ELSEVIER

Journal of Process Control 13 (2002) 423–435

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Developing a robust model predictive control architecture through regional knowledge analysis of artificial neural networks

Po-Feng Tsai[a], Ji-Zheng Chu[b], Shi-Shang Jang[a,*], Shyan-Shu Shieh[c]

[a]*Chemical Engineering Department, National Tsing-Hua University, 101, Section 2 Kuqng Fu Road, Hsin-Chu, Taiwan*
[b]*Department of Automation, Beijing University of Chemical Technology, Beijing, People's Republic of China*
[c]*Department of Occupational Safety and Hygiene, Chang Jung, University, Tainan, Taiwan*

Abstract

Chemical processes are nonlinear. Model based control schemes such as model predictive control are highly related to the accuracy of the process model. For a highly nonlinear chemical system, it is clear to implement a nonlinear empirical model, such as artificial neural network model, should be superior to a linear model such as dynamic matrix model. However, unlike linear systems, the accuracy of a nonlinear empirical model strongly depends on its original data or training data based on how the model is built up. A regional-knowledge index is proposed in this study and applied in the analysis of dynamic artificial neural network models in process control. New input patterns that imply extrapolations and thus unreliable prediction by an artificial neural network model can be recognized from a significant decrease in the regional-knowledge index. To tackle the extrapolation problem and assure stability of the control system, we propose to run a neural adaptive controller in parallel with a model predictive control. A coordinator weights the outputs of these two controllers to make the final control decision. The present state of the controlled process and the model fitness to the present input pattern determine the weightings of the controller's output. The proposed analysis method and the modified model predictive control architecture have been applied to a neutralization process and excellent control performance is observed in this highly nonlinear system.

Keywords: Regional knowledge analysis; Artificial neural networks; Neural adaptive control; Model predictive control; Robust control; Neutralization process

1. Introduction

The rapid development of computing technology makes it possible to pursue better and better performance in modeling and controlling process dynamics. In order to treat high non-linearity and complex dynamics, features of a process, as expressed by some mathematical relationships, which are called a process model, have to be taken into account in the design and operation of the corresponding control system. A number of model-based control schemes have been proposed to incorporate a process model into a control system. Hussain [1] in his review of the applications of neural networks in process control categorized them in three classes: predictive control, inverse-model based control, and adaptive control. Among them, predictive control is the most commonly used. Dutta and Rhinehart [2] gave a brief comparison of various model-based control schemes.

Instigated by the great success of dynamic matrix control (DMC) [3,4], the theory of model predictive control (MPC) has been receiving intensive attention in the process control area [5]. The importance of MPC is that it provides a general control scheme in which material and/or energy conversion equipments and the control devices can be considered as a whole in the design of control systems. The basic idea of MPC is to use a model to predict the future output trajectory of a process and compute a series of controller actions to minimize the difference between the predicted trajectory and a user-specified one, subject to constraints [6]. MPC works in a centralized manner for multivariable constrained control problems through a multivariable minimization, which provides a natural decoupling measure essential for control of highly interactive systems.

It is clear that MPC demands a dynamic process model of proper accuracy and execution speed, though the feedback mechanism of MPC tolerates some model

* Corresponding author. Tel.: +886-3-5735294; fax: +886-3-5715408.
E-mail address: ssjang@che.nthu.edu.tw (S.-S. Jang).

0959-1524/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
PII: S0959-1524(02)00067-7

mismatch. In the conventional dynamic matrix control which can be seen as linear model predictive control (LMPC), process outputs in the prediction horizon are expressed as a linear function of the input and output itself in the past, and coefficients in the model are determined from time series response data of the target process. Such a linear model is a perfect description for linear systems, and is a local approximation for nonlinear systems. However, the region of acceptable linear approximation for systems with high non-linearity will become very small, and such a linear model will not justify the efforts needed to build it.

Therefore, linear model predictive control such as DMC will work well only for slightly nonlinear or slowly responding processes [7]. For highly nonlinear processes, nonlinear models are necessary, and nonlinear model predictive control (NMPC) strategy has been studied [8–10].

In the implementation of nonlinear model predictive control [8–10], developing a valid dynamic model is the major theme of the work. Artificial neural networks (ANNs) as a process model for control purpose are superior to other conventional modeling methods, as pointed out by several authors [1,11].

(1) Models derived from first principles are usually difficult and/or costly to develop for processes, which are not well understood or very complex. Further, in order to evaluate model parameters and to make models concise enough for online execution, inevitable assumptions and simplifications included in this kind of models may cause a severe decrease in the model accuracy.

(2) Artificial neural networks provide a general approach for extracting process dynamics from input-output data only. Their learning ability makes them versatile and friendly for practical applications. With their great power for approximating complex functionality [12], their compact form and great speed of information retrieval make them highly suitable for online uses.

In a recent review, Hussain [1] summarized the active research on the application of artificial neural networks in model-based control design, and he noted that ANNs have almost become the standard empirical models for the purpose of nonlinear process control [13–15].

Because of their empirical characteristics, ANN models need to be trained with a lot of operation data to cover certain operating ranges of process. Uncertainty of an ANN model often exists and may be severe for some special ranges. For instance, in a range around the equivalence point of neutralization where the process output (pH) is highly sensitive to the manipulated variable (flow rate of acid or base stream), training data is usually scarce, and an ANN model would be hence very rough.

Uncertainty in the employed model will degrade MPC and severe errors may cause it to fail completely. When artificial neural networks are used as a process model, unknown input patterns often occur, which makes the future prediction unreliable and thus causes unstable control performance. Such a situation almost surely exists for highly sensitive and highly nonlinear processes because training patterns are hardly complete.

Since model uncertainty is inevitable, the following two points are essential to guarantee the performance of MPC. (1) Identify the uncertainty of the model; (2) Increase the robustness of MPC to cope with the model's uncertainty. In the work of Lin and Jang [16], a systematic approach based on information theory was presented for designing the data set used to train an ANN for the purpose of a complete process model. However, implementation of such designs in industrial circumstances may be very expensive and even impossible. By extending radial basis function networks (RBFN), Leonard et al. [17] proposed a "validity index network" that computes the reliability and confidence of its own output and indicates local regions of poor fit and extrapolation. Their idea of using Parzen's estimator to calculate the probability density of training data is universal for other kind of empirical models. To accommodate the uncertainty of the model used in MPC, additional mechanism is necessary, and the neural adaptive controller (NAC) of Krishnapura and Jutan [18] is worthy of consideration.

The concept of neural adaptive control is simply from the idea that the back propagation mechanism of training of an artificial neural network can be served as a general approach for dynamic programming. Psaltis et al. [19] for the first time proposed a so-called specialized learning architecture in which an artificial neural network will be trained to learn for specific outputs through a modified back propagation algorithm. Nguyen et al. [20] proposed a similar algorithm called self-learning control system with a neural network controller and a neural network model that propagates back the error. Loh et al. [21] modified it as a model reference artificial neural net control strategy and applied it to a pH control system. Krishnapura and Jutan [18] reduces the large number of the hidden layer nodes to a single one and thus eliminates a lot of connection weights that is estimated for on-line adaptations.

