Control Engineering Practice, Volume 18, Issue 12, December 2010, Pages 1418-1428

Identification and Predictive Control of a Multistage Evaporator[○]

J.C. Atuonwu [a#], Y. Cao [a*], G.P. Rangaiah [b], M.O. Tadé [c]

[○] This paper was not presented at any IFAC meeting
[a] School of Engineering, Cranfield University, Bedford, UK
[b] Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore, rangaiahGP@nus.edu.sg
[c] Department of Chemical Engineering, Curtin University of Technology, Perth, Western Australia, m.o.tade@curtin.edu
[*] Corresponding author: y.cao@cranfield.ac.uk Phone +44 (0) 1234750111 ext 5650 Fax +44 (0) 1234 754685
[#] Current Address: Systems and Control Group, Wageningen University, Wageningen, Netherlands, james.atuonwu@wur.nl

Abstract

A recurrent neural network-based nonlinear model predictive control (NMPC) scheme in parallel with PI control loops is developed for a simulation model of an industrial-scale five-stage evaporator. Input-output data from system identification experiments are used in training the network using the Levenberg-Marquardt algorithm with automatic differentiation. The same optimization algorithm is used in predictive control of the plant. The scheme is tested with set-point tracking and disturbance rejection problems on the plant while control performance is compared with that of PI controllers, a simplified mechanistic model-based NMPC developed in previous work and a linear model predictive controller (LMPC). Results show significant improvements in control performance by the new parallel NMPC-PI control scheme.

Key words: Multiple-effect evaporators, nonlinear model predictive control, nonlinear system identification, recurrent neural networks, automatic differentiation.

1. Introduction

The class of control algorithms that utilize dynamic process models as a means of incorporating process knowledge into the controller structure for solving open-loop optimal control problems at defined time steps over a receding time horizon has received significant attention in recent times. Generically referred to as model predictive control (MPC), these algorithms have been well received in industry because of their effectiveness in the high performance control of multivariable systems with strong process interactions and operating constraints, without the need for explicit input-output pairing. MPC techniques have been implemented in various forms, some of which have found their way to the market place. Examples of such implementation algorithms are available in the review by Qin and Badgwell (2003).

In industrial processes, nonlinearity is the rule. Even for processes that are linear over a wide operating range, valve saturation places a limit on the level of input for which the system remains linear (Atuonwu et al 2009). In spite of this, industrial applications of MPC have relied mainly on linear dynamic models until recently (Al-Seyab and Cao, 2008a & b). This is because the assumption of process linearity greatly simplifies model development and control design (Henson, 1998). The performance of an MPC system is inherently tied to the accuracy of prediction which in turn depends on the degree of matching between the process and its prediction model (Atuonwu et al 2009). A linear model is often sufficient in cases where the process operates at a single set-point and the control problem amounts to the rejection of small disturbances (Piché et al, 2000). However, operating points of many chemical processes vary over large regions such that a linear model is unable to adequately capture the process dynamics (Al-Seyab and Cao, 2008a & b). Moreover, some processes are severely nonlinear even in the vicinity of steady state. In view of these, the last two decades or so have witnessed a tremendous increase in research effort towards the use of nonlinear models. This has culminated in the development of various nonlinear model-based commercial products which include (Qin and Badgwell, 2003): Predictive Functional Control (PFC), courtesy of *Adersa*, Multivariable Control (MVC), courtesy of *Continental Controls*, Aspen Target, courtesy of *Aspen Technologies*, NOVA-NLC, courtesy of *DOT Products* and Process Perfecter, courtesy of *Pavilion Technologies*. Until recently, reported nonlinear model predictive control (NMPC) applications in the literature have focused mainly on small-size problems especially single-input, single-output problems (Henson, 1998; Rangaiah et al, 2002; Yu and Gomm, 2003). The main objective of this work is to develop and apply a differential recurrent neural network (DRNN) model based NMPC scheme to a moderate-size problem, a 5-input, 5-output, 12-state industrial multistage evaporator system.

Although the use of nonlinear models in the predictive control of nonlinear processes will in principle give better performance than the use of linear models, some other problems arise. First, the use of nonlinear models leads to a non-convex optimization problem for which a globally optimal solution is not guaranteed. Secondly, model development becomes more difficult. The development of first principle models is usually time-consuming and effort-demanding especially for complex processes (Henson, 1998), and involves making assumptions which most times, negatively affect accuracy. In many cases therefore, the use of empirical or black box models becomes an inevitable step in an NMPC project (Al-Seyab and Cao, 2008a & b). One of the advantages of black box modeling is that there is no need for a priori knowledge of what is going on in the plant. All that is needed is a set of input-output data obtained from plant measurements from which a nonlinear mathematical description of the plant can be inferred. This is nonlinear system identification. However, due to the complexity of nonlinear systems, it is not possible to develop nonlinear system identification techniques by a straightforward extension of linear theory.

A fundamental difficulty associated with empirical modeling approaches is the selection of a suitable modeling form (Henson, 1998). Pearson and Ogunnaike (1997) categorized the models utilized for NMPC including: Volterra, Hammerstein, Wiener, polynomial autoregressive moving average model with exogenous inputs (polynomial ARMAX), its nonlinear form, (NARMAX), and artificial neural network (ANN) models. The use of block-structured nonlinear time series models, containing linear dynamic and nonlinear static elements (e.g. Wiener and Hammerstein models) for NMPC is well-reported in the literature. To reduce computational burden, the solution of such systems is usually based on the inversion of the nonlinear element such that the nonlinear dynamic optimization is avoided (Zhu and Seborg, 1994; Norquay et al, 1998; Piché et al, 2000). However, the fact that the nonlinear element must be invertible limits the applicability of this solution. Also, to obtain unique solutions with this method, the nonlinearity of the model must be bijective, which is not generally applicable. For MIMO systems, more restrictions on the model structures are imposed. In response to this, Harnischmacher and Marquardt (2007) posed the NMPC problem as a nonlinear dynamic optimization problem constrained by the block-structured model. The result was an approximate model, which provides a viable compromise between the limited predictive capabilities of a linear model and the costly development of a rigorous nonlinear dynamic model. The model is however inferior in prediction quality to a rigorous nonlinear model such as the one provided by recurrent neural networks. Moreover, using the so-called two-step model identification approach (Nagy, et al., 2007) for system identification (to be explained later) in recurrent neural network modeling, the high costs of physical system identification can be eliminated.

The nonlinear model types mentioned above are not suitable for the multi-effect evaporator process. Wiener/ Hammerstein type nonlinear models can only capture static nonlinearities, i.e. either static input nonlinearities e.g. $\dot{x} = Ax + f(u)$, or static output nonlinearities, e.g. $y = g(x)$. However, both models are unable to capture dynamic nonlinearities, e.g. $\dot{x} = f(x,u)$. The mechanistic model equations of the evaporator (as presented in the Appendix) clearly indicate that the multiple-effect evaporator is neither a Wiener nor a Hammerstein type system. For example, dT/dt is nonlinearly dependent on other states, h_1 and ρ_1. Similarly, both dh_1/dt and $d\rho_1/dt$ are functions of E_1, which is nonlinearly dependent on ρ_1 and T_1. Furthermore, due to the recycling of the multiple-effect evaporator, such dynamic nonlinearity strengthens through internal state coupling. Therefore, Weiner / Hammerstein type nonlinear models are not sufficient to represent such a system. Some nonlinear models, such as NARMA and OBF-NARX, are one-step prediction models. The deficiency of such kind of models, e.g. FFNN, for NMPC is to be discussed below. On the other hand, Volterra series models require a large number of parameters to be optimized so that it is very difficult to apply such models to high order systems, such as the multi-stage evaporator, which has 17 states. Finally, almost all of these methods are in discrete-time, which are only valid for a predetermined sampling rate. A continuous-time DRNN model is developed successfully in this work to capture the dynamic nonlinearity of the plant in satisfactory with a relatively small parameter space. This success together with others reported in the literature demonstrates the advantage of the continuous-time DRNN type model.

Of all the nonlinear black box modeling techniques available, artificial neural networks are the most popular (Su and McAvoy, 1997). One reason for the widespread application of artificial neural networks is that as parallel signal processing devices they are inherently fast and so, hold the promise of solving problems that have proven difficult to solve by conventional computers. They also have the ability to learn from examples. The use of neural networks for the predictive control of nonlinear processes has been widely studied. Galvan and Zaldivar (1998) applied recurrent neural networks for nonlinear predictive control of the heat transfer fluid temperature in a batch reactor. Schenker and Agarwal (2000) employed an estimation-prediction approach involving two neural networks for the predictive control of a semi-batch reactor. More recently, Mohanty (2009) applied neural networks to the predictive control of a flotation column, while Mjalli and Hussain (2009) used a feed-forward neural network with delayed inputs and outputs to capture the dynamics of a bio-diesel process. In these and all other cases, the most commonly used neural network architecture is the multilayer perceptron, MLP (Nørgaard et al, 2000) which is categorized into two: the feed-forward (FFNN) and the recurrent neural networks (RNN). Majority of publications on neural network-based system identification and predictive control report the use of FFNNs , e.g., Temeng et al (1995). The main problem with FFNNs is that they can only provide predictions for a predetermined number of steps, usually only one step (Al-Seyab and Cao, 2008a & b). One way of providing long range predictions using FFNNs is to cascade them by using the one-step-ahead output of an FFNN as an input to another so that for n FFNNs, an n-step-ahead prediction is realized (Werbos, 1990). This approach, however, faces the problem of error propagation which degrades long-range prediction accuracy (Ou and Rhinehart, 2002). Jazayeri-Rad (2004), proposed the use of multiple FFNNs with one hidden layer to model an m-input, n-output nonlinear dynamic system. This system was made up of a two-dimensional array of FFNN blocks with each block consisting of a one-step-ahead predictive neural model, identified to represent each output of the MIMO system. Though this approach has been proven to solve the multi-step-ahead prediction problem of FFNNs, it demands the training of a new FFNN for every extension to the prediction horizon.

