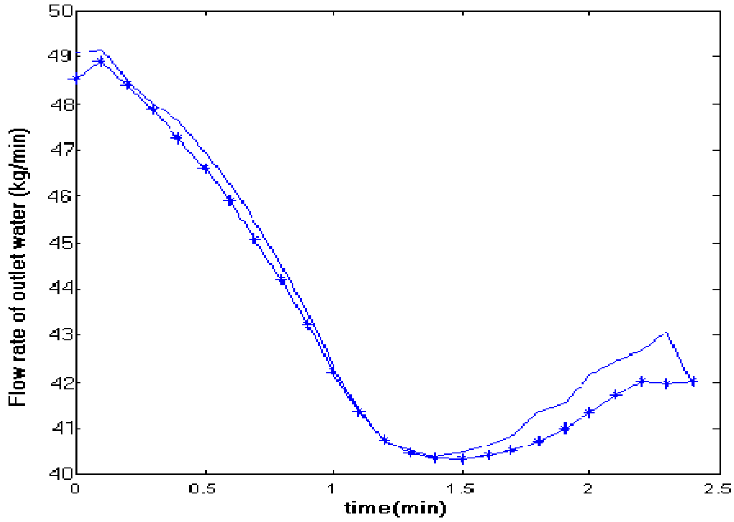

Figure 35. Controlled input 3: flow rate of outlet water.
* Output from optimal control, - Output from neural network.

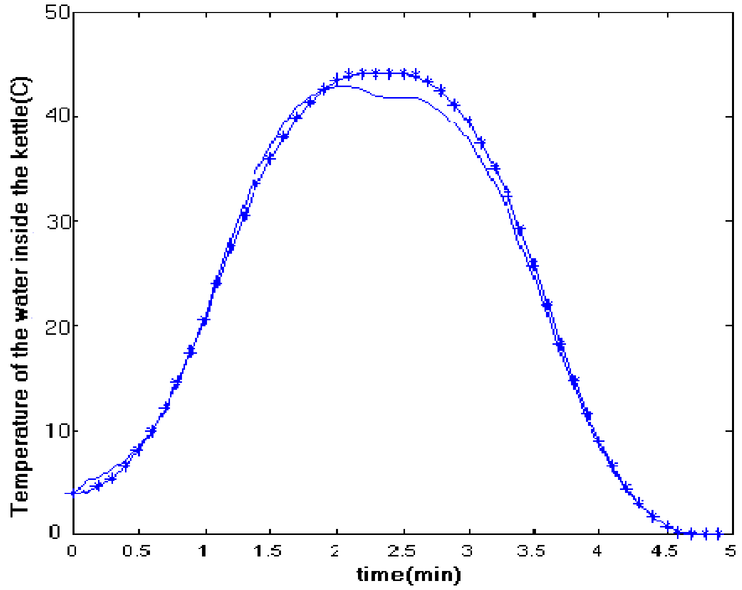

Figure 36. Trajectory for the temperature of water.
* Desired trajectory, - Trajectory produced by on-line NN controller.

8 Remarks and Future Studies

Today's chemical and biological processes in industry are very complex. They are usually nonlinear and/or MIMO. System models of these processes are usually not well defined; either they are missing or system parameters may be time varying. Due to their learning and generalization capabilities, NNs are good candidates for obtaining input-output models of systems. Furthermore, model plant mismatches and the time varying parameter changes in the plant can be overcome by the online training of NNs.

Furthermore, NNs by the “inverse model” of a plant can be used as a “controller” for the plant. Also, NN controllers can be used in MPC structure both as estimator and/or controller parts.

Instead of using NNs alone in control of these processes, they can be combined with conventional approaches such as PI or PID control, optimal control techniques or techniques such as rule based expert systems or fuzzy logic, in a hybrid manner. Such an approach improves the performance of the overall controller.

