# Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 9 (18): 3219-3235, 2009
ISSN 1812-5654


# Adaptive Feedback Linearization Control of Nonlinear Processes using Neural Network Based Approaches

[1,2]F. Hourfar and [1]K. Salahshoor
[1]Tehran Petroleum College, Petroleum University of Technology, Tehran, Iran
[2]Petropars Ltd., Petroleum University of Technology, Iran

**Abstract:** In this study, two different techniques for adaptive control of nonlinear chemical processes based on feedback linearization method are presented. The first technique utilizes a black-box modeling approach to completely model an unknown plant by an adaptive neural network whereas, the second technique focuses only on the difficult-to-model part or complicated part of the plant to identify a semi-mechanistic or grey-box model using an adaptive neural network. The remaining parts of the plant dynamics are obtained online using the combined first-principle model and special measuring methods. The performances of both adaptive control techniques have been demonstrated on a well-known Continuous Stirred Tank Reactor (CSTR) benchmark process to investigate their comparative capabilities.

**Key words:** Grey box modeling, black box modeling, CSTR, intelligent control, input-output linearization, adaptation techniques, NARMA- L2, semi-mechanistic, PID controller

## INTRODUCTION

Many chemical processes exhibit significant nonlinear behavior dynamic. If these processes are operated at a nominal steady state, the effects of the nonlinearities may not be severe and hence traditional control schemes based on local linearized models can provide satisfactory control performance. However, if the processes are required to work over a wide range of conditions, conventional linear control approaches cannot handle the system nonlinearities. Under such situations, closed-loop stability can be guaranteed only when the controllers are sufficiently detuned, leading to degradation in closed-loop performance (Henson and Seborg, 1997; Zhang and Guay, 2005; Fourati *et al*., 2008).

A reasonable mathematical model with good estimated parameters is essential for designing a high-performance control system. However, in modeling a chemical process, there unavoidably exist uncertainties due to poor process knowledge, nonlinearities, unmodeled dynamics, unknown internal or external noises, environmental influence and time varying parameters. The presence of uncertainties and parameters changes can make a mismatch between the formulated mathematical model and the true process. This degrades the control performance and may lead to serious stability problems especially when the process is nonlinear. Therefore, it is a great challenge and of highly importance for control engineers to design robust and adaptive controllers for nonlinear processes subject to model uncertainties and parameters changes (Chen and Dai, 2001).

Control schemes based on feedback linearization technique provide larger dynamic operation range than the conventional Jacobian linearization method which is based upon an operating point. Furthermore, the benefits of linear control techniques can be utilized via feedback linearization (Chen and Dai, 2001).

In recent years, many interesting results for chemical process control have been reported in the literature on the basis of feedback linearization scheme (Zhang and Guay, 2005; Henson and Seborg, 1990; Lee and Sullivan, 1988; Kravaris and Chung, 1987). These feedback linearization strategies often require exact mathematical models of the plant dynamics. However, it is generally difficult in practice to obtain an accurate model because of the inherent complexity of the chemical processes or the lack of a priori informative process knowledge. Adaptive control scheme is a viable choice to deal with such uncertainties which has drawn a great deal of interest. The conventional adaptive control, however, is limited to linearly parameterized model uncertainty. The limitation can be overcome by introducing neural network as a black box modeling tool to tackle with nonlinear parameterizing uncertainty (Zhang and Guay, 2005; Henson and Seborg, 1994; Marino and Tomei, 1995; Kar and Behera, 2009).

**Corresponding Author:** Farzad Hourfar, No. 3, 4th Lane, Kouye Parvaneh, Jalale Ale Ahmad Avenue, Tehran, Iran
Tel: +98-912-5715925 Fax: +98-21-88005596

The study presents two different techniques for adaptive control of nonlinear chemical processes via combining feedback linearization and a Neural Network (NN) methodology to cope with nonlinearity and uncertainty. The first technique utilizes an adaptive NN as a black box to completely model an unknown chemical process. While, the second technique incorporates both a priori process knowledge and an adaptive NN to identify the difficult-to-model part of the process dynamic, leading to a semi-mechanistic or grey-box model representation.

For this purpose, a Globally Linearizing Control (GLC) approach based on an input-output linearization technique is derived for a CSTR benchmark process and the resulting control structure is incorporated with some adaptation mechanisms including NN to evaluate its performances under different conditions. The final section gives the concluding remarks derived from this study.

## CONVENTIONAL FEEDBACK LINEARIZATION SCHEME

In the case of continuous-time processes, there is a vast amount of theory for developing feedback linearization of affine processes, where the process input (u) appears in linear form in the state space equation as follows:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned} \tag{1}$$

where, $x = [x_1,...,x_n]^T$ is the state vector, u is the manipulated input and y is the controlled output of the process. In general, f, g and h are assumed to represent smooth vector fields.

The objective of input-output linearization is to obtain a nonlinear control law in the following form:

$$u = p(x) + q(x)v \tag{2}$$

in such a way that the resulting controlled process, shown in Fig. 1, be linear. So, the input-output linearization results in a linear transfer function between v and y:

$$\frac{y(s)}{v(s)} = \frac{1}{\beta_r s^r + \ldots + \beta_1 s + \beta_0} \tag{3}$$

where, r denotes the relative degree of the nonlinear system. This relative degree at the operating point $x_0$ is defined by the integer r which satisfies:

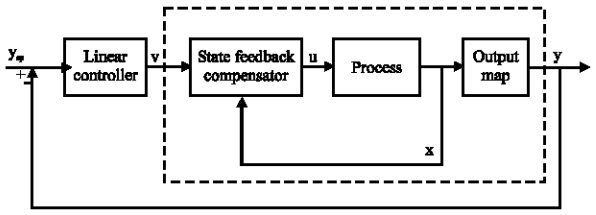


Fig. 1: Schematic block diagram of a GLC scheme

$$\begin{aligned}&L_g L_f^{i-1}h(x) = 0 \;\forall i<r \text{ and } x \text{ near } x_0 \\ &L_g L_f^{r-1}h(x) \neq 0 \;\forall\, x \text{ near } x_0\end{aligned} \tag{4}$$

where, $L_g$ and $L_f$ are Lie derivatives defined as:

$$L_f h(x) = \sum_{i=1}^{n} f_i(x)\frac{\partial h}{\partial x_i}(x) \tag{5}$$

$$L_g L_f h(x) = \sum_{i=1}^{n} g_i(x)\frac{\partial L_f h}{\partial x_i}(x) \tag{6}$$

While, the higher-order Lie derivatives can be written as:

$$L_f^{\alpha}h(x) = L_f(L_f^{\alpha-1}h) \tag{7}$$

The time derivatives of the system output can be expressed as algebraic functions of these Lie derivatives:

$$\begin{aligned}\frac{dy}{dt} &= L_f h(x) \\ &\vdots \\ \frac{d^{r-1}y}{dt^{r-1}} &= L_f^{r-1}h(x) \\ \frac{d^r y}{dt^r} &= L_f^r h(x) + L_g L_f^{r-1}h(x)u\end{aligned} \tag{8}$$

The preceding equations show that the relative degree (or relative order) represents how many times the output must be differentiated with respect to the time to recover explicitly the input u.