Recurrent neural networks (RNNs), on the other hand, are able to provide long-range predictions, even in the presence of measurement noise (Su and McAvoy, 1997). Models based on RNNs have proven to have considerably greater representational capabilities than FFNN-based models (Puskorius and Feldkamp, 1994). They have the capability of capturing various plant nonlinearities (Funahashi and Nakamura, 1993), and require less number of neurons to model dynamic systems (Hush and Horne, 1993). RNNs are also more suitable for representation in state space form which is commonly used in most control algorithms (Zamarreno and Vega, 1998). The main problem with RNNs is in their training due to the large number of sensitivity equations to be solved in the associated nonlinear optimization problem. Ou and Rhinehart (2002) considered the computational burden in RNN training so prohibitive that they preferred implementing an FFNN to approximate the dynamics of a case-study mixing flow system. Sentoni et al (1998) also identified this problem and proposed the development of "efficient and feasible training methods". They then, developed another model scheme consisting of a decoupled linear dynamic system cascaded with a single hidden layer perceptron (nonlinear static map). The efficient training problem however remained unresolved. Various training strategies for neural networks have been suggested in the literature, such as the back-propagation method (Rumelhart et al, 1986), the conjugate gradient method (Leonard and Kramer, 1990), Levenberg–Marquardt optimization (Marquardt, 1963), or methods based on genetic algorithms (Goldberge, 1989). Among these, the Levenberg–Marquardt algorithm is known to be a robust and fast gradient-based method because of its second-order converging speed without having to compute the Hessian matrix (Al Seyab and Cao, 2008a & b). Recently, automatic differentiation and Taylor series techniques have been used to simplify the dynamic sensitivity equations associated with the network training and NMPC algorithms (Cao and Al-Seyab, 2003; Cao, 2005), thus increasing the efficiency of, and reducing the computational burden associated with solving online, the associated nonlinear differential equations and nonlinear dynamic optimization problems in real time.

RNNs can be formulated in discrete-time (Zamarreno and Vega, 1998), or continuous-time (Funahashi and Nakamura, 1993). Discrete-time RNNs can only work for a particular sampling frequency and no information is given about the model trajectories between the sampling instants. If the sampling frequency is to change, the model has to be re-built. On the other hand, a continuous-time RNN can be used for any sampling frequency (Kambhampati et al, 2000) even for continuous-time NMPC. Although a continuous-time RNN has clear advantages, it has rarely been used in NMPC. The main reason probably is due to the difficulty in solving the differential parameter optimization problem associated with the continuous-time nonlinear model identification problems.

In this work, a continuous-time DRNN model of a five-stage evaporator is developed, trained using automatic differentiation techniques and implemented as the internal model of the NMPC scheme for the prediction of the dynamic behavior of the plant. The example evaporator is the first step in the liquor burning process at an alumina refinery. The plant is open-loop unstable due to the integrating properties of the flash tank levels. An open-loop identification experiment cannot be carried out without a stabilizing controller in place because of the open-loop unstable nature. Therefore, closed-loop system identification was adopted in this work. The PI controller plays the role of stabilizing the plant so that a wide range of input-output data can be collected. In addition, as pointed out by Draeger , et al (1995), closed loop system identification with PI controllers is a more practical and effective approach than open-loop system identification since the plant is always under control, and the excitation becomes much more similar to the one that will be encountered under the final nonlinear control scheme. A previously developed feedback stabilizing PI controller (Rangaiah et al., 2002) is used in stabilizing the plant during closed-loop identification where the perturbations are added directly to the manipulated inputs. The new control algorithm is then implemented with the PI controller in place.

The remainder of the paper is organized as follows. Section 2 gives details of the evaporator system and the identification results. The developed model is used for NMPC as described in Section 3. Finally, the work is concluded in Section 4.

2. Evaporator System Identification

2.1. The Multistage Evaporator: System Description

Evaporation is a key unit operation applied in a wide variety of process industries such as the food and beverage, pharmaceutical, pulp and paper, mineral processing industries, amongst others. The evaporator system chosen for predictive control is the first step in the liquor burning process associated with the Bayer process for alumina production at the Wagerup alumina factory in Western Australia. This evaporator system has being identified in previous studies (Ekawati et al, 2003) as being “notoriously” nonlinear and interacting so that linear control schemes are inadequate. The system consists of one falling-film, three forced circulation and one super-concentration evaporator connected in series. Each of these five stages basically consists of a flash tank (FT), a heater (HT) and a flash pot. Fig. 1 shows a schematic of the system with the flash pot omitted for simplicity. Fresh liquor, recovered after precipitation of the alumina from its solution is fed to the falling film stage (FT#1). The volatile component, water is removed under high recycle rate and further concentrated through the forced-circulation stages, FT#2 – FT#4 (Fig. 1). FT#5 is a super-concentration stage that serves the purpose of further concentrating the product without recycling. In each of the forced-circulation and super-concentration stages, the spent liquor is heated through a shell and tube heat exchanger (heater) and water is removed as vapour at a lower pressure in the flash tank. The flashed vapour is then used as the pre-heating medium in upstream of the heaters. The flashed vapour from FT#3 and FT#4 are combined and used in HT#2 while the vapour from FT#2 is used in HT#1. The flashed vapour from FT#5 is sent directly to the condenser, C. The steam condensates from the heaters are collected in the flash pots. Live steam is used as the heating medium for HT#3, 4 and 5. Live steam to HT#3 is set in ratio to the amount of live steam entering HT#4, while the amount of live steam to HT#5 is set depending on the amount of residual ‘flashing’ to be removed. The cooling water flow to the contact condenser is set such that all remaining flashed vapour is condensed. The evaporator system is crucial in the aluminium refinery operation and is difficult to control due to recycle streams, strong process interaction, nonlinearities and the integrating properties of the flash tank level which makes the system open loop unstable. These flash tank levels therefore have to be controlled in addition to product density which gives an indication of quality. Two mechanistic models M1 and M2 of the first four stages of the evaporator have been developed (Kam and Tadé, 1997) and implemented in various platforms (Kam et al, 2001). Complete details of the models are available in Kam and Tadé (1997). To appreciate the difference between the models, an overview of the equations of the first stage is presented in Appendix 1 (for both models). Also, the assumptions made in their development and more importantly, the differences between the models are presented. Because of the fewer assumptions made in its development, model M1 gives a much better approximation to the true process than model M2. Hence, in this work, M1 is adopted to represent the process.

For this process, there are 12 state variables whose steady-state values are shown in Table 1. The first five states (flash tank levels $h_i \; i = 1,2,3,4$ and the product liquor density of the 4th stage ρ_4) are output variables. Other states are, the product liquor temperatures T_1, T_2, T_3 and T_4 for which online measurements are available in the actual plant situation together with the unmeasured states which consist of the densities ρ_1, ρ_2 and ρ_3 of the product liquors from the first three stages. Control is achieved by manipulating the product liquor flows, $(Q_{Pi}, i = 1,2,3,4)$ from each of the first four stages and the steam flowrate to the 4th heater (m_{S4}).

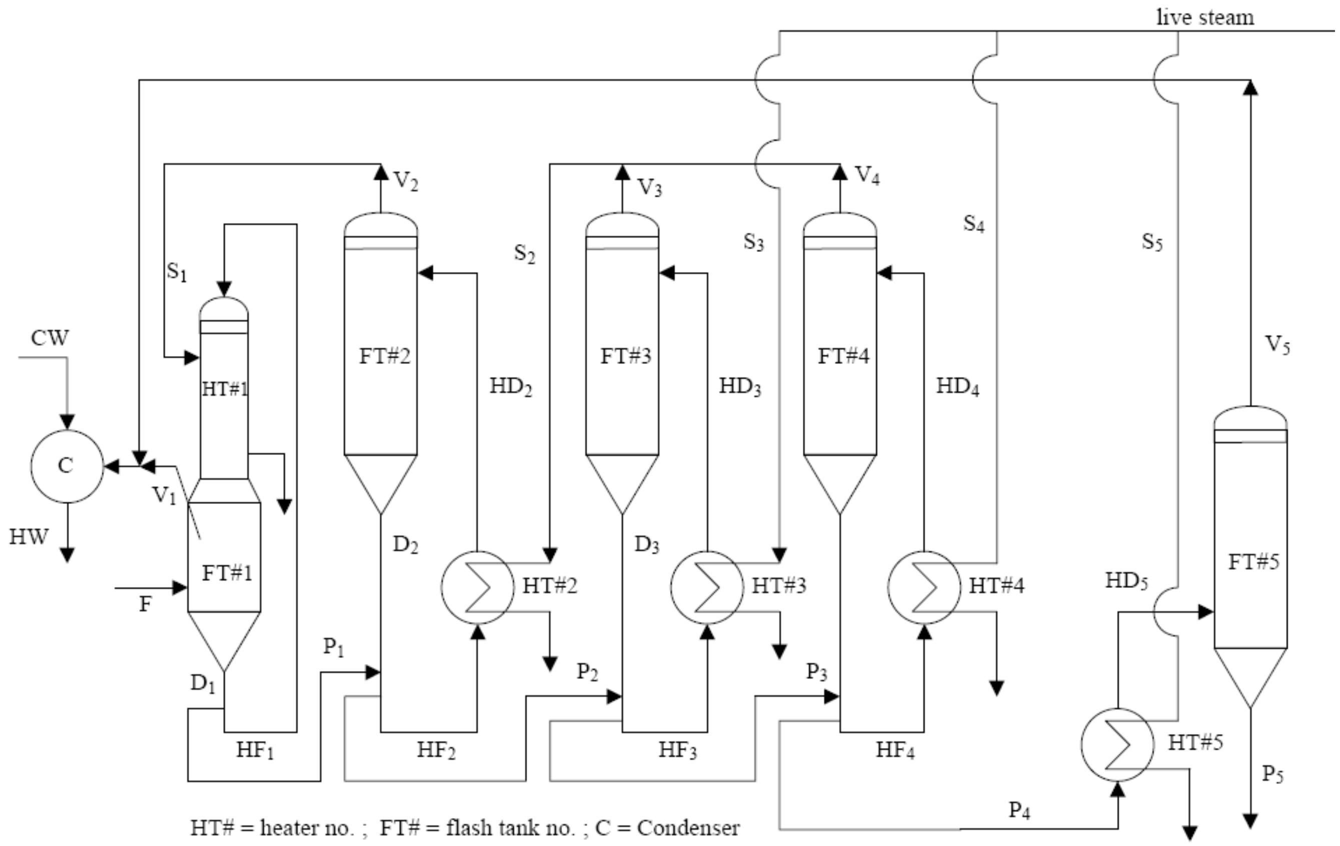

Fig. 1. The Multistage Evaporator System

Applications of neural networks to the predictive control of multi-input, multi-output (MIMO) processes are rare in literature. A few reported cases (e.g. Chen and Yea, 2002; Yu and Gomm, 2003) usually decompose the system into a number of multi-input, single-output (MISO) subsystems which are later combined to form a parallel process model. This leads to sub-optimal results. In this work, the MIMO system is represented in a single recurrent neural network model.