In this chapter different approaches utilizing neural networks for control of nonlinear processes are presented. Each of them is examined as a case study and tested on nonlinear chemical processes.

In case study I, an NN controller is developed to control a neutralization system which exhibits highly nonlinear dynamics. The controller's performance is tested for both set point tracking and disturbance rejection problems. The NN controller's results are compared with that of the conventional PID controller tuned with Ziegler-Nichols technique. The PID controller failed to control the system by showing oscillatory behavior. However, the NN controller has been able to bring the system to set point, by reducing the oscillations observed at the beginning. Moreover, this NN controller has been able to reject disturbances introduced to the system successfully.

In case study II, linear MPC is used together with NNs to control nonlinear systems. A multilayer NN is used to represent the deviation between the nonlinear system and its linear MPC model. The NN is trained off-line so that the controller operates satisfactorily at the start-up phase. Furthermore, the training of NN is continued on-line using the real-time data obtained from the process. Thus the resultant structure is an adaptive nonlinear MPC controller, AN-MPC.

The performance of the AN-MPC is tested on a simulation of a multi-component high-purity distillation column. Performance tests for disturbance rejection and set point tracking abilities showed that the AN-MPC drives this process quite efficiently, especially in case of set point changes. In contrast, the linear MPC has not been able to control the system for load and set point changes. The success of the hybrid structure, AN-MPC, is because of the fact that the linear MPC determines the coarse control action and NN does the fine tuning. The AN-MPC controller can be generalized further by considering not only the next future sampling instance but next K of them to improve the performance. This generalization is planned as a future work. Currently, we are working to extend the structure to control a MIMO plant.

A hybrid control method which is the combination of PI control and NN is introduced in case study III. The method eliminates the controller (PI) tuning problem with the help of the NN. Therefore, it reduces the parameter estimation time at each operating point. The proposed method was tested in the binary batch distillation column and encouraging results were obtained. The hybrid structure of the method uses advantages of each individual method that constructs the hybrid structure. In order to increase the operating range of the proposed controller, the NN must be trained by a large training set which covers the desired wide operating range. However, the major problem is the training pair derivation for NN. In this study training pairs are determined by heuristic methods. For each bias point in the training set, the proportional and integral constants are determined by a trial and error procedure. Therefore, the training pair extraction process can be a time consuming task for engineers who are not experts in batch distillation. Furthermore, the disturbance rejection and robustness issues of the method were not investigated in this study. Hence, they can be studied as a future work.

In case study IV, an optimal neurocontroller has been suggested for controlling MIMO systems. The proposed controller structure was tested by simulation studies on a simple steam-jacketed kettle system.

The preliminary results obtained so far have shown that this method is worth pursuing further. The only disadvantage of the method is that the number of rules to be derived in a complex plant control can be prohibitively large which also makes the derivation time too long. On the other hand, the method is very simple and can be made adaptive with some effort. Studies are continuing to generalize the method to cover the disturbance rejection and robustness problems as well.

All these case studies showed that the hybrid methods utilizing NNs are very promising for the control of nonlinear and/or MIMO systems that can not be controlled by conventional techniques.

Acknowledgments

This work is partially supported under the grant AFP-03-04-DPT-98K12250, Intelligent Control of Chemical Processes.

References

[1] Ahmed, M.S. and Tasaddug, I.A. (1998), "Neural servocontroller for nonlinear MIMO plant," *IEEE Proceedings Control Theory Applications*, vol. 145, pp. 277-291.

[2] Alkaya, D. and Özgen, C. (1991), "Determination of a suitable measurement in an industrial high purity distillation column," *Proc. of AIChE Annual Meeting on Distillation Column Design and Operation*, Los Angeles, CA, pp. 118-124.

[3] Antsaklis, P.J. (1990), "Special issue on neural networks for control systems," *IEEE Control Sys. Mag.*, vol. 10, pp. 3-87.