The GLC is an input-output linearization technique for processes of arbitrary relative degree. From the Eq. 8, the feedback control law can be expressed as:

$$u = \frac{v - \sum_{k=0}^{r}\beta_k L_f^k h(x)}{\beta_r L_g L_f^{r-1}h(x)} \tag{9}$$

Figure 1 shows that the scheme of GLC contains a linear feedback controller which controls the linearized system. In most process control applications, the

objective is to maintain the output at a non-zero set-point despite the presence of model error and unmeasured disturbances. Consequently, the linear controller is usually a PI or a PID controller (McLellan *et al*., 1990). If, for example, a PI controller is applied as:

$$v = K_c\left((y_{sp} - y) + \frac{1}{T_i}\int_{\tau=0}^{t}(y_{sp} - y)d\tau\right) \tag{10}$$

where, the gain $K_c$ and the time constant $T_i$ are additional controller tuning parameters, then the complete GLC control law yields the following closed-loop transfer function for set-point changes:

$$\frac{y(s)}{y_{sp}(s)} = \frac{K_c s + \frac{K_c}{T_i}}{\beta_r s^{r+1} + \beta_{r-1}s^r + \cdots + (\beta_0 + K_c)s + \frac{K_c}{T_i}} \tag{11}$$

Certainly, one can use other linear controller, especially for systems that have high relative degree. If the relative order of the system is 1 or 2, the PID controller is a good choice; but for higher relative orders, it may be more useful to design the linear controller directly from the linearized system.

## FEEDBACK LINEARIZATION TECHNIQUE USING BLACK-BOX NEURAL NETWORK MODEL

In case the process is unknown, a model can be estimated from historical input-output data by letting two separate neural networks approximate the functions f and g as follows:

$$\begin{aligned}\hat{y}(k|\theta) = \hat{f}[y(k-1),\cdots,y(k-n),u(k-2),\cdots u(k-m),\theta_f] \\ + \hat{g}[y(k-1),\cdots,y(k-n),u(k-2),\cdots u(k-m),\theta_g]u(k-1)\end{aligned} \tag{12}$$

or

$$\hat{y}(k|\theta) = \hat{f}[\varphi(k),\theta_f] + \hat{g}[\varphi(k),\theta_g]u(k-1) \tag{13}$$

Where:

$$\begin{aligned}\varphi(k) = [&y(k-1),....,y(k-n_a) \\ &u(k-n_k),....,u(k+1-n_b-n_k)]^T\end{aligned} \tag{14}$$

where, $\theta$ indicates a vector containing the weights and biases, $\varphi$ is the regression vector and $\hat{f}$ and $\hat{g}$ are the estimated nonlinear functions used to predict the output, $n_a$ and $n_b$ denote the number of past outputs and inputs, respectively, used to determining process output prediction and nk is time delay.

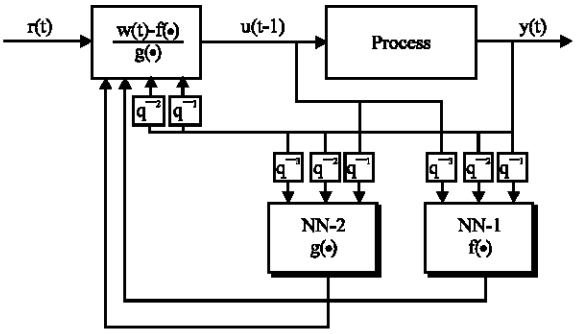


Fig. 2: Discrete input-output feedback linearization with neural networks

The closed-loop system consisting of controller and process to be controlled is shown in Fig. 2. Derivation of a training method for determination of the weights in the two neural networks used for approximating f and g in Eq. 1 is straightforward.

The prediction error approach requires knowledge of the derivative of the model output with respect to the weights. In order to calculate this derivative, the derivative of each network output with respect to the weights in the respective network must be determined as:

$$\begin{aligned}\psi_f(k,\theta_f) &= \frac{\partial\hat{f}}{\partial\theta_f} \\ \psi_g(k,\theta_g) &= \frac{\partial\hat{g}}{\partial\theta_g}\end{aligned} \tag{15}$$

The derivative of the model output with respect to the weights is often composed of derivatives of each network in the following manner:

$$\psi(k,\theta) = \frac{\partial\hat{y}(k|\theta)}{\partial\theta} = \begin{bmatrix}\psi_f(k,\theta_f) \\ \psi_g(k,\theta_g)u(k-1)\end{bmatrix} \tag{16}$$

With this derivative in hand, any of the training methods can be used without further modification. Any smooth nonlinear function can be chosen for approximating the unknown functions f ($\varphi$(k), $\theta$) and g ($\varphi$(k), $\theta$). In this study, two feedforward NNs with a hidden layer including nonlinear function (i.e., tanh(.)) have been selected for this purpose. The resulting process model is known as nonlinear autoregressive moving average model (NARMA-L2) which can be expressed by Narendra and Mukhopadhyay (1997):

$$\begin{aligned}&\hat{y}(k) = f^2(a^2) + g^2(a^2)u(k-1) = \\ &f^2(W_f^2a^1) + g^2(W_g^2a^1)u(k-1) = \\ &f^2(W_f^2f^1(W_f^1a^0)) + g^2(W_g^2g^1(W_g^1a^0))u(k-1)\end{aligned} \tag{17}$$

where, $f^1(x) = g^1(x) = \tanh(x)$ and $f^2(x) = g^2(x) = x$ represent activation functions of the hidden layer and the output layer, respectively. The quantities $a^0$, $a^1$, $a^2$ are the outputs of input, hidden and output neurons, respectively. The weights of hidden and output layers are denoted by $W^1_f$, $W^1_g$ and $W^2_f$, $W^2_g$, respectively.

It's highly recommended to remove the mean and scale of all the measured signals to the same variance (usually zero mean and variance 1). Because, the signals are likely to be measured in different physical units and without scaling, there is a tendency that the signal of largest magnitude will be too dominating. Moreover, scaling makes the training algorithm numerically robust and leads to a faster convergence. Since, the network model is a three-layer neural network with linear output units, it is straightforward to rescale the weights after completion of the training session. In this way, the final network model can work on un-scaled data.