2.2. *Model Development*

The evaporator system can be modeled in state space as

$$\dot{\mathbf{x}}_{\mathbf{p}} = f_p(\mathbf{x}_{\mathbf{p}}, \mathbf{u}) \tag{1}$$

$$\mathbf{y}_{\mathbf{p}} = g_p(\mathbf{x}_{\mathbf{p}}, \mathbf{u}) \tag{2}$$

where f_p and hence g_p are unknown nonlinear functions of state $\mathbf{x}_{\mathbf{p}}$ and input $\mathbf{u}$. The input, state and output vectors are defined respectively as:

$$\mathbf{u} = \begin{bmatrix} Q_{P1} & Q_{P2} & Q_{P3} & Q_{P4} & m_{S4} \end{bmatrix}^T \qquad (3)$$

$$\mathbf{x_p} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \rho_4 & T_1 & T_2 & T_3 & T_4 & \rho_1 & \rho_2 & \rho_3 \end{bmatrix}^T \qquad (4)$$

$$\mathbf{y_p} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & \rho_4 \end{bmatrix}^T \qquad (5)$$

Table 1. Steady-state Values of Input & State Variables

Input	*Description*	*Value*	*State*	*Description*	*Value*
Q_{P1}	1st stage product flowrate	$32.736\, m^3/h$	h_1	1st stage flash tank level	$1.5\, m$
Q_{P2}	2nd stage product flowrate	$27.713\, m^3/h$	h_2	2nd stage flash tank level	$2.25\, m$
Q_{P3}	3rd stage product flowrate	$28.350\, m^3/h$	h_3	3rd stage flash tank level	$2.25\, m$
Q_{P4}	4th stage product flowrate	$21.642\, m^3/h$	h_4	4th stage flash tank level	$2.25\, m$
m_{S4}	4th stage steam flowrate	$24\, kg/h$	ρ_4	4th stage product density	$1.54\, g/cm^3$
			T_1	1st stage product temperature	$66\,^\circ C$
			T_2	2nd stage product temperature	$90.6\,^\circ C$
			T_3	3rd stage product temperature	$129\,^\circ C$
			T_4	4th stage product temperature	$135\,^\circ C$
			ρ_1	1st stage product density	$1.357\, g/cm^3$
			ρ_2	2nd stage product density	$1.422\, g/cm^3$
			ρ_3	3rd stage product density	$1.49\, g/cm^3$

To approximate the unknown functions, an RNN model is developed with the general form:

$$\dot{\mathbf{x}} = f\left(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}\right) \qquad (6)$$

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) \tag{7}$$

where $\boldsymbol{\theta}$ is the network parameter vector comprising of the weights and biases, f and g are nonlinear functions, which are determined by iteratively adjusting model parameters, $\boldsymbol{\theta}$ until the sum of squared errors (SSE) represented in (8) is minimized.

$$\varphi = \frac{1}{2}\sum_{k=1}^{N} \mathbf{e}_k{}^T \mathbf{e}_k = \frac{1}{2}\mathbf{E}^T\mathbf{E} \tag{8}$$

Here, the error vector at each sampling point k is defined as:

$$\mathbf{e}_k = \mathbf{y}_\mathbf{p}(k) - \mathbf{y}(k, \boldsymbol{\theta}) \tag{9}$$

Hence,

$$\mathbf{E} = \mathbf{E}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{e}_1^T & \cdots & \mathbf{e}_n^T \end{bmatrix}^T \tag{10}$$

The Jacobian $\mathbf{J}$ of the error and the sensitivity $\varphi_{\boldsymbol{\theta}}$ are calculated iteratively using automatic differentiation (Cao, 2005) as follows:

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial \mathbf{E}}{\partial \boldsymbol{\theta}} \tag{11}$$

$$\varphi_{\boldsymbol{\theta}} = \frac{d\varphi}{d\boldsymbol{\theta}} = \mathbf{J}(\boldsymbol{\theta})\mathbf{E}(\boldsymbol{\theta}) \tag{12}$$

The variable $\boldsymbol{\theta}$ is iteratively adjusted using the Levenberg- Marquardt nonlinear least squares algorithm as follows

$$\boldsymbol{\theta}_{\mathbf{j+1}} = \boldsymbol{\theta}_\mathbf{j} - \left[\mathbf{J}^T(\boldsymbol{\theta}_\mathbf{j})\mathbf{J}(\boldsymbol{\theta}_\mathbf{j}) + \mu\mathbf{I}\right]^{-1}\mathbf{J}^T(\boldsymbol{\theta}_\mathbf{j})\mathbf{E}(\boldsymbol{\theta}_\mathbf{j}) \tag{13}$$

where $\boldsymbol{\theta}_{\mathbf{j+1}}$ is the vector of updated model parameters, $\boldsymbol{\theta}_\mathbf{j}$ the current parameter vector, μ a scalar which is to be adjusted by the algorithm during the iteration and $\mathbf{I}$ the identity matrix.

The governing equations of the model (as shown in Fig. 2) are:

$$\dot{\mathbf{x}} = \mathbf{W}_2\boldsymbol{\sigma}_\mathbf{s}(\mathbf{W}_\mathbf{x}\mathbf{x} + \mathbf{W}_\mathbf{u}\mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2 \tag{14}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{15}$$

where $\mathbf{W_x} \in \Re^{n_h x n_x}$, $\mathbf{W_u} \in \Re^{n_h x n_u}$ and $\mathbf{W_2} \in \Re^{n_x x n_h}$ are connection weights, and $\mathbf{b_1} \in \Re^{n_h}$ and $\mathbf{b_2} \in \Re^{n_x}$ are bias vectors. n_h represents the number of hidden nodes, n_x represents the number of network states while n_u and n_y represent the numbers of input and output nodes, respectively. Each element of the vector $\boldsymbol{\sigma}_\mathbf{s}(\bullet) \in \Re^{n_h}$ represents the sigmoid-tanh activation function defined as follows:

$$\boldsymbol{\sigma}_\mathbf{s}(\mathbf{z}) = \frac{1-e^{-2\mathbf{z}}}{1+e^{-2\mathbf{z}}} \tag{16}$$

where from equation 14,

$$\mathbf{z} = \mathbf{W_u}\mathbf{u} + \mathbf{W_x}\mathbf{x} + \mathbf{b_1} \tag{17}$$

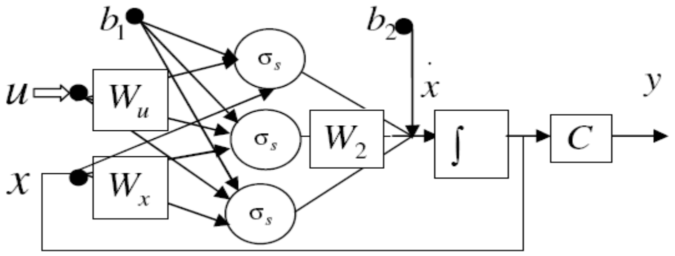

Fig. 2 Structure of the differential recurrent neural network

The parameter vector $\boldsymbol{\theta}$ is thus defined as

$$\boldsymbol{\theta} = \left\lfloor vec(\mathbf{W_x})^T \; vec(\mathbf{W_u})^T \; \mathbf{b_1}^T \; vec(\mathbf{W_2})^T \; \mathbf{b_2}^T \right\rfloor \in \Re^{n_\theta} \tag{18}$$

where the number of elements of $\boldsymbol{\theta}$ is

$$n_\theta = n_x(n_h + 1) + n_h(n_x + n_u + 1) \tag{19}$$

The output linear activation function $\mathbf{C}$ is defined as

$$\mathbf{C} = \left\lfloor \mathbf{I}_{ny \times ny,} \, \mathbf{0}_{ny \times (nx - ny)} \right\rfloor \tag{20}$$

2.3. Model Topology

The predictive power of an artificial neural network (ANN) depends largely on its topology. The topology consists of the number of model parameters as well as their distribution in the input, hidden and output layers. The errors in the parameter estimates are directly proportional to the number of model parameters (n_θ) and inversely proportional to the number of data points (N_s). For a given number of data points, the number of model parameters should be kept as small as possible, but large enough to learn the system dynamics, while carrying out model parameter estimation (Sjoberg, et al., 1995). A number of studies have been done in this respect. For instance, Bhat and McAvoy (1992) formulated an optimization problem with a two-term objective function. The first term, a complexity term attempts to force all weights towards zero while the error term selectively favours the positive increment of weights that contribute largely to error minimization. The result is that only the least contributing weights are driven to the zero region where they are deleted or stripped. This method is thus called "network stripping". More recently, Nagy (2007) proposed a pruning method based on the so-called optimal brain surgeon algorithm. In this approach, weights that do not contribute significantly to performance are eliminated thus, preventing over fitting and reducing training time. In this work, the size of the network is defined by equation (18) corresponding to the structure shown in Fig. 2. The plant to be identified in this study consists of 5 inputs, 5 outputs and 17 states comprising the original 12 plant states and 5 additional states from PI controllers. Each input is represented by an input neuron and each output is represented by an output neuron, while the number of network state parameters is 17. The system is thus constrained to $n_u = n_y = 5$ and n_x =17. The only decision variable therefore is the number of hidden nodes, n_h.This was determined by varying n_h during training and checking training errors (for learning performance) and validation errors (for generalization). Table 2 shows the sum of squared errors at the end of training and at the end of validation with a fresh dataset. Using 1251 data points for both training and validation (to be discussed in the next section), it is seen that for $n_h < 3$, under fitting occurs as both training and validation results are poor. For $n_h > 3$, over fitting occurs as validation becomes poorer even though training results are still good. Moreover, training time is severely increased. Thus, an optimal number $n_h = 3$ is chosen and correspondingly the total number of model parameters given in (19) is 137. This corroborates the results of Wanas, et al. (1998), who showed empirically that neural networks perform best when the number of hidden nodes equals to the logarithm of the number of training samples.

Table 2. Error dependence on number of hidden nodes

Number of hidden nodes	*SSE for training data*	*SSE for validation data*
1	27.4191	29.7812
2	16.1644	21.6379
3	14.2811	21.4595
4	14.1660	404.3486

2.4. Model Training and Validation

The identification of industrial plants is a costly and time-consuming exercise involving many experiments. The detailed mechanistic model M1 (Kam and Tadé, 1997) is a good approximation to the plant due to fewer assumptions and the incorporation of some plant measurements in addition to the heat and mass balances. However, it is too complex to be used in controller design (Rangaiah, et al., 2002). Thus in this study, a two-step model identification approach (Nagy, et al., 2007) is exploited as follows:

- The development of M1 from the actual plant done in previous work was taken as the first step.
- The RNN model is then subsequently identified from the mechanistic model M1. The identification scheme assumes that the plant model equations are unknown and the only available information is the input–output data which are generated through various runs of model M1.