A neural adaptive controller (NAC) can be thought of as an auto-tuning feedback controller without any model requirement, and it provides a strong feedback mechanism through its localized learning algorithm and thus has a strong power to stabilize a process. We can adjust the step size of the weight adaptation by setting the learning rate. If the system responds slowly or does not possess strong non-linearity, the neural adaptive controller will have enough time for adapting and learning so that the control performance is good. But if the system responds fast and has high non-linearity, such as in the case of pH control systems, the neural adaptive controller will need much more time than usual to learn the path toward the set point and

may cause the whole system to oscillate. It also becomes clumsy for frequent or large disturbances/set-point changes just because of its local adaptation mechanism.

From the features of the neural adaptive control discussed above, we believe that the neural adaptive control and the conventional model predictive control can be a complement to each other. MPC can be used to provide global guidance to NAC, while NAC can be used as the backup to MPC when the model is not reliable around some operation regions. Of course, some certain coordination is needed to make them work harmonically. We determine the model performance at every time step with the application of probability density function, which shows the relative operating regional knowledge as a guiding index to coordinate the two controllers. This is the main idea in designing the robust model predictive control architecture proposed in this study.

For demonstrating the effectiveness of the methods proposed in this study, a neutralization process is chosen as the target system. The pH control system is a challenging control problem due to its high non-linearity and sensitivity and also is a widely used process in industrial practice. The equivalence point and the titration curve depend on the compounds. This problem has been benchmarked in many control schemes such as various PID control strategies [22,23] and linear and nonlinear model-based techniques [24,25]. Palancar et al. [26] proposed a rather complicated control structure that was similar to the conventional MPC and included two artificial neural networks to model the process and the inverse of the process respectively. In the work of Syu [27], a recurrent neural network model was applied in the control of the pH environment in a penicillin reactor.

The rest of this paper is organized in five sections. Section 2 presents the regional knowledge analysis of artificial neural network model. In Section 3, the proposed robust model predictive control architecture is introduced, and testing results of the pattern analysis method and the new control architecture on a neutralization process are demonstrated in Section 4. Concluding remarks are finally made in Section 5.

2. Regional knowledge analysis of artificial neural network models

In this paper, a feed-forward artificial neural network (FFN) model as depicted in Fig. 1, is implemented to control a neutralization process. Note that the following analysis is general and thus not limited to the case of FFN. For a single input and single output (SISO) dynamic system, any dynamic model, can be expressed in the following form:

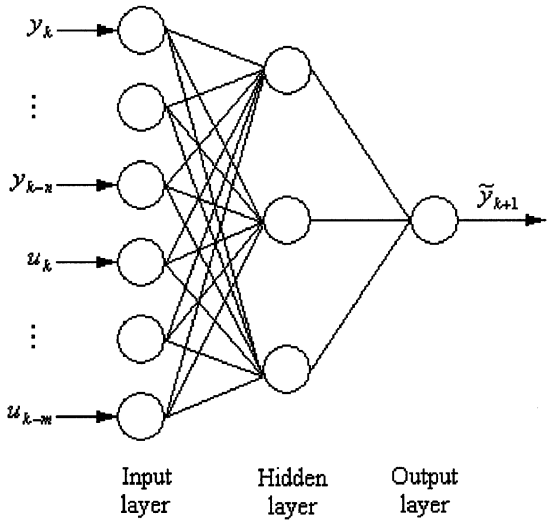

Fig. 1. A feed-forward neural network as the process model.

$$\tilde{y}_{k+1} = f(y_k, y_{k-1}, \ldots, y_{k-n}, u_k, u_{k-1}, \ldots, u_{k-m}) \quad k = 0, 1, 2, \ldots \tag{1}$$

where u and y are the input and the measured output, $?$ is the predicted output, k stands for the current time instant, and n and m are the output and input orders.

For convenience of statement, we define

$$\omega = (y_k, y_{k-1}, \ldots, y_{k-n}, u_k, u_{k-1}, \ldots, u_{k-m}) \tag{2}$$

and ω is called an event in the dynamic space and is a general form of the input pattern to the artificial neural network model. Because y_{k+1} is uniquely determined by the real process, we have the corresponding augmented event as follows:

$$\theta = (y_{k+1}, \omega) = (y_{k+1}, y_k, y_{k-1}, \ldots, y_{k-n}, u_k, u_{k-1}, \ldots, u_{k-m}) \tag{3}$$

Assume that the following training data set are used in building the above neural network model:

$$\Theta = \{\theta_i = (y^i_{k+1}, y^i_k, y^i_{k-1}, \ldots, y^i_{k-n}, u^i_k, u^i_{k-1}, \ldots, u^i_{k-m})|_{i=1,\ldots,N}\} \tag{4}$$

And the corresponding set of input patterns is

$$\Omega = \{\omega_i = (y^i_k, y^i_{k-1}, \ldots, y^i_{k-n}, u^i_k, u^i_{k-1}, \ldots, u^i_{k-m})|_{i=1,\ldots,N}\} \tag{5}$$

Our problem is how to know whether an input pattern is included in the training data set or not. On the

other hand, for a highly dimensional input event, we have to estimate how much the system knowledge we have around the neighborhood of the input event. If there are many training data points in this region, it means the event sits in a well-explored area and the model is suitable for predicting the outcome of the event. Otherwise, if the input event sits in an unfamiliar region, which has only few data points in it, the model prediction will very likely fail when it is highly nonlinear. This problem is meaningful because an artificial neural network model is usually assumed to be reliable only when interpolation among learnt patterns is performed. In deriving a criterion for the above judgment, the concept of Parzen–Rosenblatt probability density function [28] is used and extended as an index to measure the reliability of the model prediction.

The Parzen–Rosenblatt density estimate of a new event, ω_{new}, based on the training data set, Ω, is defined as:

$$f_{\Omega}(\omega_{\text{new}}) = \frac{1}{Nh^{m_0}} \sum_{i=1}^{N} K\left(\frac{\omega_{\text{new}} - \omega_i}{h}\right) \tag{6}$$

where the smoothing parameter, h, is a positive number called bandwidth or simply width, which controls the span size of the kernel function, $K\left(\frac{\omega_{\text{new}}-\omega_i}{h}\right)$ and m_0 is the dimensionality of the event set, Ω. The kernel functions, K, are various and, however, both theoretical and practical considerations limit the choice. A well-known and widely used kernel is the multivariate Gaussian distribution:

$$K\left(\frac{\omega_{\text{new}} - \omega_i}{h}\right) = \frac{1}{(2\pi h^2)^{m_0/2}} \exp\left(\frac{\|\omega_{\text{new}} - \omega_i\|^2}{2h^2}\right) \tag{7}$$

Parzen-Rosenblatt probability density function is based on the distances between the new event, ω_{new}, and the events, ω_i, of training data set, Ω, through the kernel functions. Once ω_{new} is close to some ω_i, the relative kernel functions will give higher values and those ω_i which are not in the neighborhood will give lower values in the summation. The above probability density function (6) is denoted as *regional knowledge index* of each new event occurred to the process and every training data point is involved in calculating it. The kernel function plays a role just like a membership function of distance. In other words, if a region is crowded with data points, the "density" will be high, and it also implies that we probably have enough knowledge about this region. The role of regional knowledge index in the proposed coordinator will be thoroughly discussed in Section 3.3.

The above treatment on the reliability of empirical models, direct use of Parzen–Rosenblatt probability density over all the existing training data, provides a concise approach for control applications. Such a density is virtually the same as used by Leonard et al. [17], except that they calculate the density of an event by weight-averaging the densities of the cluster centers obtained in training radial basis function networks (RBFN). As a matter of fact, the weight averaging approach will be reduced to the direct use of Parzen–Rosenblatt probability density if every point in the training set is taken as a center and the same activation function is used for each center, which is the idea of general regression neural network (GRNN). The density of any new event will locate among the values at the existing centers due to the interpolation [17], which requires a careful decision on the number and distribution of the centers in order to guarantee a true representation of the centers to all the existing data points in the training set.