Neural network learning speed is very important for real time system identification. To realize fast learning, a recursive Levenberg-Marquardt minimization method is used in this study. It is an intermediate method between the steepest descent and Gauss-Newton, having good convergence properties. The online training algorithm is derived so as to minimize the following criterion:

$$F(\theta_k) = \hat{e}^2(k) = (\hat{y}(k) - y_r(k))^2 \quad (18)$$

where, $\hat{e}(k)$ denotes the prediction error, $y_r(k)$ is the actual plant output and $\theta_k$ includes all the neural network weights. The NN weights are adjusted as follows (Hagan *et al*., 1996):

$$\theta_{k+1} = \theta_k - \Delta\theta_k = \theta_k - [G(\theta_k)G^T(\theta_k) + \lambda_k I]^{-1} G(\theta_k)e(\theta_k) \quad (19)$$

where, G is the Jacobian matrix i.e., $(\partial F(\theta_k)/\partial\theta_k)$ and I is the identity matrix and $\lambda_k$ is a constant. This algorithm reduces to the steepest descent method with small learning rate as $\lambda_k$ is increased. If $\lambda_k$ is decreased to zero, the algorithm approaches the Gauss-Newton method. Thus, the algorithm provides a nice compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

**Control of a CSTR benchmark process with online feedback linearization technique:** Figure 3 shows the schematic of a cooled exothermic CSTR benchmark process. The reaction is first order in reactant A. A well-mixed cooling jacket surrounds the reactor to remove the heat of reaction. Cooling water is added to the jacket at a rate of $F_j$ and an inlet temperature of $T_{j0}$. The volume V of the reactor contents and the volume $V_j$ of water in the jacket are both assumed to be constant. So, it can be written:

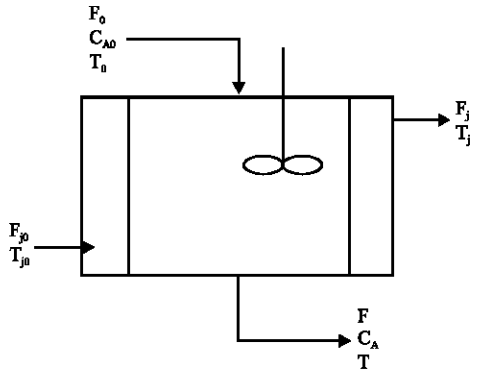


Fig. 3: Schematic of CSTR

Table 1: Nominal values of the model parameters

| Notations | Description | Values and unit |
|---|---|---|
| $F_0$ | Feed flowrate | 0.2 $m^3$ $min^{-1}$ |
| V | Reactor volume | 2 $m^3$ |
| $K_0$ | Reaction rate coefficient | $3.5\times10^5$ $min^{-1}$ |
| $E_a$ | Activation energy | 49.884 kJ $mol^{-1}$ |
| R | Ideal gas constant | $8.313\times10^{-3}$ kJ $mol^{-1}$ |
| $H_r$ | Heat of reaction | 500 kJ $mol^{-1}$ |
| $C_{A0}$ | Concentration of A in feed | 1000 mol $m^3$ |
| $T_0$ | Feed temperature | 30°C |
| $\rho$ | Density of solution | 1000 kg $m^{-3}$ |
| $c_P$ | Heat capacity of solution | 4.2 kJ $kg^{-1}$ °C |
| $U_A$ | Heat transfer coefficient (surface) | 252 kJ $min^{-1}$ °C |
| $V_j$ | Jacket volume | 0.4 $m^3$ |
| $T_{jf}$ | Inlet temperature of coolant | 10°C |

$$\begin{aligned} u &= F_j = F_{j0} \\ F_0 &= F \end{aligned} \quad (20)$$

The first-principles model of the CSTR process is:

$$\begin{aligned} \frac{dC_A}{dt} &= \frac{F_0}{V}(C_{A0} - C_A) - k_0 \exp\left(\frac{-E_a}{RT}\right)C_A \\ \frac{dT}{dt} &= \frac{F_0}{V}(T_0 - T) - \frac{H_r}{\rho c_p V}k_0 \exp\left(\frac{-E_a}{RT}\right)C_A + \frac{UA}{\rho c_p V}(T_j - T) \\ \frac{dT_j}{dt} &= \frac{UA}{\rho c_p V_j}(T - T_j) + \frac{u}{V_j}(T_{j0} - T_j) \end{aligned} \quad (21)$$

where, CA (mol $m^{-3}$) is the concentration of the A component in the reactor, T(°C) is the temperature of the reactor and $T_j$(°C) is temperature of the reactor jacket, while the process input is the flow rate of the cooling water u ($m^3$ $min^{-1}$). The controlled output of the process is the reactor temperature. The parameters and their nominal values of the model are shown in Table 1.

Because of the CSTR nature, the model adaptation is completely necessary in system identification. It is very common that one of the parameters of the process like feed flow rate, feed temperature, concentration of components in feed and inlet temperature of coolant change with time. Thus, by incorporating them in online process identification, the controller can compensate for these changes.