The above approach has the advantage of eliminating extensive real plant experiments. In addition, it is valid as M1 closely approximates the actual plant. For training, $N = 1251$ samples of input-output data were collected by applying normally distributed random signals u_{rand} to the process which consists of M1 with the PI controllers connected in closed loops as shown in Fig. 3. u_{rand} , which represents the perturbations of the input variables from their steady-state values, is the input while y is the output. In this identification, the sampling rate was 10 samples per hour (6 min per sample) over a period of 125 hours. At the same time, another dataset consisting of 1251 points but at a higher sampling rate of 20 samples per hour (3 min per sample) was collected and used for validation over a period of 62.5 hours. The excitation input was chosen to be random such that it contains a wide range of frequencies capable of persistent excitation. The sampling time was chosen to be 0.1 hours, about a quarter of the smallest time constant in the system and hence, it is low enough to capture the system dynamics. At the same time, the sampling time together with the amplitude of the input perturbations were chosen to be high enough to ensure output amplitudes are driven to the expected range of operation. For instance, the first flash tank level, h_1 is made to cover the range from below 1.5 to above 1.8 m so that the neural network is trained adequately in the region relevant to one of the previously studied set-point tracking cases (Rangaiah, et al., 2002) to be examined in this work. Fig. 4 shows the training and validation input excitations. The training results (Fig. 5) show a good approximation with the coefficient of determination, γ^2 of 95%. The statistical tool “coefficient of determination” (Colannino, 2006) in this case, is the proportion of the variations in the plant output data explainable by the neural network model. Numerically, it satisfies the inequality $0 \le \gamma^2 \le 1$. Mathematically, it is defined as:

$$\gamma^2 = 1 - \frac{\sum_{i=1}^{i=n_y N} \left(y_p(i) - y(i)\right)^T \left(y_p(i) - y(i)\right)}{\sum_{i=1}^{i=n_y N} \left(y_p(i) - \bar{y}_p(i)\right)^T \left(y_p(i) - \bar{y}_p(i)\right)} \tag{21}$$

where $\bar{y}_p$ is the mean value of y_p over N plant output measurements, and y is the neural network output. The validation results (Fig. 6) show reasonably good generalization properties with the γ^2 of 78%.