3. Architecture for robust model predictive control

In the introduction of this paper, a brief analysis is presented about the features of the conventional model predictive control and the neural adaptive control. Serious uncertainty of the used model will endanger the stability of model predictive control. In such cases, additional or backup tuning measures are necessary, and the neural adaptive controller is suggested because its localized learning algorithm provides a model-free auto-tuning feedback mechanism. Based upon this analysis, a new architecture called robust model predictive control (RMPC) is proposed as a modification to the conventional MPC and is illustrated in Fig. 2. The proposed RMPC is composed of three parts: (1) a model predictive control scheme; (2) a neural adaptive controller (NAC) running in parallel with the MPC, and (3) a coordinator to decide the final control action according to the outputs of the parallel NAC and MPC and the fitness of the model used. In the following context, the three elements of RMPC are introduced.

3.1. Model predictive control

The proposed robust model predictive control shown in Fig. 2 is reduced to a standard model predictive control if we set $u = u_{\text{MPC}}$ in the coordinator. The optimizer performs the following constrained minimization problem:

$$\underset{u_{k+1},u_{k+2},\ldots,u_{k+P}}{\text{Min}} \sum_{j=1}^{P} \varphi_j\left(\tilde{y}_{k+j} - y_{d,k+j}\right)^2 + \gamma_j \Delta u_{k+j}^2 \tag{9}$$

subject to

$$u_{\min} \leqslant u_{k+j} \leqslant u_{\max} \quad j = 1, 2, \ldots, P \tag{10a}$$

$$\left|\Delta u_{k+j}\right| \leqslant \Delta u_{\max} \quad j = 1, 2, \ldots, P \tag{10b}$$

where y_d is the set point value, P is the length of the prediction horizon, and φ and γ are weights and are set

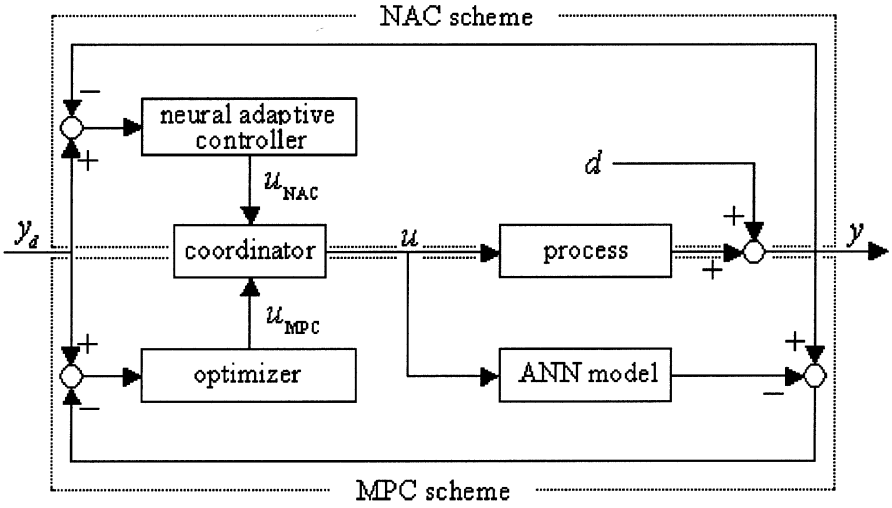

Fig. 2. Architecture of the robust model predictive control (RMPC).

to be unit in this study. The optimal control sequence $\{u_{k+1}, u_{k+2}, \ldots, u_{k+P}\}$ can be found with a multivariate optimization procedure.

3.2. Neural adaptive control

If we set $u = u_{\mathrm{NAC}}$ in the coordinator, Fig. 2 is reduced to the neural adaptive control by Krishnapura and Jutan [18] with a detailed structure shown in Fig. 3. The neural adaptive controller has three inputs, the set point $y_{d,}$ the past value of the output y, and the control action u. There are two nonlinear nodes with sigmoid activation function in the hidden layer and the output layer, respectively, and four adjustable weights (black dots in Fig. 3) for the three inputs and the one output of the hidden node. Such a controller can be expressed as an augmented network by including the process as an unchangeable node.

The whole system works by updating all the four connecting weights in the network to minimize the deviation (E) of the process output from its set-point value at current time instant k:

$$E_k = \frac{1}{2}\left(y_{d,k} - y_k\right)^2 \tag{11}$$

This error signal is generated at the output of the plant and is passed backward to the neural network controller through the plant and is minimized with the steepest descent method according to the following equation:

$$W_{ij,k}^{(l)} = W_{ij,k-1}^{(l)} - \alpha\left(\frac{\partial E}{\partial W_{ij}^{(l)}}\right)_k \tag{12}$$

where $W_{ij}^{(1)}$ is the weight connecting node i in the upstream layer l and node j in the downstream layer $l+1$ ($l=1$, 2 and 3 for the input, hidden, and output layers respectively), and α is a constant denoting the

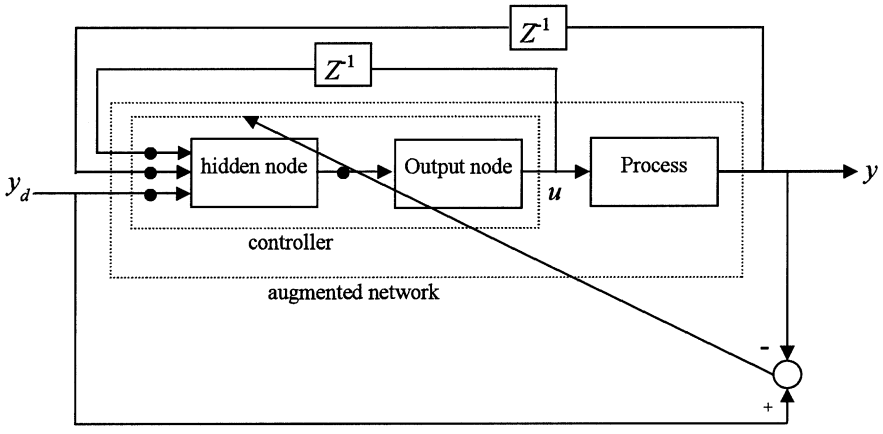

Fig. 3. Neural adaptive controller structure.