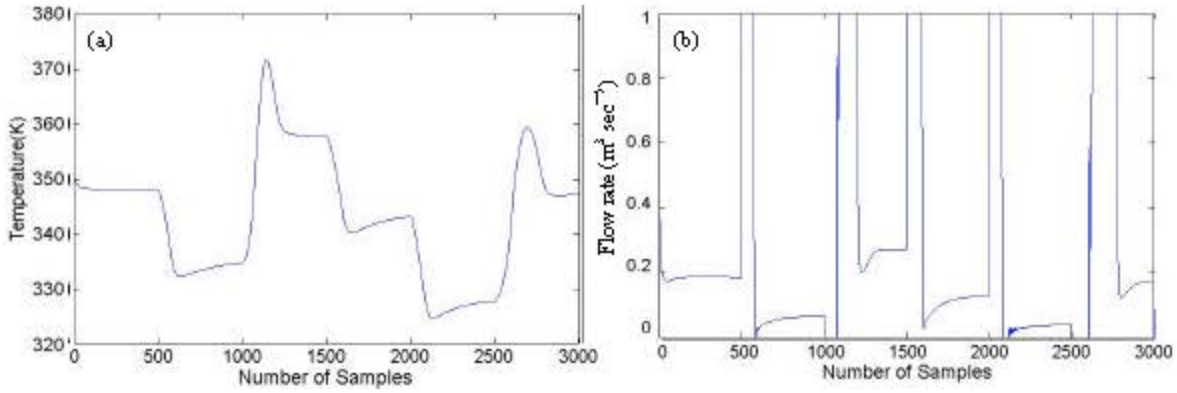


Fig. 4: Unscaled (a) output and (b) input data

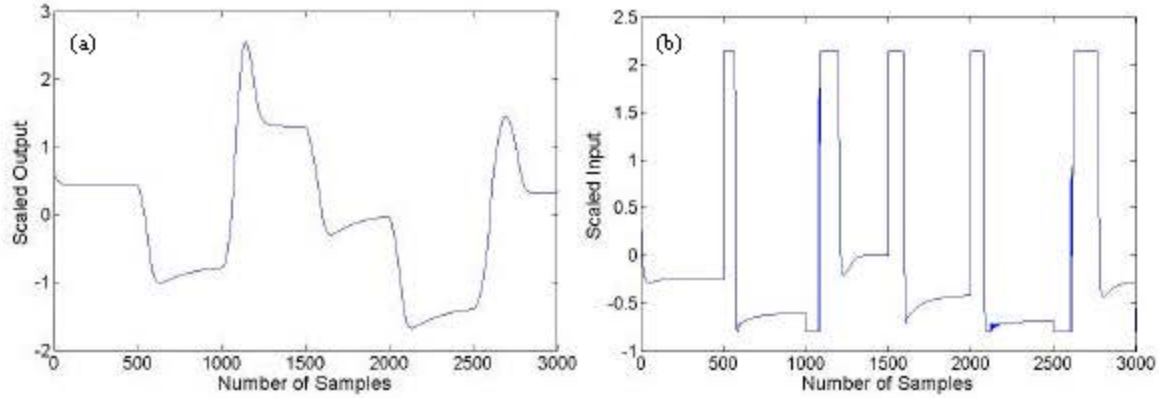


Fig. 5: Scaled (a) output and (b) input data

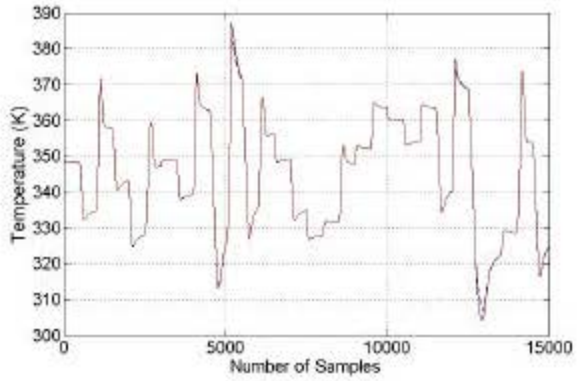


Fig. 6: Model's output and real output for a set of test data. Solid line: output, dashed line: one step ahead prediction

In the first stage, input-output data should be generated for obtaining the NARMA-L2 model of the process (Fig. 4a, b). It is clear that more information about the system is obtained by using a very long and powerful input signal. This is relevant for industrial processes which typically have slow dynamics and high level disturbances. On the other hand, the cost of the experiment becomes low by keeping the experiment time short and the signals small. From an industrial and economical perspective, the test data must lead to a suitable model within an acceptable time period in order to deviate as little as possible from normal operation. The inputs of the NARMA-L2 were selected as y(k-1), y(k-2), u(k-1) and u(k-2).

It should be mentioned here that for making the training algorithm numerically robust and having a faster convergence, the input-output data have been scaled to zero mean and variance 1 (Fig. 5a, b).

Figure 6 shows the model's output and the real process output for a set of test data, while Fig. 7 shows the error between the model's output and real process output. It can be seen that the validity of the NARMA-L2 model is acceptable.

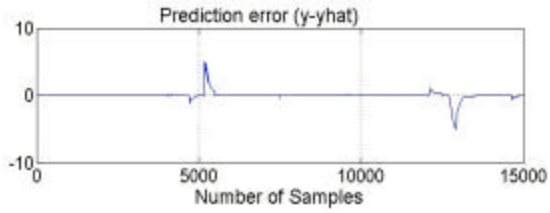


Fig. 7: Error between model's output and real output

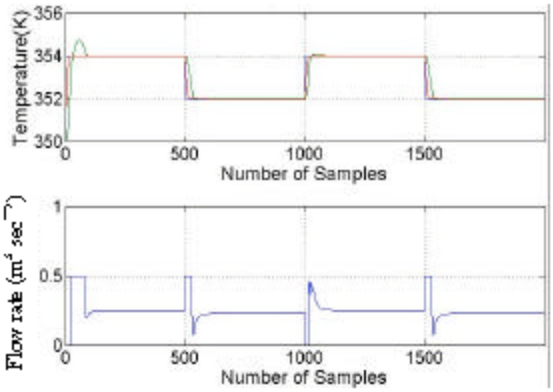


Fig. 8: Set-point, Plant Output, Desired Polynomial Output and flow rate

In this case, the desired polynomial has been selected to corresponding to two stable poles.

Figure 8 shows he resulting set-point tracking performance of the CSTR process.

## FEEDBACK LINEARIZATION TECHNIQUE USING NEURAL NETWORK AND SEMI-MECHANISTIC MODEL

Neural Networks (NN's) provide suitable tools to model plant uncertainties that are in form of unknown functions if appropriate input/output data is available (Widrow *et al*., 1994; Bazaei and Majd, 2003). In Nikolaou and Hanagandi's (1993) contribution, feedback linearization has been applied to continuous-time recurrent NN's. In Braake *et al*. (1998) study, some data-based Exact Feedback Linearization (EFL) and Approximate Feedback Linearization (AFL) schemes in discrete-time via NN's have been developed. In the feedback linearization mentioned in the previous section, the NN's are used as a black-box model of the plant and no part of the plant dynamic equations are assumed a priori known. Hence, the black-box NN model may have poor extrapolation properties and its validity may remain within the range of the training data.

The models totally generated by the first principles knowledge o the basis of general physical rules governing the plants are called white-box models. Although, these models have good extrapolation properties, their generation, if possible, is expensive in general (Van Can *et al*., 1996).

It may possible to generate models based on a combination of first principles knowledge of the plant and neural, fuzzy, or other types of models, resulting in what is called first-principles-based gray box, hybrid, semi-mechanistic, or semi-physical models.