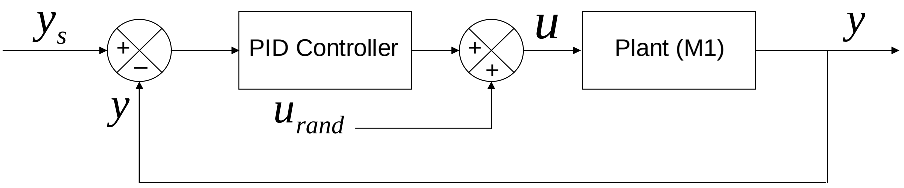

Fig. 3. Schematic signal connections in system identification experiments

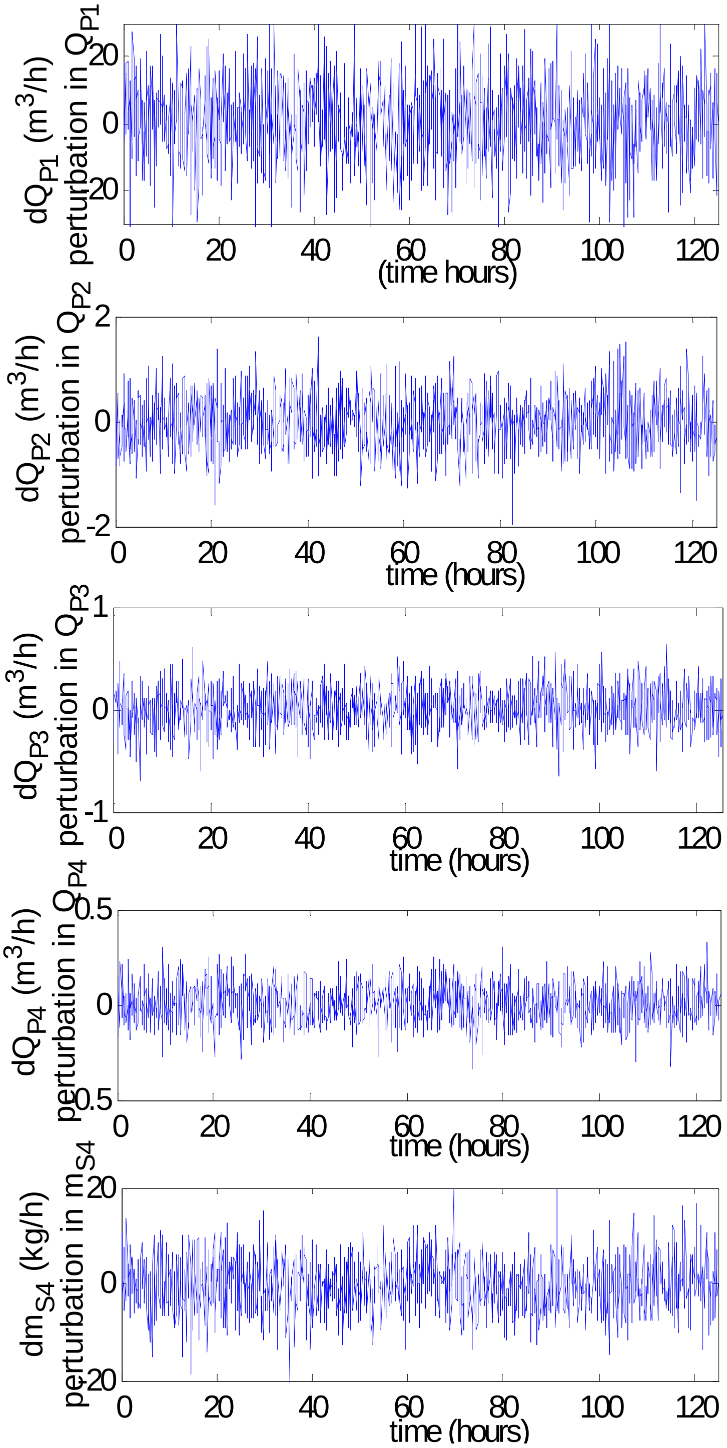

Fig. 4.(a) Training input excitation

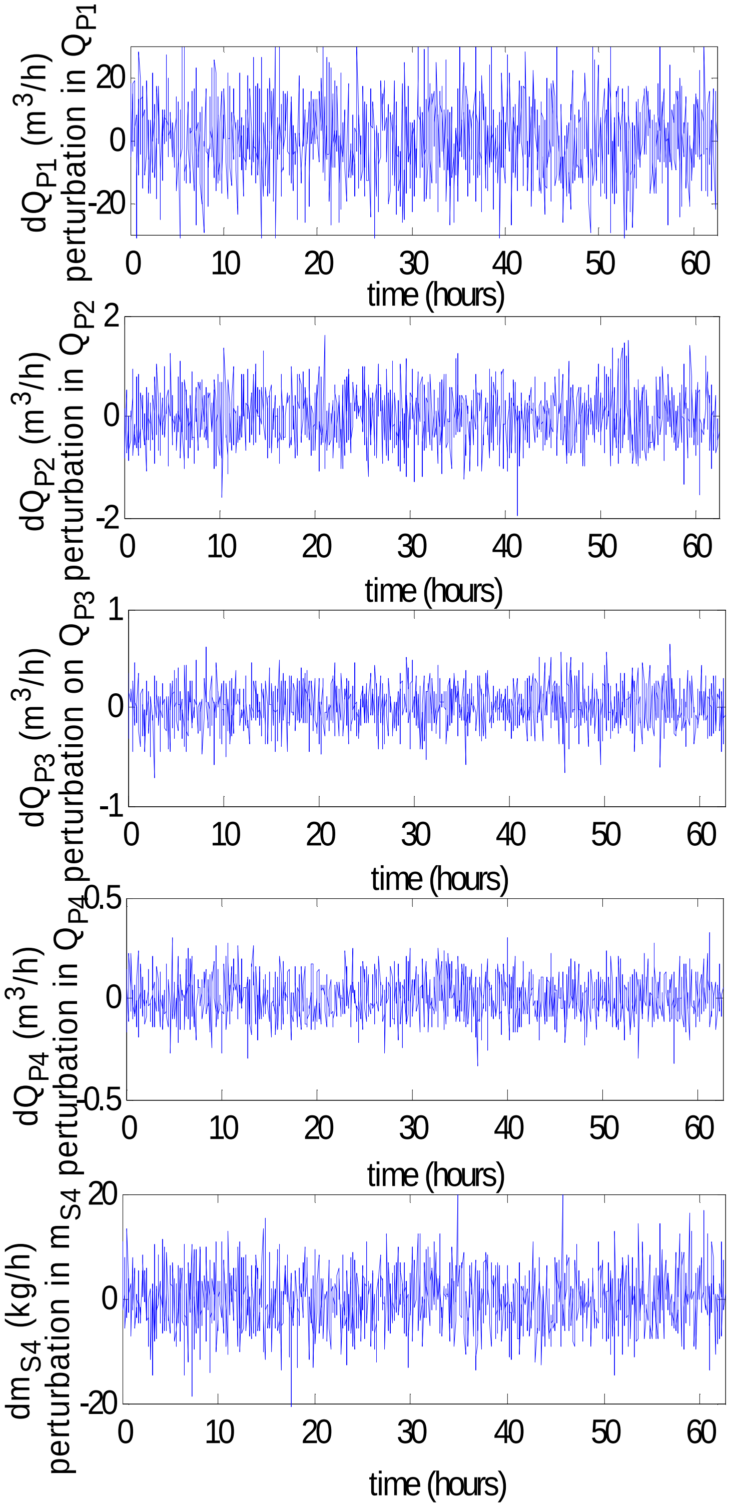

Fig. 4(b).Validation input excitation

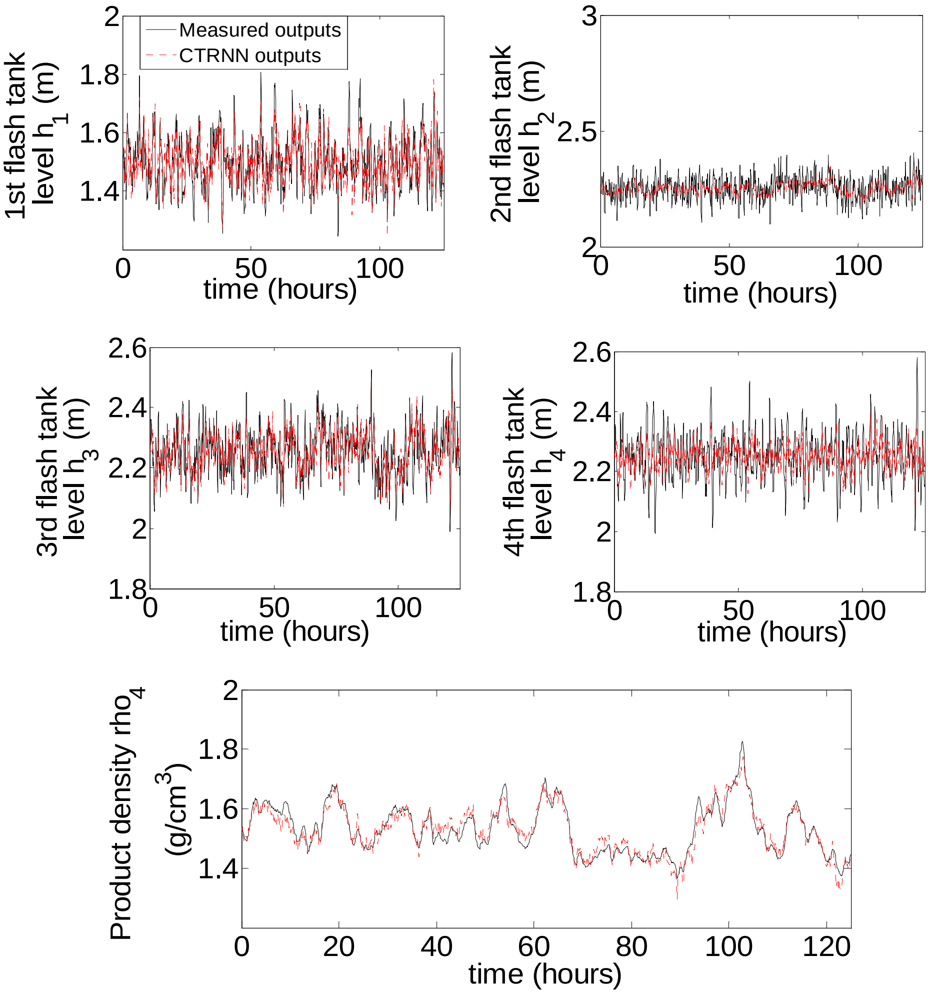

Fig. 5. Model training results

3. Implementation of NMPC on the Evaporator System

Due to its open-loop unstable nature, the plant was identified with the PI controller in place (as shown in Fig. 3). Hence, the control configuration adopted in this work is an NMPC-PID parallel supervisory control scheme which has been adopted in previous works (e.g. Bulut et al., 2000; Balbis, et al., 2006) to improve closed-loop performance. This set-up permits the NMPC to directly manipulate the input in parallel with the PID controllers, which stabilize the plant. An alternative arrangement (not adopted in this study) is the cascade control configuration where the NMPC dictates, based primarily on economics through a higher level economic optimizer, the set-point of the lower level regulatory PID, while the PID does all the regulation. The PI controller parameters determined in previous work (Rangaiah et al., 2002) are $\mathbf{K_c} = [-53.28 \quad -106.45 \; -93.62 \quad -93.15 \quad 0.05]$ and $\boldsymbol{\tau}_\mathbf{i} = [2.528 \quad 19.53 \;\; 10.13 \quad 0.2112 \quad 0.0501]$ for proportional gain and integral time, respectively. Fig. 7 shows the final control scheme incorporating the PID and NMPC controllers, with the NMPC represented by a dashed box containing the components, namely the RNN and optimizer. The

RNN performs model predictions which are then used by the optimizer to generate an input sequence to track the set-point of the plant. The manipulated input to the plant is the sum of the outputs of both controllers configured in a parallel supervisory control structure (Balbis et al., 2006).

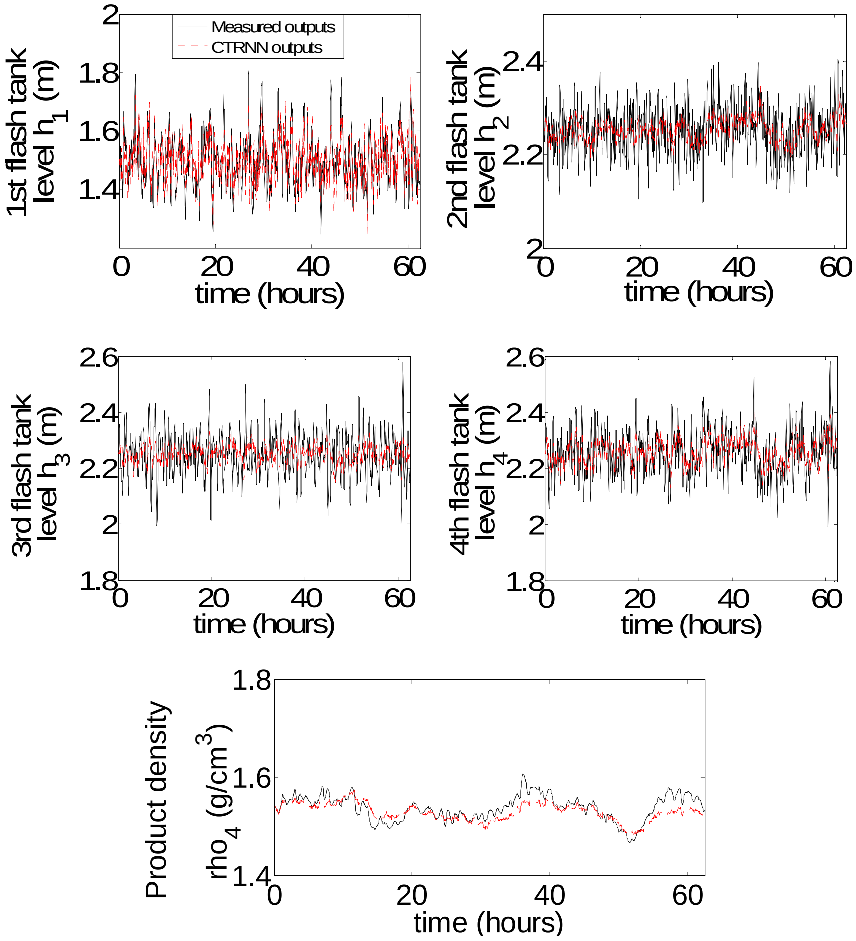

Fig. 6. Model validation results

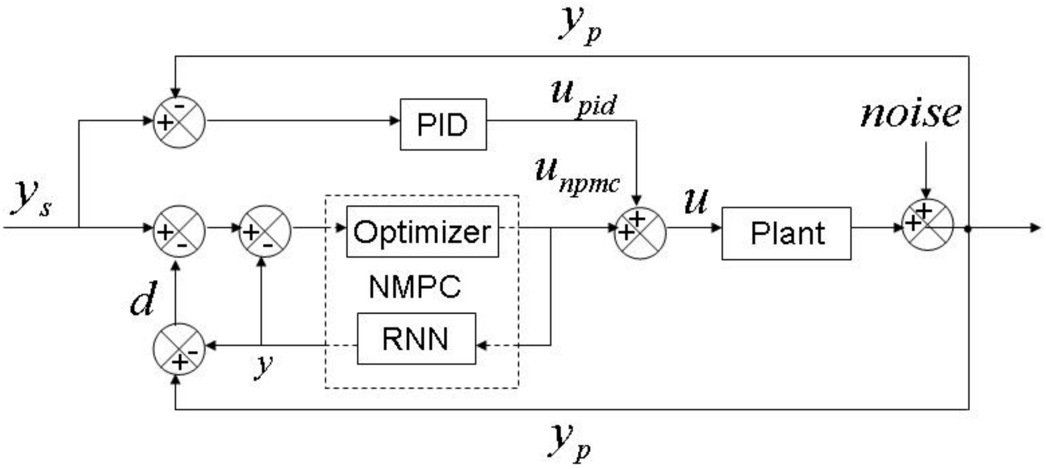

Fig. 7. Block diagram of control scheme

The NMPC optimization problem is formulated as:

$$\min_{u_0,u_1,\ldots,u_{M-1}} \Phi = \frac{1}{2}\sum_{k=1}^{P}\mathbf{e}_{\mathbf{y}k}{}^{T}\mathbf{Q}\mathbf{e}_{\mathbf{y}k} + \sum_{k=1}^{M}\Delta\mathbf{u}_k{}^{T}\mathbf{R}\Delta\mathbf{u}_k \qquad (22)$$

Subject to:

$$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t),\mathbf{u}(t)\right) \qquad t \in [t_{0,} \quad t_P]$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{d}(t) \qquad (23)$$

$$\mathbf{x}_k = \mathbf{x}(t_0 + kT_s)$$

$$\mathbf{u}_k = \mathbf{u}(t_k) = \mathbf{u}(t) \qquad t \in [t_k, \quad t_{k+1}]$$

$$\mathbf{e}_{\mathbf{y}k} = \mathbf{y}_k - \mathbf{y}_{\mathbf{s}k} \qquad k \in [1, \quad P]$$

$$\Delta\mathbf{u}_k = \mathbf{u}_{k+1} - \mathbf{u}_k \quad k \in [1, \quad M] \qquad (24)$$

$$\mathbf{u}_k = \mathbf{u}_{M-1} \quad k \in [M, \quad P-1]$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}$$

In these equations, Φ is the cost function and $\mathbf{d}$ is the difference between measurements and model predictions added at each time step to reduce process-model mismatch. M and P are the control and prediction horizons respectively while $\mathbf{y}_{\mathbf{s}}$ is the set point vector. $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices penalizing output errors and input signal changes respectively. Performing online tuning heuristically, the controller parameters are chosen as sampling time T_s = 0.05 hours, $\mathbf{Q}$ = diag$[100 \quad 50 \quad 100 \quad 100 \quad 100]$, $\mathbf{R}$ = diag$[1 \quad 0.2 \quad 1 \quad 1 \quad 0.6]$, $M = 1$ and $P = 5$ for optimum performance.

Problem 1: Disturbance Rejection

The disturbances that commonly affect the evaporator can be divided into two categories: changes in upstream of the plant, such as feed flowrate, temperature and density, and changes in the heat transfer coefficients of the heaters. To demonstrate its effectiveness, the proposed NMPC is applied to the following case studies (Rangaiah, et al., 2002):

1. Simultaneous changes of an increase in feed flow rate Q_f from the nominal value of 37.7 to 39.7 m^3/h and a 15% decrease in the heat transfer coefficients, UA_1 and UA_2 in heaters 1 and 2.