learning rate of the network. The derivative can be formulated by applying the chain rule of differentiation:

$$\frac{\partial E}{\partial W_{ij}^{(l)}} = -(y_d - y) I_i^{(l)} \left(\frac{\mathrm{d}f_j^{(l+2)}}{\mathrm{d}x}\right) \times \left(\frac{\partial y}{\partial u}\right) \sum_r W_{jr}^{(l+1)} \left(\frac{\mathrm{d}f_r^{(l+1)}}{\mathrm{d}x}\right) \quad (l = 1, 2) \tag{13}$$

where $I_i^{(1)}$ is the input to node i in layer l, $\left(\frac{\mathrm{d}f_i^{(l)}}{\mathrm{d}x}\right)$ is derivative of activation function of node i in layer l with respect to its only input x. It is not easy to estimate the Jacobian of the process, $\frac{\partial y}{\partial u}$ in Eq. (13) at every sampling instant. Psaltis et al. [29] used an iterative approach to evaluate it. Krishnapura and Jutan [18] proposed the use of a sigmoid function to approximate the process input-output gain information. Nguyen et al. [20] constructed an ANN model for the same purpose. Yang et al. [30] proposed a so-called online adaptive neural-network-based controller (OANNC) to implicitly estimate the Jacobian values. As a matter of fact, $\frac{\partial y}{\partial u}$ can be thought of as a scaling factor, which decides the direction (sign of $\frac{\partial y}{\partial u}$) and magnitude $\left(\left\|\frac{\partial y}{\partial u}\right\|\right)$ of the gradient vector. In the training phase with steepest decent optimization, the error term $(y_d - y)$ in Eq. (13) is useful in speeding up the minimization process and in making a more flexible learning rate. The learning rate α is usually fixed and a small value of it is used to perform a fixed-step-size evolution whose direction is determined by the normalized gradient. Thus, magnitude part, $\left\|\frac{\partial y}{\partial u}\right\|$ of $\frac{\partial y}{\partial u}$ can be ignored, since α and $(y_d - y)$ are enough to determine the overall learning rate of the neural adaptive network. In this study, we propose that α be tuned based on appropriate technique such as IAE.

3.3. Coordinator

In the proposed architecture as shown in Fig. 2, the model predictive controller and the neural adaptive controller run in parallel. A coordinator is designed to make the final decision based on the outputs of the above two controllers. Obviously, knowledge about the current state of the process and the accuracy of the ANN model used by the MPC is necessary to make a wise decision.

As a preliminary test, the following equation is used to combine the outputs of the MPC and the NAC:

$$u = \psi u_{\mathrm{MPC}} + (1 - \psi) u_{\mathrm{MPC}} \tag{14}$$

where ψ is a decision factor with the following properties:

1. The decision factor ψ is a positive number between 0 and 1.
2. ψ is a model-reliability index that weights the control actions from model predictive controller and neural adaptive controller. If the model is built with perfect predicting precision, ψ is taken to be 1, otherwise, it decreases with the precision.
3. The model predictive controller may have different performance in different operating area. So the value of decision factor is changing with the operating conditions.
4. The decision factor is thus determined by the regional knowledge index in Eq. (6). The higher values of regional knowledge index means the region around the input event is well explored and implying we have more knowledge about it so that higher ψ is suggested. In the other words, ψ is a function of region-knowledge index, $\hat{f}_\Omega(\omega_{\mathrm{new}})$.

For simplicity, the following linear form denotes the decision factor ψ is implemented in this work:

$$\psi = \Psi\left(\hat{f}_\Omega(\omega_{\mathrm{new}})\right) = \frac{1}{b-a}\hat{f}_\Omega(\omega_{\mathrm{new}}) - \frac{a}{b-a} \quad \text{for } a < \hat{f}_\Omega(\omega_{\mathrm{new}}) \leqslant b \tag{15}$$

where a and b are constants, and $\hat{f}_\Omega(\omega_{\mathrm{new}}) \leqslant a$, $\psi = 0$ and $\hat{f}_\Omega(\omega_{\mathrm{new}}) > b$, $\psi = 1$ as shown in Fig. 4.

4. pH control

In this part, the proportional-integral (PI) controller tuned by classical one-quarter decay method and by the internal model control (IMC) algorithm of Chien et al. [31], the conventional model predictive controller (MPC), the neural adaptive controller (NAC), and the proposed robust model predictive control (RMPC) are used in a simulated neutralization process, and comparison results are presented to show the great superiority of RMPC over all its counterparts.

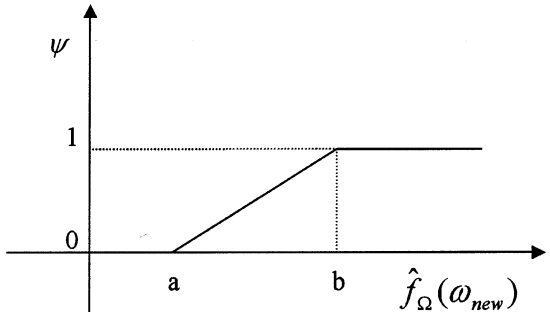

Fig. 4. ψ vs. $\hat{f}_\Omega(\omega_{\mathrm{new}})$.

4.1. pH control system

The simulated pH control system is adopted from Palancar [26] and is shown in Fig. 5. There are two inlet streams to the continuous stirred tank of reaction (CSTR), the acid flow, Q_A, an aqueous solution of acetic acid and propionic acid, and the base flow, Q_B, an aqueous solution of sodium hydroxide. The outlet stream is Q.

In this single input and single output system, the manipulated variable is the base stream flow rate and controlled variable is the pH value of the system. The titration curve for this system is shown in Fig. 6 which shows an equivalence point around pH = 8.9.

The neutralization reactions are as follows:

$$\mathrm{AcH} \Longleftrightarrow \mathrm{Ac^-} + \mathrm{H^+} \tag{17}$$

$$\mathrm{PrH} \Longleftrightarrow \mathrm{Pr^-} + \mathrm{H^+} \tag{18}$$

$$\mathrm{NaOH} \Longleftrightarrow \mathrm{OH^-} + \mathrm{Na^+} \tag{19}$$

$$\mathrm{H_2O} \Longleftrightarrow \mathrm{OH^-} + \mathrm{H^+} \tag{20}$$

The reactions take place in the aqueous solution and the concentrations inside the CSTR are easily calculated by material balance equations. Those equations also provide the dynamic transient states of the system and they are:

$$Q_A C_{0\mathrm{PrH}} = QC_{\mathrm{PrH}} + V\frac{\mathrm{d}C_{\mathrm{PrH}}}{\mathrm{d}t} \tag{21}$$

$$Q_A C_{0\mathrm{AcH}} = QC_{\mathrm{AcH}} + V\frac{\mathrm{d}C_{\mathrm{AcH}}}{\mathrm{d}t} \tag{22}$$

$$Q_B C_{0\mathrm{NaOH}} = QC_{\mathrm{NaOH}} + V\frac{\mathrm{d}C_{\mathrm{NaOH}}}{\mathrm{d}t} \tag{23}$$

where C_{AcH}, C_{PrH} and C_{NaOH} are concentrations of components AcH, PrH and NaOH, V is the volume of the reactor. The pH value is calculated with the compositions and the dissociation constants by the following formula:

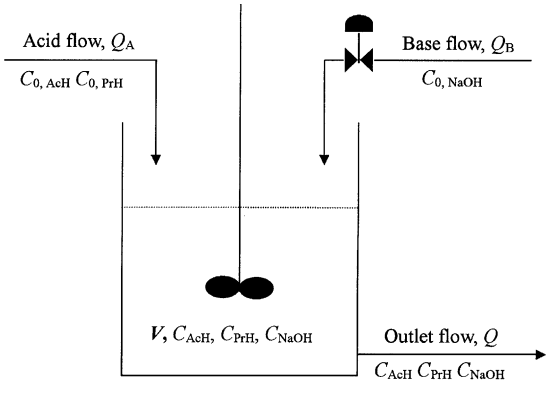

Fig. 5. Diagram of the simulated neutralization CSTR.

$$\frac{C_{\mathrm{AcH}}}{1+\dfrac{10^{-\mathrm{pH}}}{K_{\mathrm{AcH}}}} + \frac{C_{\mathrm{PrH}}}{1+\dfrac{10^{-\mathrm{pH}}}{K_{\mathrm{PrH}}}} + 10^{(\mathrm{pH}-14)} = C_{\mathrm{NaOH}} + 10^{-\mathrm{pH}} \tag{24}$$

$$pK_{\mathrm{AcH}} = 4.75 \tag{25}$$

$$\mathrm{p}K_{\mathrm{PrH}} = 4.87 \tag{26}$$

$$\mathrm{pH} = -\log_{10}[H^+] \tag{27}$$

The system's initial conditions are listed in Table 1.