The grey-box models can be classified into parallel and serial types. In the parallel grey-box models discussed by Lee *et al*. (2002), Cote *et al*. (1995) and Su *et al*. (1992), the NN's are placed in parallel to the first principles models. This type may have better interpolation properties compared to the black-box one. In serial gray-box models, the NN is trained to model the unknown or uncertain part of the plant dynamics. The serial type may yield better dimensional extrapolation property than that of the black-box and the parallel grey-box types. However, when the available mechanistic model, derived from the process knowledge, is not sufficiently accurate, the parallel hybrid-modeling scheme may exhibit better extrapolation, we mean that some variables or parameters of the plant, which have been fixed during model training, are allowed to change during the use of the model without need for re-training (Van Can *et al*., 1996).

Because in serial gray-box modeling the identification effort is only on the known part of the plant, the training time as well as the modeling error will decrease while the validity domain (i.e., the range extrapolation) of the resulting model will improve compared to those of the black-box model.

By the proposed method we take the advantages of feedback linearization technique, first principles knowledge and grey-box neural modeling to improve the control performance.

The underlying philosophy of semi-mechanistic modeling is that black-box models, like neural networks, can be used to represent the otherwise difficult-to-obtain parts of first-principle models. Fortunately, a first-principle model can be easily extended by exchanging parts of the model. In chemical engineering, first-principles models are mostly derived from dynamic mass, energy and momentum balances. These balances are based on conservation principle and for lumped process systems; they lead to differential equations formulated as:

$$\begin{bmatrix}\text{Accumulation}\\ \text{of } x_i\end{bmatrix} = \begin{bmatrix}\text{Inflow}\\ \text{of } x_i\end{bmatrix} - \begin{bmatrix}\text{Outflow}\\ \text{of } x_i\end{bmatrix} + \begin{bmatrix}\text{Amount of}\\ x_i \text{ generated}\end{bmatrix} - \begin{bmatrix}\text{Amount of}\\ x_i \text{ consumed}\end{bmatrix} \quad (22)$$

where, $x_i$ is a conserved extensive quantity, for example mass or energy.

In chemical engineering, usually the last two terms of Eq. 22 (i.e., amount of generated components in a chemical reaction) are especially difficult to model, but the first two terms (i.e., inlet flow or heat transfer) can be obtained more easily. Certainly, it always turns out in the modeling phase which parts of the first-principle model are easier and which parts are more laborious to obtain.

**First-principle linearization control for CSTR:** Here, the control law of GLC is derived on the bases of the first principle model of the controlled process. The state-variables $x = (x_1, x_2, x_3)$ are: $x_1 = CA$ the concentration of A component, $x_2 = T$ the reactor temperature and $x_3 = T_j$ the jacket temperature. The manipulated input u is the coolant flow rate while the controlled output y is the temperature in the reactor. So, the model equations of this affine system can be represented by:

$$f(x) = \begin{bmatrix} \frac{Q}{V}(C_{AF} - x_1) - k_0 \exp(\frac{-E_a}{Rx_2})x_1 \\ \frac{Q}{V}(T_f - x_2) - \frac{H_r}{\rho c_p}k_0 \exp(\frac{-E_a}{Rx_2})x_1 + \frac{UA}{\rho c_p V}(x_3 - x_2) \\ \frac{UA}{\rho c_p V_j}(x_2 - x_3) \end{bmatrix} \quad (23)$$

$$g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{V_j}(T_{jf} - x_3) \end{bmatrix}$$

$$h(x) = x_2$$

Then, the relative order of the system can be determined based on:

$$\begin{aligned} L_g h(x) &= 0 \\ L_g L_f h(x) &= \frac{1}{V_j}(T_{jf} - x_3)\frac{UA}{\rho c_p V} \end{aligned} \quad (24)$$

So, the relative degree of the system is r = 2. Therefore, the feedback control law can be formulated as follows:

$$u = \frac{v - \left(\beta_0 h(x) + \beta_1 L_f h(x) + \beta_2 L_f^2 h(x)\right)}{\beta_2 L_g L_f h(x)} \quad (25)$$

with the following Lie derivatives:

$$\begin{aligned} L_f h(x) &= f_2(x) \\ L_f^2 h(x) &= f_1(x)\frac{\partial f_2}{\partial x_1}(x) + f_2(x)\frac{\partial f_2}{\partial x_2}(x) + f_3(x)\frac{\partial f_2}{\partial x_3}(x) \end{aligned} \quad (26)$$

Where:

$$\begin{aligned} \frac{\partial f_2}{\partial x_1}(x) &= \frac{H_r}{\rho c_p}k_0 \exp\left(\frac{-E_a}{Rx_2}\right) \\ \frac{\partial f_2}{\partial x_2}(x) &= -\frac{Q}{V} + \frac{H_r}{\rho c_p}k_0 \frac{E_a}{Rx_2^2}\exp\left(\frac{-E_a}{Rx_2}\right)x_1 - \frac{UA}{\rho c_p V} \\ \frac{\partial f_2}{\partial x_3}(x) &= \frac{UA}{\rho c_p V} \end{aligned} \quad (27)$$

In the ideal case, Eq. 25 on the basis of input-output feedback linearization results in the following second-order transfer function:

$$\frac{y(s)}{v(s)} = \frac{1}{\beta_2 s^2 + \beta_1 s + \beta_0} \quad (28)$$

But, Eq. 28 cannot be perfectly realized in practice due to some practical difficulties. The first difficulty is that the manipulated input is constrained. However, this is not significant if âi parameters are selected appropriately. The second is that the model parameters are not known accurately; and the third is that the state variables cannot be measured perfectly. These difficulties influence the performance of the GLC controller.

The linear controller can be designed using Eq. 25. Assuming that the desired closed-loop transfer function is a first-order filter represented by:

$$\frac{y(s)}{y_{sp}(s)} = \frac{1}{1 + T_c s} \quad (29)$$

where, y is the output, $y_{sp}$ is the setpoint and $T_c$ is the filter time-constant; then the transfer function of linear controller becomes:

$$\frac{v(s)}{e(s)} = \frac{\beta_2}{T_c}s + \frac{\beta_1}{T_c} + \frac{\beta_0}{T_c}\frac{1}{s} \quad (30)$$

Actually, Eq. 30 is a classical PID controller with the gain $K = \beta_1/T_c$, the integral time constant $T_i = \beta_1/\beta_0$ and derivative time constant $T_d = \beta_2/\beta_0$.

The parameters of feedback linearization (i.e., $\beta_0$, $\beta_1$ and $\beta_2$) were set by trial-and-error method. Adjusting these parameters is easy because the meanings of these parameters are very simple, representing the linear parameters of the desired trajectory. Virtually, tuning of these values is necessary only because the manipulated input is constrained.