2. A step decrease in feed density ρ_f from the nominal value of 1.310 to1.245 g/cm^3

3. A step increase in feed temperature T_f from the nominal value of 60 to $66°C$

The NMPC results of the above cases are presented in Figs. 8, 9 and 10 respectively with comparison to those of previously developed decentralized PI controllers. From these, it is seen that the new NMPC scheme performs better than previously developed PI controllers, particularly in terms of settling time.

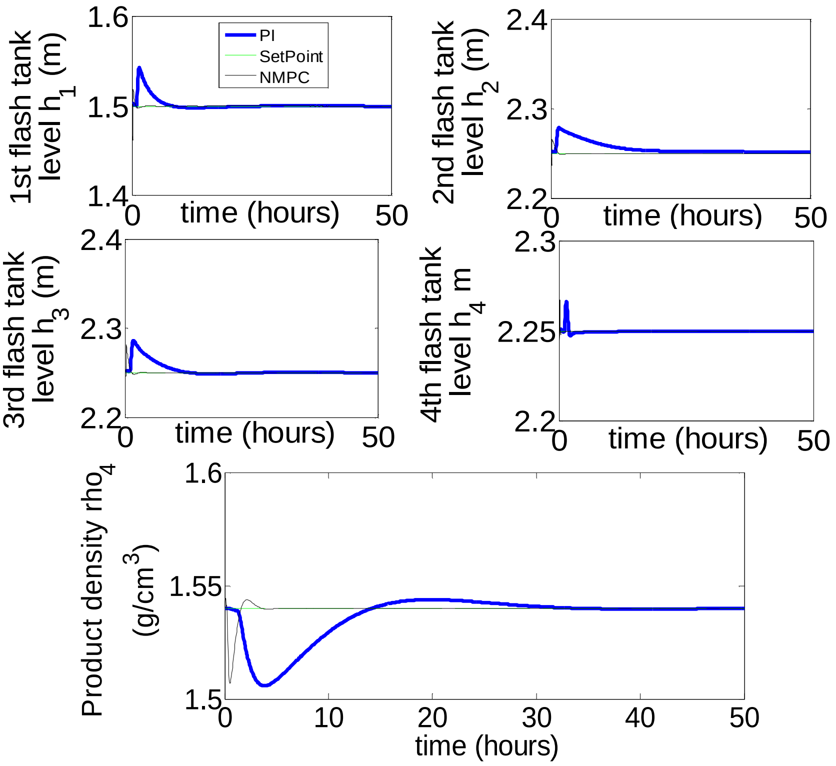

Fig. 8. NMPC & PI Control performance for simultaneous step changes in feed flowrate and heat transfer coefficients of heaters.

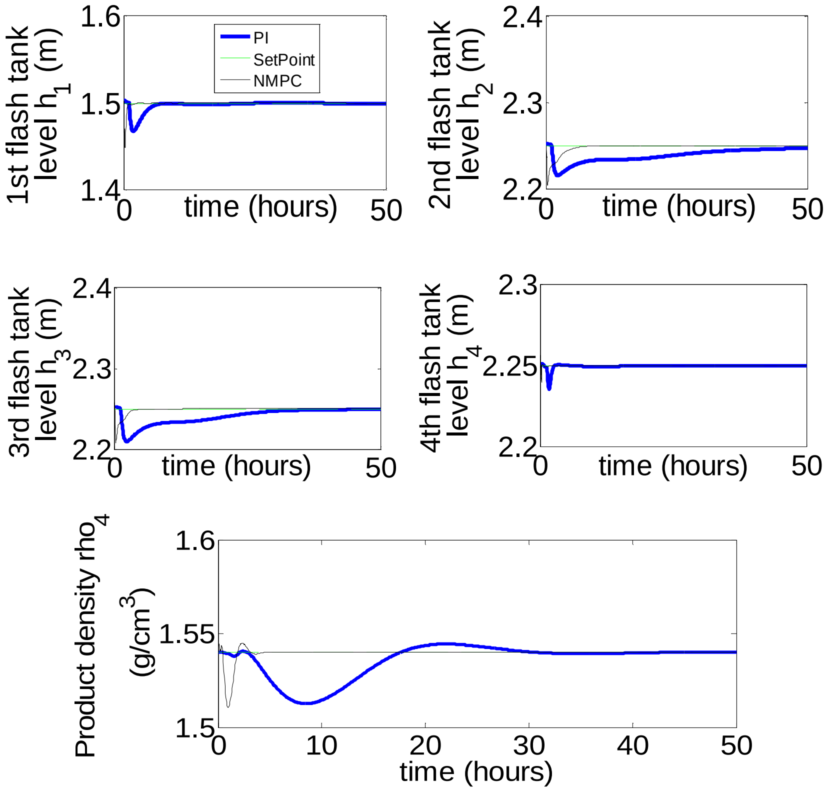

Fig. 9. NMPC & PI Control performance for a step decrease in feed density from 1.310 to 1.245 g/cm^3

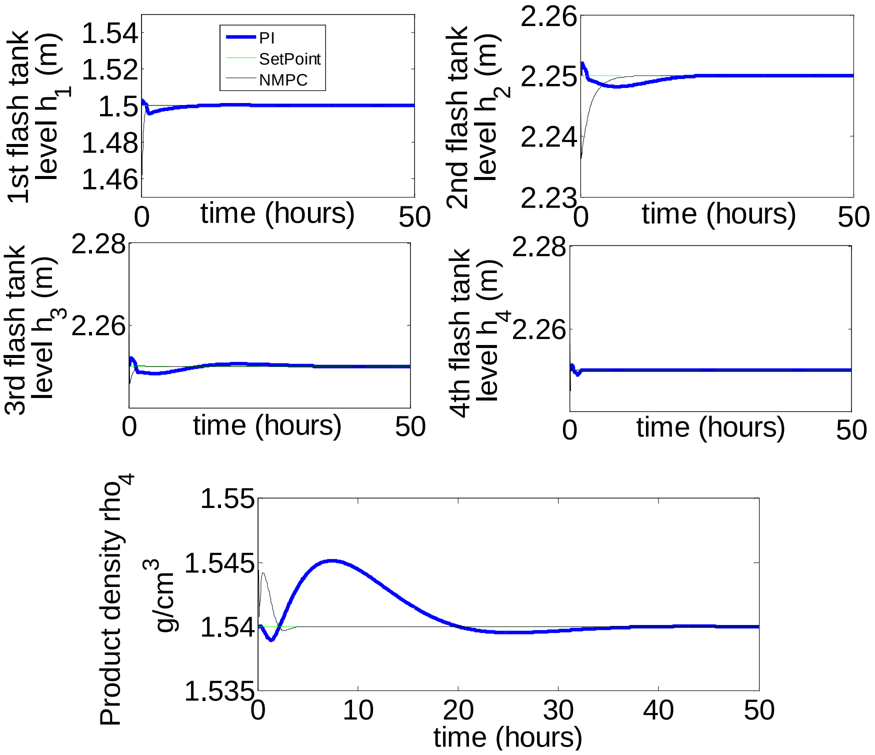

Fig. 10. NMPC & PI Control performance for a step increase in feed temperature from 60 to 66°C .

Problem 2: Set point Tracking

Changes in product specification may necessitate set point tracking. The following servo problems (Rangaiah, et al., 2002) are considered:

A. Step increase in the 4th stage product density ρ_4 from the steady-state value of 1.54 to 1.62 g/cm^3

B. Simultaneous step changes in set points of the 1st flash tank level h_1 from the steady state value of 1.5 to 1.8 m and of the 4th stage product density ρ_4 from 1.54 to 1.46 g/cm^3

The results of the above two cases presented in Figs. 11 and 12 respectively show superior servo performance of the new NMPC scheme over the PI controllers. Fig. 13 shows the input behaviour for the Case B: Simultaneous step changes in set points of the 1st flash tank level h_1 from the steady state value of 1.5 to 1.8 m and of the 4th stage product density ρ_4 from 1.54 to 1.46 g/cm^3

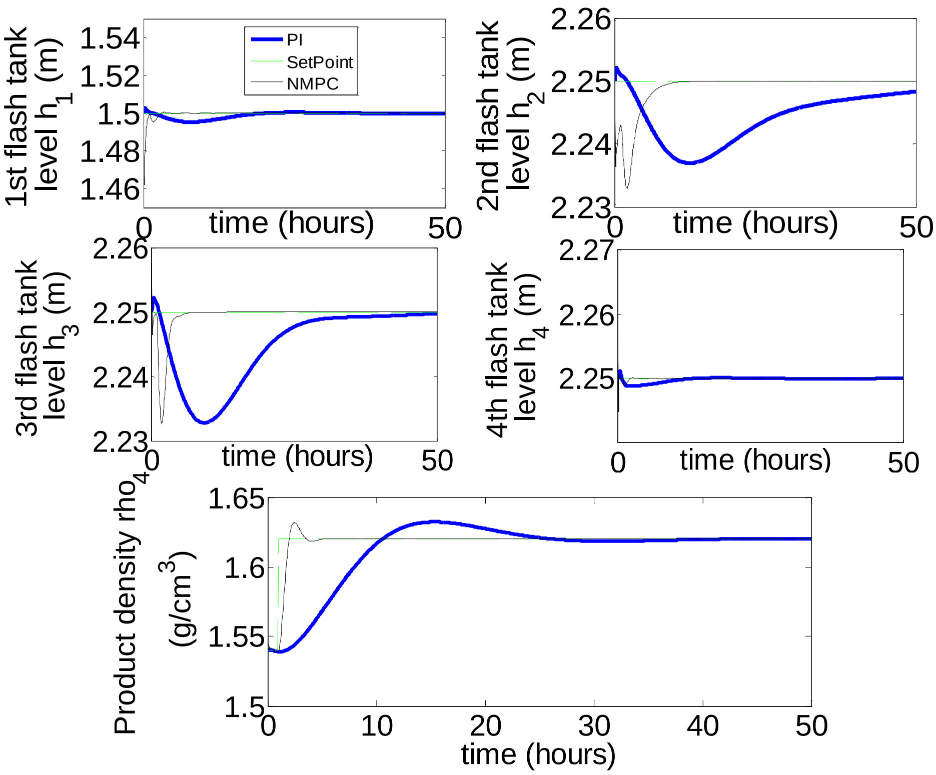

Fig. 11. NMPC &PI Control Performance for a step increase in 4th stage product density from 1.54 to 1.62 g/cm^3.