4.2. Data for training the ANN model

The data for training the ANN model used in the conventional MPC, and the proposed RMPC is generated by changing the set-point values to the PI controller, according to a five-cell pseudo-random binary sequence (PRBS). The PRBS includes 2^5-1 = 31 signal patterns. The sampling and the control actions take place every 10 s. The training data thus obtained are shown in Fig. 7.

4.3. Artificial neural network model

A three-layered feed-forward neural network with 6 inputs as shown in Fig. 1 is employed to build a model for MPC and RMPC. The inputs are the base flow rates and pH values at the current time instant, k, and the previous two time instants, $k-1$ and $k-2$. Since a simple structured ANN model may not be able to work well in the pH system as pointed out by Wang et al. [32], eight nodes are used in the hidden layer to catch the strong non-linearity and the output is the predicted pH value at the next time instant, $k+1$.

The training set consists of 320 data points while the testing set 150 data points. The training and testing results are illustrated in Figs. 8 and 9. If we consider that the predicted points falling beyond the range of ± 1 of real pH value are outliers, then 4 points out of 320 points are outliers in the training set while 40 out of 150 in the testing set. In other words, the outlier percentages for the training and testing set are 1.25 and 26.67% respectively. These two figures show that good agreement between the "experimental" data and the predicted ones of the ANN model for most data in the training set, but not so for the testing set. Most of these outliers take pH values near to the equivalence point of this neutralization system. The data points sitting in the region from pH 7 to pH 11 are very few and most data points crowd in the regions lower than pH 7 and higher than pH 11. The region close to the equivalence point is very poorly explored and the FFN model is also failed here.

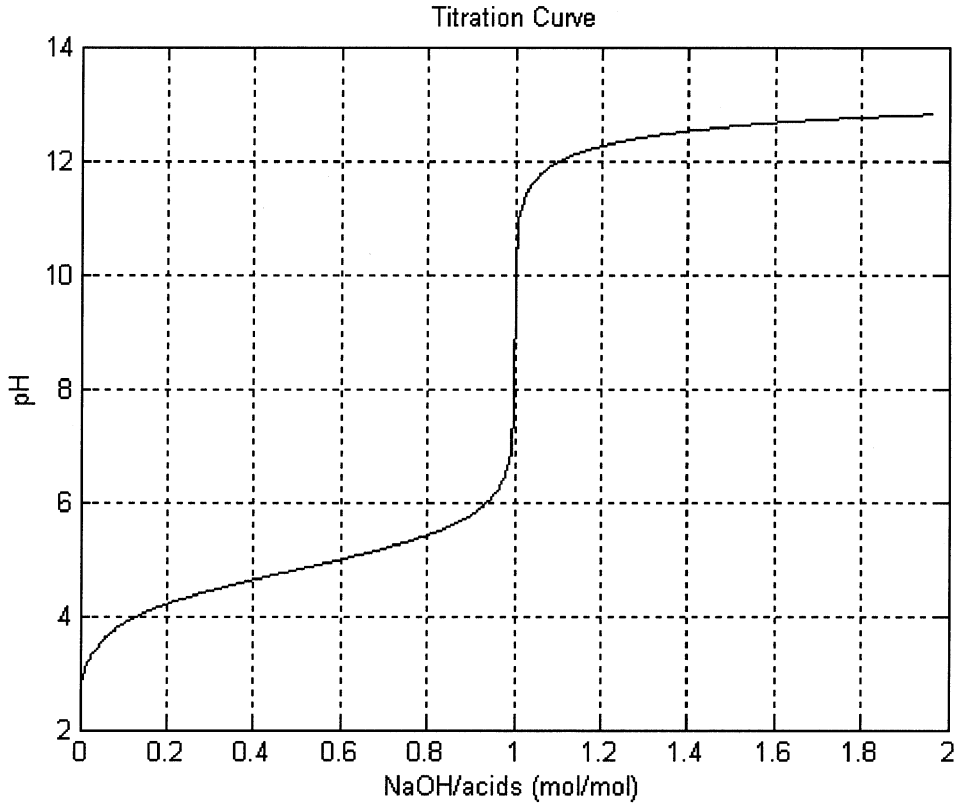

Fig. 6. Titration curve of pH simulation.

4.4. Comparison of the simulation results

The neural adaptive controller used has three inputs:$Q_{B,k-1}$, pH_k and pH_{k-1}, and is initiated by running it for some time at the steady state of the process. In tuning the PI controller by the one-quarter decay ratio response, the ultimate gain and period of the process are found to be $K_{cu} \cong 750$ and $T_u \cong 30$. The ANN model as stated above is used in the conventional MPC and the proposed RMPC. The prediction horizon for both MPC and RMPC is one time step ahead.

The setup of RMPC include the bandwidth (h) in Eq. (6) and parameters a and b in Eq. (15). As pointed out by Leonard et al. [17], the bandwidth is crucial for the discussion of a "local" error measure for a model. If h is too small, the neighborhood considered to be local to the test point will not contain enough data to estimate the model accuracy and the test for extrapolation, namely lack of data, will be overly sensitive. If h is too large, the local variation in accuracy will be missed and the test for extrapolation will be insensitive. Specht [33] has suggested that the bandwidth be selected between 0.1 and 0.3. For the studied pH control system, the bandwidth is set to be 0.1 according to a test result shown in Fig. 10. In the test, the IAE was recorded for a step change from pH 7 to 10 at each selected value of the bandwidth. It is clear from this figure that bigger bandwidth causes smoother and wider density distribution and that every point is taken as well known after bandwidth being greater than 0.3. Then the MPC dominates the control actions and the NAC stops working. However, for the bandwidth lower than 0.05, the NAC takes over the control priority. Therefore, the IAE will only change when the bandwidth value lies between 0.05 and 0.3. A minimum occurs at 0.1, which is thus used as our bandwidth. Three cases have been studied and the results are listed separately as following subsections. Parameters a and b in Eq. (15) can be easily determined by observing the sharp drops in RKI when extrapolation occurs, as will be seen in the regional knowledge index (RKI) plots in the following examples.