So, these parameters must be tuned in such a way that the calculated manipulated input will approximately satisfy these constraints during the control. The selected parameters were set to $\beta_0 = 1$, $\beta_1 = 10$ and $\beta_2 = 5$. The PID

parameters were determined by direct synthesis Eq. 30 as $K_c = 1$, $T_i = 10$ and $T_d = 0.2$. It should be noted here that both the first principle and the semi-mechanistic GLC controller had the same parameters.

**Semi-mechanistic GLC for CSTR:** The first-principle GLC controller builds upon the complete first-principle model of the process. There may be situations when some parts of the first-principle model are not available. In contrast to first-principle GLC, the key strength of semi-mechanistic GLC is that it does not need a complete first-principle model. In the followings, an example will be presented that illustrates this problem through the application example of CSTR.

The control-related model of CSTR consists of two conservation equations:

The heat balance of the reactor and the heat balance of the jacket. The first balance is associated with the controlled variable (i.e., reactor temperature):

$$\begin{bmatrix}\text{Heat accumulated}\\ \text{in reactor}\end{bmatrix} = \begin{bmatrix}\text{Heat of}\\ \text{inlet flow}\end{bmatrix} - \begin{bmatrix}\text{Heat of}\\ \text{outlet flow}\end{bmatrix} - \begin{bmatrix}\text{Heat transferred}\\ \text{to jacket}\end{bmatrix} + \begin{bmatrix}\text{Heat generated}\\ \text{from reaction}\end{bmatrix} \tag{31}$$

while the second balance is associated with the manipulated variable (i.e., flow rate of coolant):

$$\begin{bmatrix}\text{Heat accumulated}\\ \text{in jacket}\end{bmatrix} = \begin{bmatrix}\text{Heat of}\\ \text{coolant inlet}\end{bmatrix} - \begin{bmatrix}\text{Heat of}\\ \text{coolant outlet}\end{bmatrix} + \begin{bmatrix}\text{Heat transferred}\\ \text{from reaction}\end{bmatrix} \tag{32}$$

The mathematical formalization of these terms is well-known in chemical engineering. Certainly, during the formalization of the model terms, some assumptions must be made with respect to certain a priori knowledge, e.g., the heat transfer coefficient was assumed to be constant.

But, it is not enough to provide an exact description of each term, the model parameters must be provided, too. In the course of modeling of chemical reactors, the parameters of chemical reaction rate are generally problematical. In this application example, it is assumed that this part of the model is not available. Hence, the neural network will model the heat released by the chemical reaction. Therefore, the semi-mechanistic model of CSTR can be written as:

$$\begin{aligned}\frac{dx_2}{dt} &= \frac{Q}{V}(T_f - x_2) + f_{NN}(z) + \frac{UA}{\rho c_p V}(x_3 - x_2)\\ \frac{dx_3}{dt} &= \frac{UA}{\rho c_p V_j}(x_2 - x_3) + \frac{u}{V_j}(T_{jf} - x_3)\end{aligned} \tag{33}$$

where, $f_{NN}$ is the neural network, z is the input of this neural network (i.e., temperature of the reactor), $x_2 = T$ is the reactor temperature, $x_3 = T_j$ is the jacket temperature. It can be seen that the semi-mechanistic model Eq. 33 does not contain the x1 state variable. It means that the semi-mechanistic GLC controller does not need to measure the concentration in reactor.

In the followings, the above described semi-mechanistic model will be utilized in the GLC design scheme. The procedure is the same as the first principle model.

For the sake of simplicity, the state variables will be denoted in the same way: $x_2$ is the reactor temperature, $x_3$ is the jacket temperature and thus the state vector is denoted by $x = (x_2, x_3)$. The manipulated input u is the coolant flow rate, while the controlled output y is the temperature in the reactor. Therefore, the model equations become:

$$\begin{aligned}f(x) &= \begin{bmatrix}\frac{Q}{V}(T_f - x_2) - f_{NN}(z) + \frac{UA}{\rho c_p V}(x_3 - x_2)\\ \frac{UA}{\rho c_p V_j}(x_2 - x_3)\end{bmatrix}\\ g(x) &= \begin{bmatrix}0\\ \frac{1}{V_j}(T_{jf} - x_3)\end{bmatrix}\\ h(x) &= x_2\end{aligned} \tag{34}$$

It should be noted that Eq. 32 assumes that the input of the neural network is a function of state variables, i.e., z = f z(x); for example, the input of neural network is composed of measured state-variables.

The relative order of the semi-mechanistic model can be determined through Lie derivatives:

$$\begin{aligned}L_g h(x) &= 0\\ L_g L_f h(x) &= \frac{1}{V_j}(T_{jf} - x_3)\frac{UA}{\rho c_p V}\end{aligned} \tag{35}$$

So, the relative degree of the system is r = 2. The Lie derivatives of the semi-mechanistic model are:

$$\begin{aligned}L_f h(x) &= f_1(x)\\ L_f^2 h(x) &= 0 + f_1(x)\frac{\partial f_1}{\partial x_2}(x) + f_2(x)\frac{\partial f_1}{\partial x_3}(x)\end{aligned} \tag{36}$$

with the following partial derivatives:

$$\begin{aligned}\frac{\partial f_1}{\partial x_2}(x) &= -\frac{Q}{V} + \frac{\partial f_{NN}}{\partial x_2}(z) - \frac{UA}{\rho c_p V}\\ \frac{\partial f_1}{\partial x_3}(x) &= \frac{\partial f_{NN}}{\partial x_3}(z) + \frac{UA}{\rho c_p V}\end{aligned} \tag{37}$$

The relative degree is r = 2, thus the feedback control law is identical to Eq. 25 (but the Lie derivatives differ from each other), that is:

$$u = \frac{v - \left(\beta_0 h(x) + \beta_1 L_f h(x) + \beta_2 L_f^2 h(x)\right)}{\beta_2 L_g L_f h(x)} \quad (38)$$

It is important to consider that for applying the obtained u by feedback linearization method, the stability of the internal dynamics and zero dynamics should be checked. But, by using the semi-mechanistic modeling r is obtained equal to 2 and because the system order also reduces to 2, there is no need for doing stability analysis. Hence, the linear controller of semi-mechanistic GLC controller is the same, i.e., it will be a PID controller.

Now, suppose that some parameters of the system change. It's necessary to apply a method that our controller recognizes these changes for performing the compensatory strategies. From Eq. 34, the Parameters of the plant are Q, V, U, A, $\rho$, $c_p$, $V_p$, $T_i$, $T_f$. It is supposed that V, $V_p$, A, are always constant and they never change but it's possible that the inlet temperature of feed $T_i$, the inlet temperature of coolant $T_f$ and the feed flow rate Q change. It is also possible that the parameters $\rho$ and $c_p$ and U don't remain constant. Clearly the changes of parameters $T_i$, $T_f$ and Q can be measured online by sensors from plant and then they can be incorporated in the controller.