[4] Antsaklis, P.J. (1992), "Special issue on neural networks for control systems," *IEEE Control Sys. Mag.*, vol. 12, pp. 8-57.

[5] Bhat, N. and McAvoy, T.J. (1989), "Use of neural networks for dynamic modeling and control of chemical process systems," *Proc. Amer. Contr. Conf.*, pp. 1336-1341.

[6] Bhat, N. and McAvoy, T.J. (1990), "Use of neural networks for dynamic modeling and control of chemical process systems," *Comput. and Chem. Eng*., vol. 14, pp. 573-583.

[7] Boskovic, J.D. and Narendra, K.S. (1995), "Comparison of linear, nonlinear and neural network based adaptive controllers for a class of fed-batch fermentation processes," *Automatica*, vol. 31, pp. 817-840.

[8] Brengel, D.D, and Seider, W.D. (1988), "Multistep nonlinear predictive controller," *Ind. Eng. Chem. Res.*, vol. 28, p. 1812.

[9] Cai, Z.X. (1997), *Intelligent Control: Principles, Techniques and Applications*, World Scientific Publishing.

[10] Chan, H.C. and Yu, C.C. (1995), "Autotuning of gain-scheduled pH control: an experimental study," *Ind. Eng. Chem. Res.*, vol. 34, pp. 1718-1729.

[11] Chen, S., Billings, S.A. and Grant, P.M. (1990), "Nonlinear system identification using neural networks," *Int. J. of Control*, vol. 51, pp. 1191-1199.

[12] Coughanowr, D.R. and Koppel, L.W. (1965), *Process Systems Analysis and Control*, McGraw-Hill.

[13] Cutler, C.R. and Ramaker, B.L. (1979), "Dynamic matrix control – a computer control algorithm," *AIChE National Meeting*, Houston, Texas.

[14] Dreager, A. and Engell, S. (1994), "Nonlinear model predictive control using neural plant models," *NATO-ASI on Model Based Process Control*, Turkey.

[15] Eaton, J.W. and Rawlings, J.B. (1992), "Model predictive control of chemical processes," *Chemical Engineering Science*, vol. 47, pp. 705-720.

[16] Etxebarria, V. (1994), "Adaptive control of discrete systems using neural networks," *IEE Proc. Control Theory Appl.*, vol. 141, pp. 209-215.

[17] Garcia, C.E., Prett, D.M., and Morari, M. (1989) "Model predictive control: theory and practice - a survey," *Automatica*, vol. 25, pp. 335-348.

[18] Gustafsson, T.K., Skrifvars, B.O., Sandstrom, K.V., and Waller K.V. (1995), "Modeling of pH for control," *Ind. Eng. Chem. Res.*, vol. 34, pp. 820-827.

[19] Hamburg, J.H., Booth, D.E., and Weinroth, G.J. (1996), "A neural network approach to the detection of nuclear material losses," *Journal of Chemical Information and Computer Sciences*, vol. 36, pp. 544-553.

[20] Haykin, S. (1996), *A comprehensive Foundation of Neural Networks,* Prentice-Hall, NJ.

[21] Henson, M.A. and Seborg, D.E. (1994), "Adaptive nonlinear control of a pH neutralization process," *IEEE Trans. on Control Systems Technology,* vol. 2, pp. 169-182.

[22] Hernandez, E. and Arkun, Y. (1990), "Neural networks modeling and an extended DMC algorithm to control nonlinear processes," *Proc. Amer. Contr. Conf.*, pp. 2454-2459.

[23] Hernandez, E. and Arkun, Y. (1992), "A study of the control relevant properties of backpropagation neural net models of nonlinear dynamical systems," *Comp. and Chem. Eng.*, vol. 16, pp. 227-240.

[24] Hunt, K.J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P.J. (1992), "Neural networks for control systems – a survey," *Automatica*, vol. 28, pp. 1083-1099.