Table 1
Initial states of the pH control system

pH	8.9
$C_{0,\,AcH}$	0.1 mol/l
$C_{0,\,PrH}$	0.1 mol/l
$C_{0,\,NaOH}$	0.2 mol/l
Q_A	0.003 l/s
Q_B	0.003 l/s
Q	0.006 l/s
V	1.75 l

4.4.1. Case 1. Step change in set point

In this case, the above five control schemes are tested against a step change of the set point from pH = 7 up to

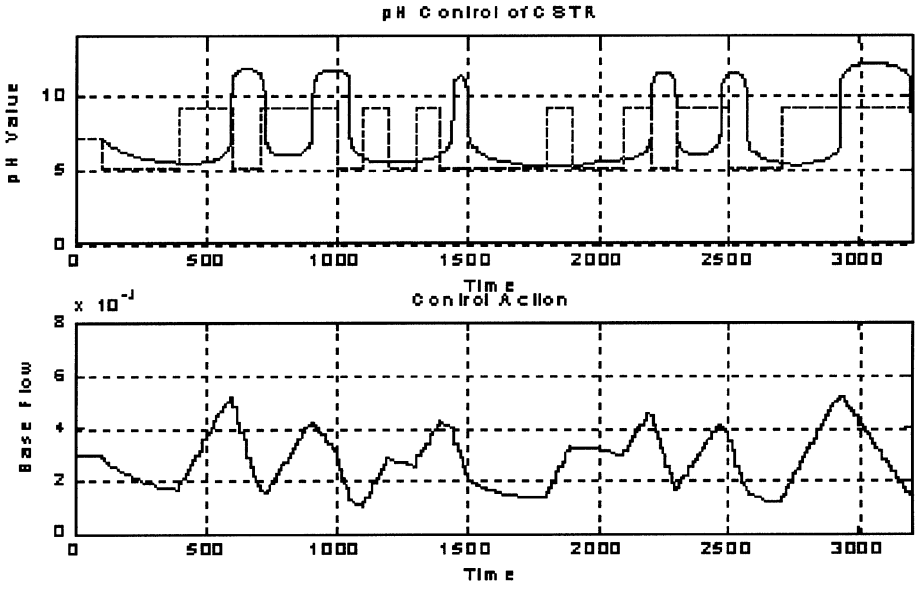

Fig. 7. Plots of training data (solid line: response; dotted line: set point).

pH = 10 at the 700th s. The parallel running results are shown in Figs. 11–15 for the five controllers. Figs. 11 and 12 show that the PI controllers tuned by the one-quarter decay method and by the internal model control (IMC) algorithm ($\tau_{cl} = 500$) do not work, though the IMC-tuned PI performs better at big τ_{cl} than the PI tuned by the one-quarter decay method. The MPC does not work either in this system as shown in Fig. 13. The neural adaptive controller works well if its learning rate parameter α is well tuned ($\alpha = 0.1$ for this particular example), as shown in Fig. 14, though long time is needed for it to compensate the error gradually in a feedback manner. The proposed RMPC behaves excellently as illustrated in Fig. 15.

It should be noted that as shown in Fig. 15, the regional knowledge index, the probability density function decreases abruptly around the equivalence point. This is due to few experimental data around this point as shown in Figs. 8 and 9. The decrease of the regional knowledge index implies that the system is moving to a poorly known region where the model is unreliable and the weighting for MPC is reduced by the coordinator. In order to show how the coordinator works, we zoom in the area around the 700th s where the set point changes from pH 7 to pH 10 in Fig. 16. The MPC suggests a much more aggressive control action than NAC does and the coordinator combines them according to the regional knowledge index, which leads to a mild control action between them.

4.4.2. Case 2. A sequence of step changes in set point

The surviving two controllers in case 1, namely NAC and the proposed RMPC are further tested against a sequence of step changes in set point, and the results

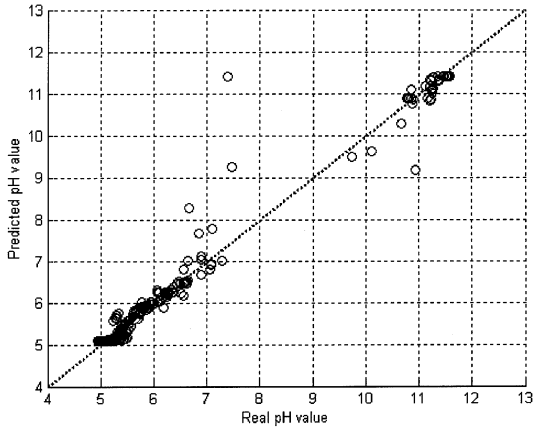

Fig. 8. Training result of the ANN model.

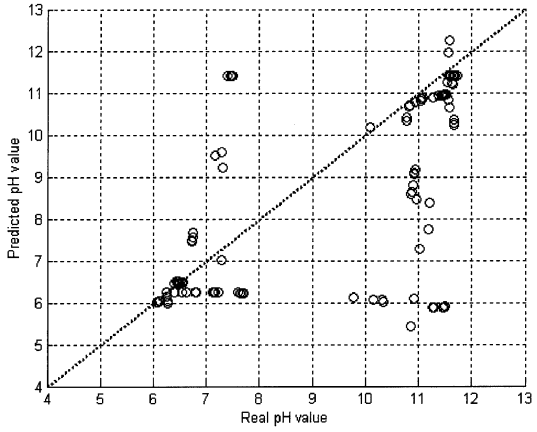

Fig. 9. Testing result of the ANN model.

are depicted in Figs. 17 and 18. As expected, NAC deteriorates rapidly when facing successive fast changes of large magnitudes, because its feedback mechanism needs long enough time to bring the system to a steady state. Contrary, RMPC works well even at the sharp equivalence point of neutralization.

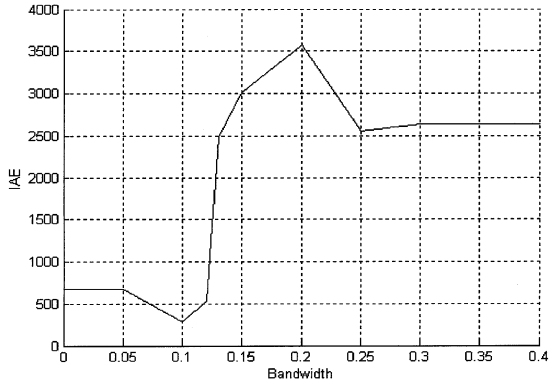

Fig. 10. Results of the test for determining the bandwidth in Eq. (6).

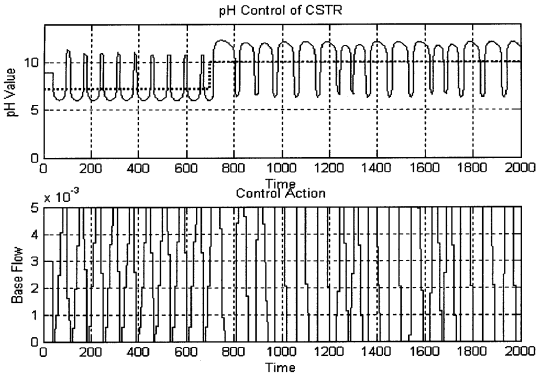

Fig. 11. PI control tuned by 1/4 decay method against a step change in set point from pH = 7 to 10 (solid line: response; dotted line: set point).

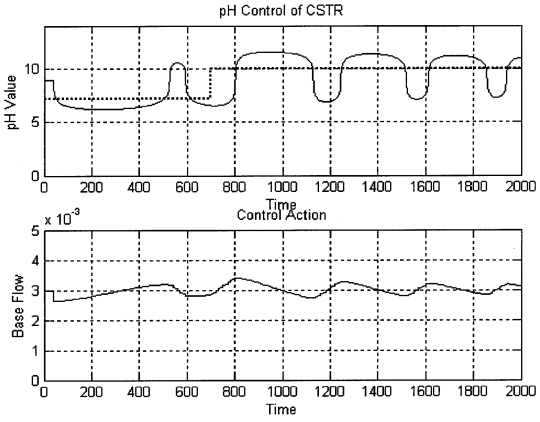

Fig. 12. PI control tuned by the IMC algorithm ($\tau_{cl} = 500$) against a step change in set point from pH = 7 to 10 (solid line: response; dotted line: set point).