**Adaptation for any change in Q:** In practical applications, assumption of invariability of feed flow rate is totally inaccurate and almost it is impossible to fix the feed flow rate in a desired value.

In the first experiment, without using adaptation, the feed flow rate (Q) decreased by 10% at time = 600 (i.e., 90% decreasing). It can be seen that because of the nature of feedback, the system would tolerate this deviation and remain stable (Fig. 9a, b).

Now, by applying adaptation in the controller and online applying of the change in the feed flow rate to the controller, the results in Fig. 10a and b would be obtained. It can be seen that in this case, the performance of the system is not much better than before.

Now, if the value of flow rate goes 40% more than its nominal value at time = 600, without applying adaptation the results in Fig. 11a and b would be obtained. It is clear that in spite of lack of adaptation, the plant remains stable.

And if adaptation is applied, the results are shown in Fig. 12a and b.

For more changes in Q, because of the physical limitations (e.g., saturation in the flow rate), the system with or without adaptation can not work properly (Fig. 13a, b).

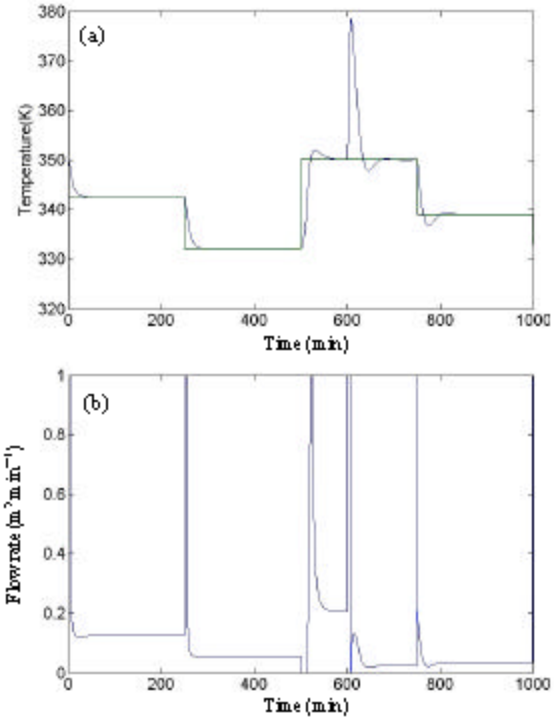


Fig. 9: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in absence of adaptation and 90% decreasing of feed flow rate (Q)

**Adaptation for any change in $T_f$:** If the inlet temperature of the feed changes 1% at time = 600, in the absence of adaptation the results shown in Fig. 14a and b will be obtained.

By applying adaptation in the controller, for 1% increasing in Tf the results shown in Fig. 15a and b are obtained. It is clear that the controller works a little better than the time that it is not adaptive. For more increasing in $T_i$, the online controller may not work properly. This is due to the saturation in manipulated variable.

For 3% decreasing of $T_f$ (without adaptation), the results shown in Fig. 16a and b are obtained.

In case of applying adaptation, the results shown in Fig. 17a and b are accessed.

Based on the results, it can be concluded that there is a negligible difference between the adaptive and non-adaptive cases.

For more decreasing in $T_f$ (with or without applying adaptation), results shown in Fig. 18a and b are obtained. It can be seen that the system is not able to track the set point. This phenomenon is because of lower limitation for manipulated variable. For instance, the input flow rate of jacket can not be less than zero).

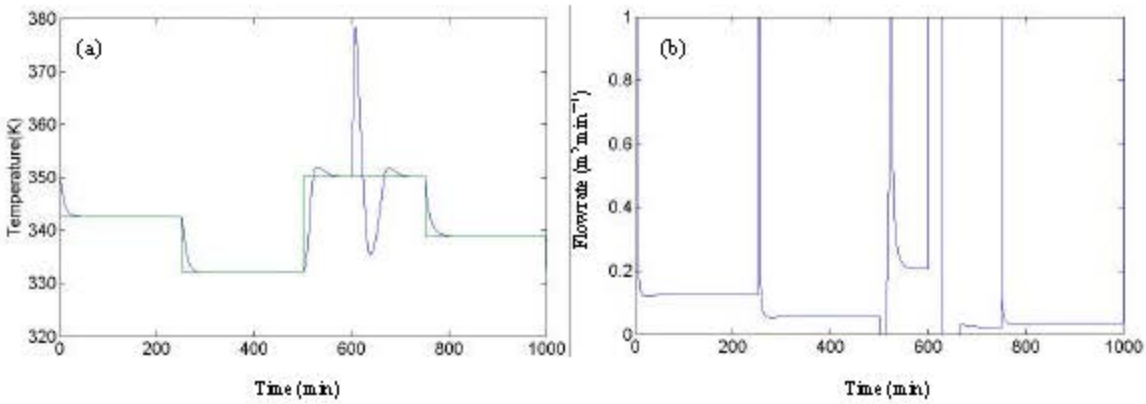


Fig. 10: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 90% decreasing of feed flow rate (Q)

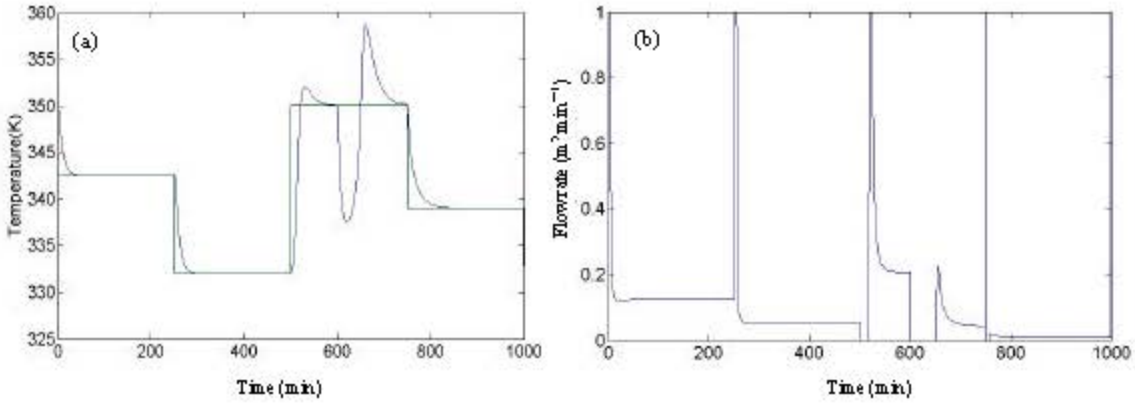


Fig. 11: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in absence of adaptation and 40% increasing of feed flow rate (Q)