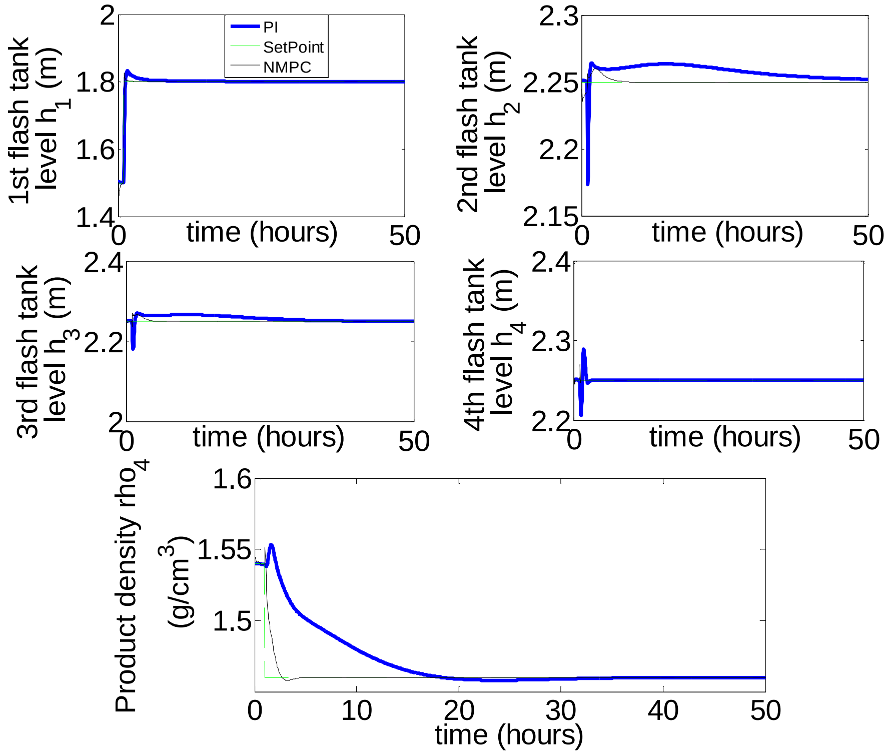

Fig. 12. NMPC &PI Control Performance for simultaneous step changes in 1st flash tank level & 4th stage product density

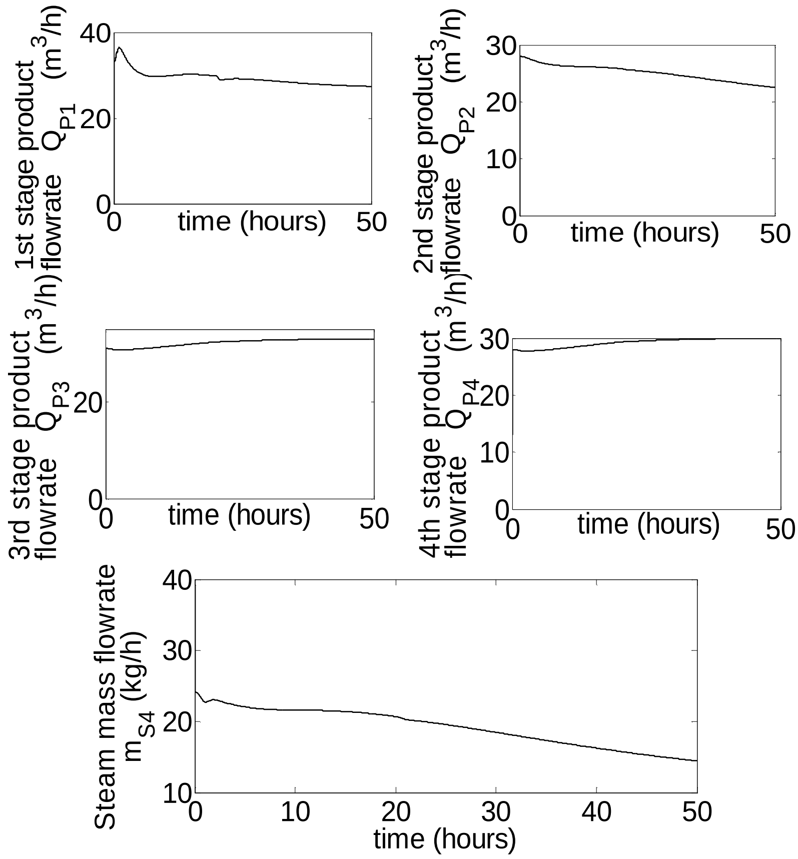

Fig. 13. Plant input behaviour for simultaneous step changes in 1st flash tank level & 4th stage product density

3.1. Quantitative Evaluation of NMPC Performance: Comparative Assessment

The control performance of the newly developed DRNN based NMPC scheme (denoted by NMPCNN in Table 3) is evaluated using the Integral of Squared Errors (ISE) which mathematically is defined as:

$$ISE = \int_{0}^{t_f} \left(\mathbf{y_p}(t) - \mathbf{y_s}(t)\right)^T \left(\mathbf{y_p}(t) - \mathbf{y_s}(t)\right) dt \qquad (25)$$

Table 3. Quantitative Comparison of NMPC & PI Control Performances using Integral of Squared Errors (ISE)

Case	Control	Integral-Squared Errors for each output and Total					
		h_1	h_2	h_3	h_4	ρ_4	Total
1	NMPCNN	2.5E-6	7.5E-5	4.6E-5	8.0E-8	1.3E-4	2.6E-4
	NMPCM2	0.0035	0.0226	6.3E-5	1.0E-5	7.1E-6	0.0616
	PI	0.0026	0.0037	0.0032	1.3E-4	0.0057	0.0153
	LMPC	0.0018	0.0091	0.0027	1.5e-5	0.0037	0.0119
2	NMPCNN	5.6E-6	1.6E-4	9.1E-5	4.9E-8	1.6E-4	3.9E-4
	NMPCM2	0.0240	0.0230	2.3E-4	4.2E-5	1.3E-6	0.0465
	PI	0.0017	0.0091	0.0086	9.1E-5	0.0048	0.0243
	LMPC	0.0012	0.0102	0.0076	2.0E-4	0.3493	0.3685
3	NMPCNN	4.0E-6	1.0E-4	5.5E-5	7.2E-8	1.1E-4	2.7E-4
	NMPCM2	2.4E-4	9.3E-5	1.6E-6	4.7E-7	8.7E-4	3.4E-4
	PI	3.3E-5	2.5E-5	1.8E-5	2.9E-7	1.9E-4	2.6E-4
	LMPC	3.2E-5	3.9E-5	3.0E-5	2.7E-7	6.1E-4	7.1E-4
A	NMPCNN	4.6E-6	1.3E-4	7.4E-5	5.7E-8	1.4E-4	3.5E-4
	NMPCM2	0.0810	0.0940	2.6E-5	2.7E-5	0.0026	0.178
	PI	1.8E-4	0.0025	0.0031	7.0E-6	0.0256	0.0314
	LMPC	0.013	0.0061	0.0058	2.1E-4	0.2991	0.3124
B	NMPCNN	2.2E-6	6.2E-5	3.4E-5	1.0E-7	7.6E-5	1.7E-4
	NMPCM2	0.0850	0.0920	5.6E-5	9.6E-6	0.0020	0.1790
	PI	2.2E-4	0.0042	0.0047	7.3E-6	0.0264	0.0355
	LMPC	0.0023	0.0094	0.0097	3.3E-4	0.5180	0.5397

The results are computed over a simulation time t_f =50 hours and performance compared with those of other control schemes. The control schemes used for comparison are the mechanistic model based NMPC scheme (NMPCM2) developed in Rangaiah et al. (2002), the system of decentralized PI controllers and a linear model predictive control (LMPC) scheme, which has the same parameters as the NMPC but takes the linear model around the operating point as the internal model. The new NMPC scheme is seen to have superior performance as seen in Table 3. The superior performance of the DRNN based NMPC compared to the mechanistic model M2 based NMPC is explained thus. The mechanistic model NMPC was based on a simplified model of the plant, M2 with additional assumptions (see Appendix) which introduce errors, but whose use was justified in that it was simple enough for fast control (Rangaiah et al., 2002). The DRNN was however determined using a highly detailed model M1 (Rangaiah et al., 2002) which better approximates the plant. Further, an NMPC scheme was implemented directly (i.e., without any PI controller) in the earlier study whereas the DRNN-based NMPC is implemented in parallel with PI controllers in the present work. This is another possible reason for the improved performance. However, due to the strong nonlinearity of the process, the LMPC, although it is also configured in parallel with the PI controllers, interferes with the PI control loops, making the performance worse particularly for large set-point tracking cases as indicated in cases A and B (Table 3).

4. Conclusions

In this paper, a continuous-time DRNN model of an industrial, five-stage evaporator is developed, and its suitability for representing the plant nonlinear dynamics demonstrated. An efficient training algorithm based on the Levenberg-Marquardt method and automatic differentiation is applied to the network. The trained network, which shows good generalization properties, has been implemented as the internal prediction model of an NMPC scheme. The developed controller, implemented in parallel with PI controllers, has been tested and proven to perform well in disturbance rejection and set-point tracking with significant improvements over PI controllers, LMPC and a previously developed approximate mechanistic model based NMPC implemented without PI controllers. The parallel NMPC-PI scheme is similar to the well-known combined feed-forward and feedback scheme. In this scheme, the NMPC enhances the PI control through prediction and optimization. The advantage of this scheme is demonstrated by the superior performance of the new NMPC. Nevertheless, the results of the LMPC also indicate that in such a parallel scheme, if the MPC is configured with a poor internal model, it can also severely upset the PI control so that the control performance is significantly deteriorated. Therefore, a good internal model is essential to the success of a parallel NMPC-PI control system.

References

Al-Seyab, R.K and Cao, Y. (2008a). Nonlinear system identification for predictive control using continuous-time recurrent neural networks. *Journal of Process Control*, 18(6), 568-581.

Al-Seyab, R.K and Cao, Y. (2008b). Differential recurrent neural network based predictive control. *Computers and Chemical Engineering*, 32(7), 1533-1545.

Atuonwu, J.C., Cao, Y., Rangaiah, G.P. and Tadé, M.O. (2009). Nonlinear model predictive control of a multistage evaporator system using recurrent neural networks. *Proceedings, IEEE Conference on Industrial Electronics and Applications*, Xian, China, 1662-1667.

Balbis, L., Katebi, R. and Ordys, A. (2006). Model predictive control design for industrial applications. *International Control Conference* (ICC2006), Glasgow, UK.

Bhat, N.V. and McAvoy, T.J. (1992). Determining model structure for neural models by network stripping. *Computers and Chemical Engineering*, 16(4), 271-281.