4.4.3. Case 3. Disturbance in the acidic stream flow rate

In order to evaluate the disturbance-proof capability of the proposed RMPC, it is tested against a 20% reduction in the unmeasured acidic stream flow rate at the 500th s. Parallel tests are carried out for NAC and the conventional MPC. The testing results are shown in Figs. 19–21.

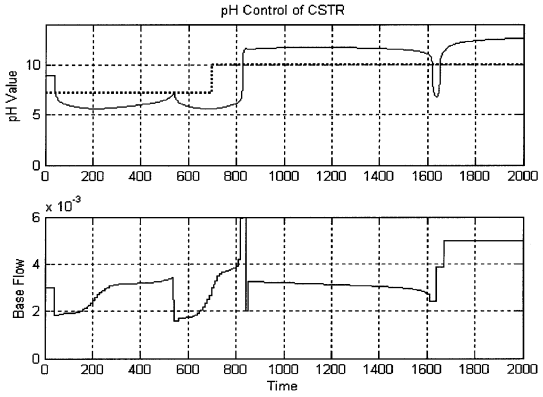

Fig. 13. MPC against a step change in set point from pH = 7 to 10 (solid line: response; dotted line: set point).

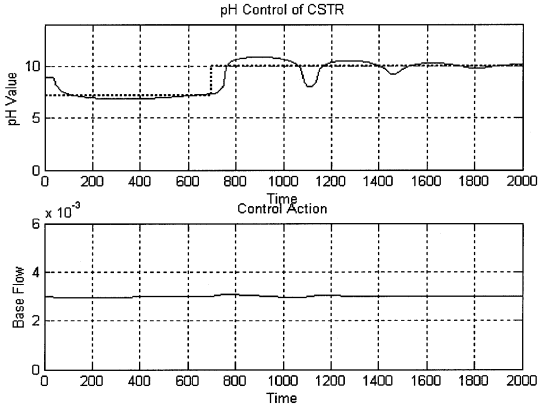

Fig. 14. NAC against a step change in set point from pH = 7 to 10 (solid line: response; dotted line: set point).

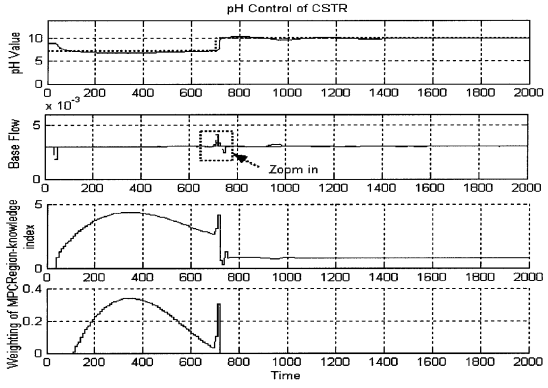

Fig. 15. RMPC against a step change in set point from pH = 7 to 10 (solid line: response; dotted line: set point).

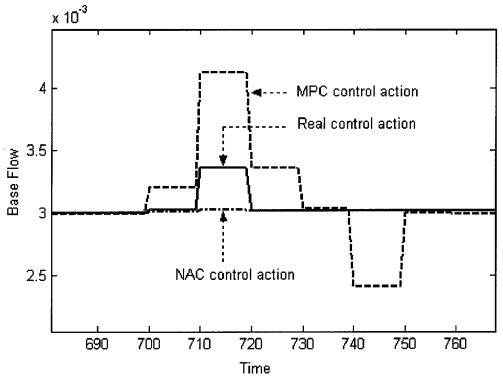

Fig. 16. Zoom-in of the control actions.

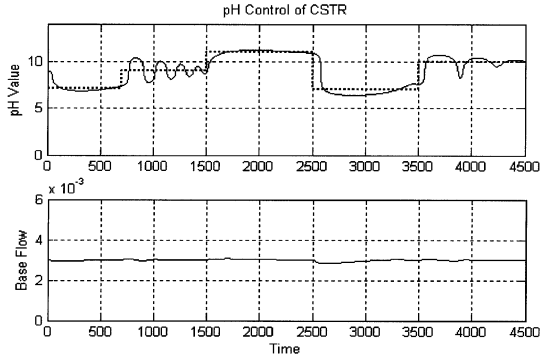

Fig. 17. NAC against a sequence of step changes in set point (solid line: response; dotted line: set point).

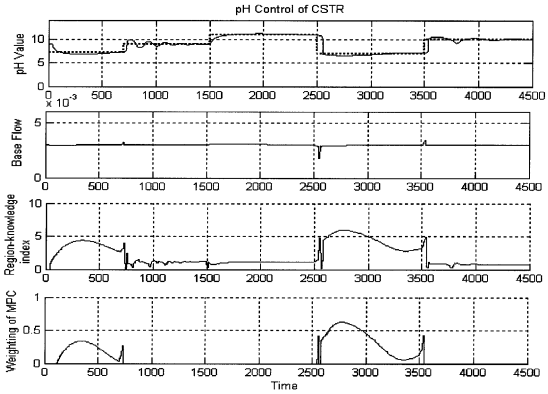

Fig. 18. RMPC against a sequence of step changes in set point (solid line: response; dotted line: set point).

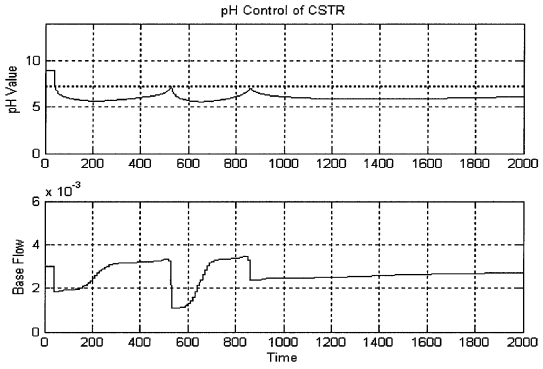

Fig. 19. MPC against a disturbance in the acidic stream flow rate (solid line: response; dotted line: set point).

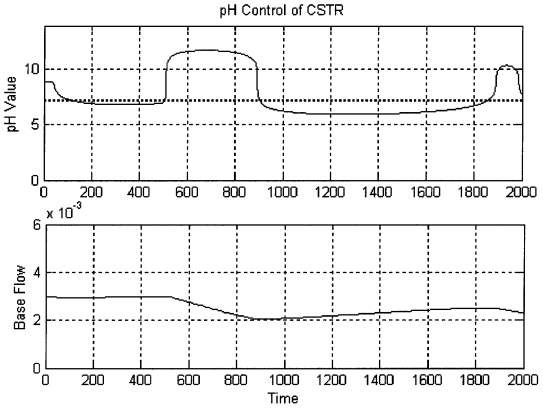

Fig. 20. NAC against a disturbance in the acidic stream flow rate (solid line: response; dotted line: set point).

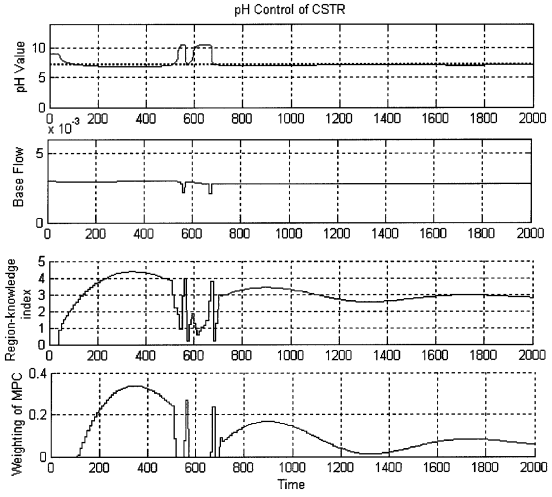

Fig. 21. RMPC against a disturbance in the acidic stream flow rate (solid line: response; dotted line: set point).