[25] Johnson, E.F. (1967), *Automatic Process Control*, McGraw-Hill, New York.

[26] Karahan, O., Ozgen, C., Halici, U., and Leblebicioglu, K. (1997), "A nonlinear model predictive controller using neural networks," *Proc. of IEEE International Conference on Neural Networks,* Houston, USA, pp. 690-693.

[27] Karahan, O. (1997), *An Adaptive Nonlinear Model Predictive Controller Using A Neural Network*, M.Sc. Thesis, Chem. Eng. Dept., METU, Ankara, Turkey.

[28] Kirk, D.E. (1970), *Optimal Control*, Prentice-Hall, NJ.

[29] Koulouris, A. (1995), *Multiresolution Learning in Nonlinear Dynamic Process Modeling and Control*, Ph.D. Thesis, MIT.

[30] Koulouris, A. and Stephanopoulos, G. (1997), "Stability of NN-Based MPC in the presence of unbounded model uncertainty," *AIChE Symp. Series*, vol. 93, pp. 339-343.

[31] Kuzuoglu, M. and Leblebicioglu, K. (1996), "Infinite dimensional multilayer perceptions," *IEEE Trans. Neural Networks*, vol. 7, pp. 889-896.

[32] Leblebicioglu, K. and Halici, U. (1997), "Infinite dimensional radial basis function neural networks for nonlinear transformations on function spaces," *Nonlinear Analysis*, vol. 30, pp. 1649-1654.

[33] Li, S., Lim, K.Y., and Fisher, D.G. (1989), "A state-space formulation for model predictive control," *AIChE Journal*, vol. 35, pp. 241-249.

[34] Linkens, D.A. and Nyogesu, H.O. (1996), "A hierarchical multivariable fuzzy controller for learning with genetic algorithms," *Int. Journal of Control*, vol. 63, pp. 865-883.

[35] Luyben, W.L. (1990), *Process Modeling, Simulation and Control for Chemical Engineers*, McGraw-Hill.

[36] McAvoy, T.J. (1972), "Time optimal and Z-N control," *Ind. Eng. Chem. Process Res.*, vol. 11, pp. 71-78.

[37] Miller, W.T., Sutton, R.S., and Werbos, P.J. (1990), *Neural Network for Control*, MIT Press.

[38] Narendra, K.S. and Parthasarathy, K. (1990), "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, vol. 1, pp. 1-16.

[39] Nehas, E.P., Henson, M.A., and Seborg, D.E. (1992), "Nonlinear internal model control strategy for neural network models," *Comput. and Chem. Eng.*, vol. 16, pp. 1039-1057.

[40] Nesrallah, K. (1998), *Simulation Study of an Adaptive Fuzzy Knowledge Based Controller and Neural Network Controller on a pH System*, M.Sc. Thesis, Chem. Eng. Dept., METU, Ankara, Turkey.

[41] Nguyen, D.H. and Widrow, B. (1991), "Neural networks for self-training control system," *Int. J. Control*, vol. 54, pp. 1439-1451.

[42] Nie, J. (1997), "Fuzzy control of multivariable nonlinear servo-mechanisms with explicit decoupling scheme," *IEEE Trans. on Fuzzy Systems*, vol. 5, pp. 304-311.

[43] Noriega, J.R. and Wang, H. (1998), "A direct adaptive neural network control for unknown nonlinear systems and its application," *IEEE Trans. on Neural Networks*, vol. 9, pp. 27-34.

[44] Ogata, K. (1990), *Modern Control Engineering*, Prentice-Hall Inc.

[45] Palancar, M.G., Aragon, J.M., Miguens, J.A., and Torrecilla, J.S. (1996), "Application of a model reference adaptive control system to pH control: effects of lag and delay time," *Ind. Eng. Chem. Res.*, vol. 35, pp. 4100-4110.