Bulut, B., Katebi, M.R. and Johnson, M.A. (2000). Industrial predictive control for hot rolling mill co-ordination. *Proceedings of the American Control Conference*.

Cao, Y. (2005). A formulation of nonlinear model predictive control using automatic differentiation. *Journal of Process Control*, 15(8), 851-858.

Cao, Y. and Al-Seyab, R.K. (2003). Nonlinear model predictive control using automatic differentiation, In: *European Control Conference'2003* (CDROM), Cambridge, UK.

Chen, J. and Yea, Y. (2002). Neural network based predictive control for multivariable processes. Chemical Engineering Communications 189, 865-894.

Colannino, J. (2006). *Modeling of Combustion Systems: A Practical Approach*. CRC Press, Taylor and Francis Group, USA

Draeger, A., Engell, S. and Ranke, H. (1995). Model predictive control using neural networks. *IEEE Control Systems Magazine*, 61-66.

Ekawati, E., Bahri, P.A., Lepage, G. and Noraii, A. (2003). Controllability analysis of industrial five-effect evaporator system. *The Foundations of Computer Aided Process Operations (FOCAPO) 2003 Conference*, Coral Springs, Florida, US.

Funahashi, K.L. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous-time recurrent neural networks. *Neural Networks*, 6, 183-192.

Galvan, I.M. and Zaldivar, J.M. (1998). Application of recurrent neural networks for nonlinear inverse and predictive control of the heat transfer fluid temperature in a batch reactor. *Chemical Engineerng and Processing*, 37, 149-161.

Goldberge, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addision-Wesley.

Harnischmacher, G. and Marquardt, W. (2007). Nonlinear model predictive control of multivariable processes by block-structured models. *Control Engineering Practice*, 15, 1238-1256.

Henson, M.A. (1998). Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering* 23(2), 187-202.

Hush, D.R. and Horne, B.G. (1993). Progress in supervised neural networks. *IEEE Signal Processing Magazine*, 10(1), 8-39.

Jazayeri-Rad, H. (2004). The nonlinear model predictive control of a chemical plant using multiple neural networks. *Neural Computers and Applications* 13, 2-15.

Kam, K.M., Saha, P., Tadé, M.O. and Rangaiah, G.P. (2001). Models of an industrial evaporator for education and research in process control *Dev. Chem. Eng. Mineral Process*, 10, 105-127.

Kam, K.M. and Tadé, M.O. (1997). *Dynamic modeling and geometric analysis of an industrial multi-stage evaporation unit, Technical Report 1/97*. School of Chemical Engineering, Curtin University of Technology, Western Australia.

Kambhampati, C., Craddock, R. J., Tham, M. and Warwick, K. (2000), Inverse model control using recurrent networks, *Mathematics and Computers in Simulation*, 51, 181–199.

Leonard, J. A., & Kramer, M. A. (1990). Improvement of the back-propagation algorithm for training neural networks. *Computers and Chemical Engineering*, *14*, 337–341.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11, 431–441.

Mjalli, F.S. and Hussain, M.A. (2009). Approximate predictive versus adaptive control strategies of biodiesel reactors. *Industrial Engineering Chemistry Research*, 48 (24), 11034-11047.

Mohanty, S. (2009). Artificial neural network based system identification and model predictive control of a flotation column. *Journal of Process Control*, 19, 991-999.

Nagy, Z.K. (2007). Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks. *Chemical Engineering Journal*, 127, 95-109.

Nagy, Z., Mahn, B., Franke, R. and Allgöwer, F. (2007). Evaluation study of an efficient output feedback nonlinear model predictive control for temperature tracking in an industrial batch reactor. *Control Engineering Practice* 15, 839-850.

Nørgaard, M., Ravn, O., Poulsen,N.K. and Hansen, L.K. (2000). *Neural networks for modeling and control of nonlinear systems*. Springer-Verlag London Ltd., London.

Norquay, S. J., Palazoglu, A., & Romagnoli, J. A. (1998). Model predictive control based on Wiener models. *Chemical Engineering Science*, 53(1), 75–84.

Ou, J. and Rhinehart, R.R. (2002). Grouped neural network modeling for model predictive control. *ISA Transactions*, 41, 195-202.

Pearson, R.K. and Ogunnaike, B.A. (1997). Nonlinear process identification, In: *Nonlinear process control*, M.A. Henson and D.E. Seborg Eds. New Jersey: Englewood Cliffs, Prentice-Hall.

Piché, S., Sayyar-Rodsari, B., Johnson, D. and Gerules, M. (2000). Nonlinear model predictive control using neural networks. *IEEE Control Systems Magazine*, 20(3), 53-62.

Puskorius, G.V. and Feldkamp, L.A. (1994). Neurocontrol of nonlinear dynamic systems with Kalman filter trained recurrent neural networks. *IEEE Transactions on Neural Networks* 5(2), 279-297.

Qin, J.S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733-764.

Rangaiah, G.P., Saha, P. and Tadé, M.O. (2002). Nonlinear model predictive control of an industrial four-stage evaporator system via simulation. *Chemical. Engineering. Journal*, 87(3), 285-299.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing*. Cambridge MA: MIT Press.

Sentoni, G.B., Biegler, L.T., Guiver, J.B. and Zhao, H. (1998). State space nonlinear process modeling: identification and uncertainty. *AIChE Journal*, 44(10), 2229-2239

Schenker, B. and Agarwal, M. (2000). Online-optimized feed switching in semi-batch reactors using semi-empirical dynamic models. *Control Engineering Practice*, 8, 1393-1403.

Sjoberg, J., Zhang., Q., Ljung, L., Benveniste, A., Delyon., B., Glorennec, Y. Hjalmarsson., H. and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified review. *Automatica* 31(12), 1691-1724.

Su, H.T. and McAvoy, T.J. (1997). Artificial neural networks for nonlinear process identification, In: *Nonlinear process control*, M.A. Henson and D.E. Seborg Eds. New Jersey: Englewood Cliffs, Prentice-Hall.

Temeng, K.O., Schenelle, P.D. and McAvoy, T.J. (1995). Model predictive control of an industrial packed reactor using neural networks. *Journal of Process Control* 5(1), 19-28.

Wanas, N., Auda,G., Kamel, M.S. and Karray, F. (1998). On the optimal number of hidden nodes in a neural network. *IEEE Canadian Conference on Electrical and Computer Engineering*, 2, 918-921.

Werbos, P.J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the. IEEE*, 78(10), 1550-1560.

Yu, D.L. and Gomm, J.B. (2003). Implementation of neural network predictive control to a multivariable chemical reactor. *Control Engineering Practice*, 11, 1315-1323.

Zamarreno, J.M. and Vega, P. (1998). State-space neural network properties and applications. *Neural Networks* 11, 1099-1112.

Zhu, X., & Seborg, D. E. (1994). Nonlinear predictive control based on Hammerstein models. *Proceedings: Process Sysems Engineering*, 995–1000.

Appendix 1: Mechanistic models of the evaporator

Model M1 for Stage 1

$$\frac{dh_1}{dt}=\frac{1}{A_1}\left[Q_f-Q_{P1}-\frac{E_1}{\rho_w}\right]$$

$$\frac{d\rho_1}{dt}=\frac{1}{h_1A_1}\left[E_1\left(\frac{\rho_1}{\rho_w}-1\right)-Q_f\rho_f\left(\frac{\rho_1}{\rho_f}-1\right)\right]$$

$$\frac{dT_1}{dt}=\frac{1}{CoT_1}\left[\frac{1}{(V_1-h_1A_1)}\left(E_1-m_{V1}+\rho_{V1}\left(Q_f-Q_{P1}-\frac{E_1}{\rho_w}\right)\right)\right]$$
$$-\frac{1}{CoT_1}\left[CoRho_1\left(\frac{1}{h_1A_1}\left(E_1\left(\frac{\rho_1}{\rho_w}-1\right)-Q_f\rho_f\left(\frac{\rho_1}{\rho_f}-1\right)\right)\right)\right]$$

where

$$E_1=\frac{Q_f\rho_f c_f-Q_{P1}\rho_1c_1T_1+m_{S1}\lambda_{S1}}{\lambda_{V1}}$$

$$m_{S1}=m_{V2}$$
$$m_{V1}=E_1$$
$$BPE_1=102.69\rho_1-128.82$$
$$\rho_{V1}=MP_1/R(273.1+T_1-BPE_1)$$
$$P_1=0.133\exp\left(A-\frac{B}{C+273.1+T_1-BPE_1}\right)$$

$$CoT_1=\rho_{V1}\left(\frac{B}{(C+273.1+T_1-BPE_1)^2}-\frac{1}{273.1+T_1-BPE_1}\right)$$

Model M2 for Stage 1

$$\frac{dh_1}{dt}=\frac{1}{A_1}\left[Q_f-Q_{p1}-\frac{E_1}{\rho_w}\right]$$

$$\frac{d\rho_1}{dt} = \frac{1}{h_1 A_1}\left[E_1\left(\frac{\rho_1}{\rho_w} - 1\right) - Q_f \rho_f \left(\frac{\rho_1}{\rho_f} - 1\right)\right]$$

$$\frac{dT_1}{dt} = \frac{1}{CoT_1}\left[\frac{1}{(V_1 - h_1 A_1)}\left(E_1 - m_{V1} + \rho_{V1}\left(Q_f - Q_{P1} - \frac{E_1}{\rho_w}\right)\right)\right]$$

$$P_1 = 0.75T_1 - 33.96$$

$$CoT_1 = (V_1 - A_1 h_1)\left(\frac{0.75M}{R(273.1 + T_1 - BPE_1)} - \frac{\rho_{V1}}{273.1 + T_1 - BPE_1}\right)$$

Table A1 Assumptions made in the derivation of models (Kam and Tadé, 1997)

Assumptions for both Models M1 and M2	*Additional assumptions for Model M2*
1.Dynamics of instrumentation and control valves are very fast and negligible when compared to the dynamics of flash tanks	8. Liquor boiling point elevation in each flash tank remains constant
2. Liquor density and temperature in each flash tank are respectively the same as those of corresponding discharged stream since perfect mixing is usually achieved in each flash tank	9. Vapour pressure-temperature relation of flashed vapour is linear
3. Specific heat capacities of all process streams remain constant	10. Latent heat of vaporization of liquor remains constant
4. Effect of falling film on the heat transfer rate and the dynamics of heater discharge temperatures are neglected	11. No accumulation of vapour in the flash tanks and the steam space in the shell side of the heaters remains unchanged.
5. Liquor in each flash tank is in equilibrium with the corresponding flashed vapour	
6. Heat losses from the evaporator system are negligible	
7. No flashing of liquor in the heaters	