Fig. 19 shows that MPC results in an oscillatory system. Fig. 20 tells us that NAC can resist the disturbance after a long time of regulation. The excellence of RMPC is clearly revealed in Fig. 21 where fast and stable control performance is observed.

5. Conclusion

Artificial neural network model predictive control has been an active and important research topic because model predictive control provides a promising and general architecture to treat complex control problems and artificial neural networks are a general approach for industrial modeling. However, the incompleteness and inaccuracy of artificial neural network models generally exist and deteriorate the performance of the above control scheme. For highly nonlinear and/or sensitive processes, the deterioration is so bad that stable control is impossible. In solving this problem, regional knowledge analysis is proposed in this study and applied to analyze artificial neural network models in process control. New input patterns, which mean extrapolation and, thus unreliable prediction by an artificial neural network, can be recognized from steep decrease in the probability density function, which can be calculated from the training data set. In case that the predictive control is unable to achieve stable and smooth control, the conventional model predictive control is modified by incorporating a parallel-running neural adaptive controller to provide sufficient stability. A coordinator regulates the weights of the outputs from the model predictive controller and the neural adaptive controller to make the final control decision. The regional knowledge index in the coordinator determines the weights by considering the present state of the controlled processes and the model fitness to the present input pattern. The proposed analysis method and the modified model predictive control architecture are applied to a neutralization process. Excellent control performance is observed in this highly nonlinear and sensitive system.

Acknowledgements

The Authors thank the financial support from National Science Council through the grant NSC90–2622-E007–003

References

[1] M.A. Hussain, Review of the application of neural networks in chemical process control—simulation and online implementation, Artificial Intelligence in Engineering 13 (1999) 55–68.
[2] P. Dutta, R.R. Rhinehart, Application of neural network control to distillation and an experimental comparison with other advanced controllers, ISA Transactions 38 (1999) 251–278.
[3] C.R. Cutler, R.L. Ramaker, Dynamic matrix control—a computer control algorithm. AIChE 86th National Meeting, Houston, TX (1979).
[4] C.R. Cutler, R.L. Ramaker, Dynamic matrix control—a computer control algorithm. Joint Automatic Control Conference, San Francisco, CA (1980).
[5] M. Morari, J.H. Lee, Model predictive control: past, present and future, Computers and Chemical Engineering 23 (1999) 667–682.
[6] C.E. Garcia, D.M. Prett, M. Morari, Model predictive control: theory and practice—a survey, Automatica 25 (1989) 335–348.
[7] K.R. Muske, J.B. Rawlins, Model predictive control with linear models, A.I.Ch.E. Journal 39 (1993) 262–287.
[8] B.R. Maner, F.J. Doyle, B.A. Ogunnaike, R.K. Pearson, Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order volterra model, Automatica 32 (1996) 1285–1301.
[9] A.A. Patwardhan, J.B. Rawlings, T.F. Edgar, Non-linear model predictive control, Chemical Engineering Communications 87 (1990) 123–135.
[10] P.B. Sistu, B.W. Bequette, Non-linear predictive control of uncertain processes: application to a CSTR, A.I.Ch.E. Journal 37 (1991) 1711–1723.
[11] J.C. MacMurray, D.M. Himmelblau, Modeling and control of a packed distillation column using artificial neural networks, Computers and Chemical Engineering 19 (1995) 1077–1088.
[12] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems 2 (1989) 303–314.
[13] F.R. Gao, F.L. Wang, M.Z. Li, Predictive control for processes with input dynamic nonlinearity, Chemical Engineering Science 55 (2000) 4045–4052.
[14] N. Bhat, J.J. McAvoy, Use of neural nets for dynamic modeling and control of chemical process systems, Computers and Chemical Engineering 14 (1990) 573–582.
[15] E.P. Nahas, M.A. Henson, D.E. Seborg, Nonlinear internal model control strategy for neural network models, Computers and Chemical Engineering 16 (1992) 1039–1057.
[16] J.S. Lin, S.S. Jang, Nonlinear dynamic artificial neural network modeling using an information theory based experimental design approach, Ind. Eng. Chem. Res. 37 (1998) 3640–3651.
[17] J.A. Leonard, M.A. Kramer, L.H. Ungar, A neural network architecture that computes its own reliability, Computers and Chemical Engineering 16 (1992) 819–835.
[18] V.G. Krishnapura, A. Jutan, A neural adaptive controller, Chemical Engineering Science 55 (2000) 3803–3812.
[19] D. Psaltis, A. Sideris, A.A. Yamamura, A multilayered neural network controller. IEEE Control Systems Magazine (1988) 17–21.
[20] D.H. Nguyen, B. Widrow, Neural networks for self-learning control systems, International journal of control 54 (6) (1991) 1439–1451.
[21] A.P. Loh, K.O. Looi, K.F. Fong, Neural network modeling and control strategies for a pH process, J, Proc. Control 5 (1995) 355–362.
[22] K.J. Åström, T. Hägglund, Automatic Tuning of PID Controllers, ISA, Research Triangle Park, NC, USA, 1988.
[23] G.L. Williams, R.R. Rhinehart, J.B. Riggs, In-line process-model- based control of waste-water pH using dual base injection, Ind. Eng. Chem. Res. 29 (1990) 1254–1259.
[24] H.C. Chan, C.C. Yu, Autotuning of gain-scheduled pH control: an experimental study, Ind. Eng. Chem. Res. 34 (1995) 1718–1729.
[25] M.C. Palancar, J.M. Aragón, J.A. Miguéns, J.S. Torrecilla, Application of a model reference adaptive control system to the pH-control, Effects of lag and delay time. Ind. Eng. Chem. Res. 35 (1996) 4100–4110.
[26] M.C. Palancar, J.M. Aragón, J.A. Miguéns, J.S. Torrecilla, pH-

control system based on artificial neural networks, Ind. Eng. Chem. Res. 37 (1998) 2729–2740.
[27] M.J. Syu, J.B. Chang, Recurrent backpropagation neural network adaptive control of penicillin acylase fermentation by arthrobacter viscosus, Ind. Eng. Chem. Res. 36 (1997) 3756–3761.
[28] Haykin, S., Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall International, Inc., 1999.
[29] D. Psaltis, A. Sideris, A.A. Yamamura, A multilayered neural network controller, IEEE Control Systems Magazine 8 (1988) 17–21.
[30] Y.Y. Yang, D.A. Linkens, Adaptive neural-network-based approach for the control of continuously stirred tank reactor, IEE Proc.-Control Theory Appl 141 (5) (1994) 341–349.
[31] I.L. Chien, P.S. Fruehauf, Consider IMC tuning to improve controller performance, Chem. Engng Progr 86 (1990) 33–41.
[32] H. Wang, Y. Oh, E.S. Yoon, Strategies for modeling and control of nonlinear chemical processes using neural networks, Computers and Chemical Engineering 22 (1998) S823–S826.
[33] D.F. Specht, A general regression neural network, IEEE Transaction on Neural Networks 2 (1991) 568–576.