[46] Piovoso, M., Kosanovich, K., Rohhlenko, V., and Guez, A. (1992), "A comparison of three nonlinear controller designs applied to a nonadiabatic first-order exothermic reaction in a CSTR," *Proc. Amer. Cont. Conf.*, pp. 490-494.

[47] Pottman, M. and Seborg, D. (1993), "A radial basis function control strategy and its application to a pH neutralization process," *Proc. 2nd European Cont. Conf.*, pp. 206-212.

[48] Pottman, M. and Seborg, D. (1992), "Identification of nonlinear processes using reciprocal multiquadratic functions," *J. Process Control*, vol. 2, pp. 189-203.

[49] Pottmann, M. and Seborg, D. (1992), "A nonlinear predictive control strategy based on radial basis function networks," *Proc. IFAC DYCORD Symposium*, pp. 536-544.

[50] Sablani, S.S., Ramaswamy, H.S., Sreekanth, S., and Prasher, S.O. (1997), "Neural network modeling of heat transfer to liquid particle mixtures in cans subjected to end-over-end processing," *Food Research International*, vol. 30, pp. 105-116.

[51] Saint-Donat, J., Bhat, N., and McAvoy, T.J. (1991), "Neural net based model predictive control," *Int. J. Control*, vol. 54, pp. 1453-1468.

[52] Seborg, D.E., Edgar, T.F., and Mellichamp, D.A. (1989), *Process Dynamics and Control*, Wiley Series, NY.

[53] Seborg, D. (1994), "Experience with nonlinear control and identification strategies," *Proc. Control'94*, pp. 217-225.

[54] Sing, C.H. and Postlethwaite, B. (1997), "pH control: handling nonlinearity and deadtime with fuzzy relational model-based control," *IEE Proc. Control Theory Appl.*, vol. 144, pp. 263-268.

[55] Skogestad, S. (1996), "A procedure for SISO controllability analysis – with application to design of pH neutralization processes," *Computers Chem. Eng.*, vol. 20, pp. 373-386.

[56] Skogestad, S. and Postlethwaite, I. (1997), *Multivariable Feedback Control*, John Wiley & Sons.

[57] Sorensen, E. and Skogestad, S. (1996) "Optimal startup procedures for batch distillation," *Computers and Chem. Eng.*, vol. 20, pp. 1257-1262.

[58] Stephanopoulos, G. (1984), *Chemical Process Control: An Introduction to Theory and Practice*, Prentice-Hall Int., NJ.

[59] Tan, S. (1992), "A combined PID and neural control scheme for nonlinear dynamical systems," *Proc. SICICI 92*, pp. 137-143.

[60] Tuncay, S. (1999), *Hybrid Methods in Intelligent Control*, M.Sc. Thesis, EE Eng. Dept., METU, Ankara, Turkey.

[61] Ungar, L.H., Hartman, J.E., Keeler, J.D., and Martin, G.D. (1996), "Process modeling and control using neural networks," *Int. Conf. on Intelligent Systems in Process Engineering, AIChE Symp. Series*, pp. 312-318.

[62] Utkin, V.I. (1992), *Sliding Modes in Control Optimization*, Springer-Verlag, Berlin.

[63] Wang, M. and Li, B.H. (1992), "Design of a neural network based controller for control system," *Proc. of SICICI'92*, pp. 1333-1339.

[64] Widrow, B. and Lehr, M.A. (1990), "30 years of adaptive neural networks: perception, madaline and backpropagation," *Proceedings of the IEEE*, vol. 78, pp. 1441-1457.

[65] Willis, M.J., DiMassimo, C., Montague, G.A., Tham, M.T., and Morris, A.J. (1991), "Artificial neural networks in process engineering," *IEE Proc-D*, vol. 138, pp. 3-11.

[66] Wright, A.W. and Kravaris, C. (1991), "Nonlinear control of pH processes using the strong acid equivalent," *Ind. Eng. Chem. Res.*, vol. 30, pp.1561-1572.