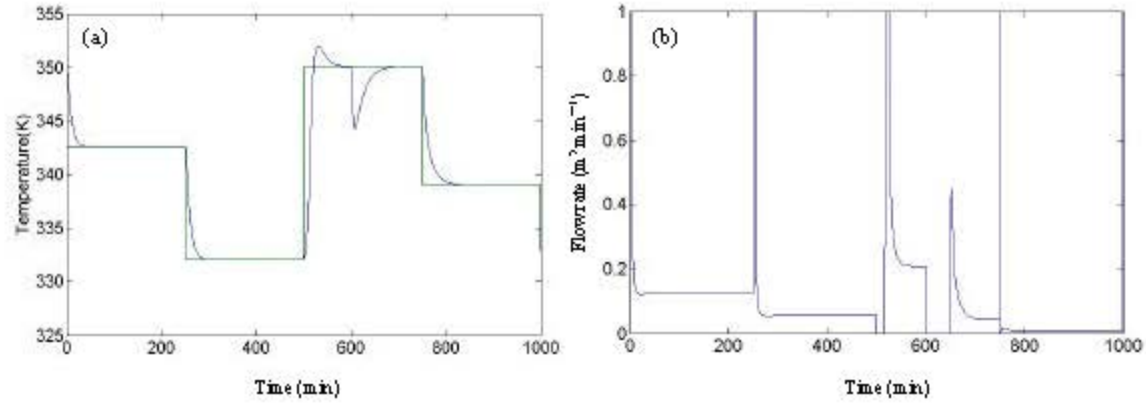


Fig. 12: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 40% increasing of feed flow rate (Q)

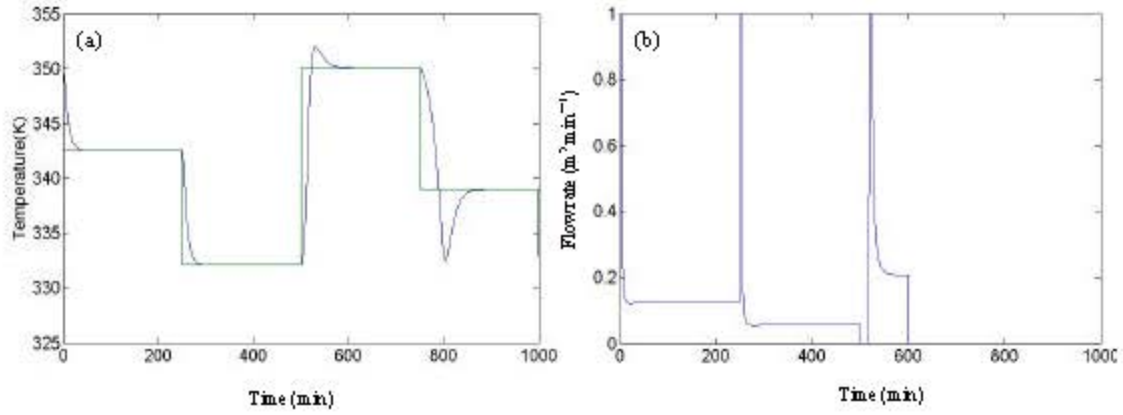


Fig. 13: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 45% increasing of feed flow rate (Q)

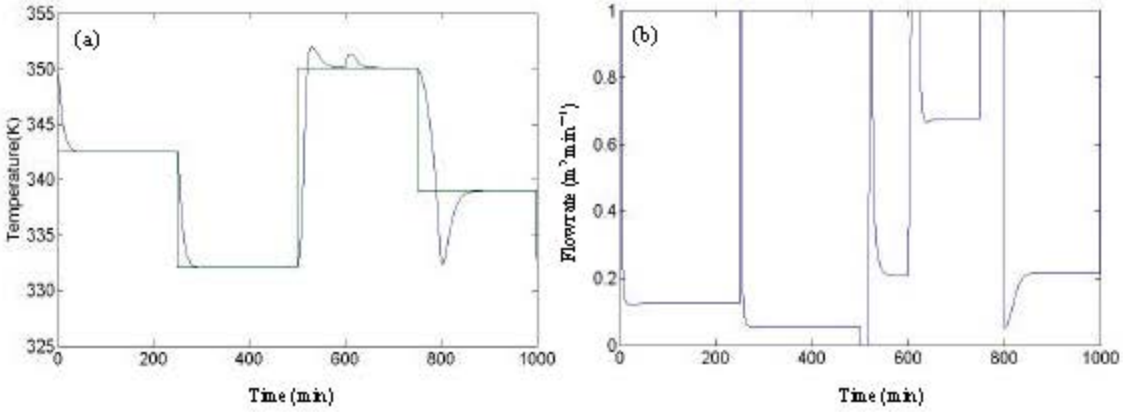


Fig. 14: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in absence of adaptation and 1% increasing of feed temperature

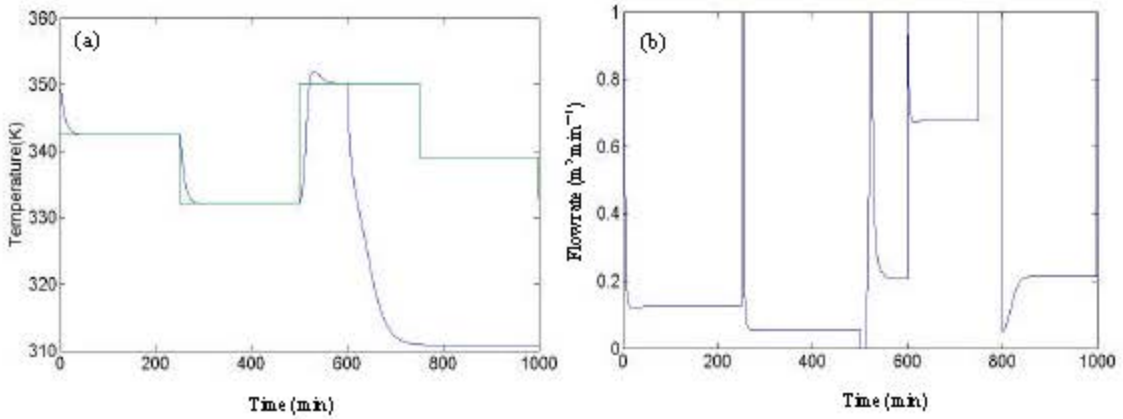


Fig. 15: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 1% increasing of feed temperature

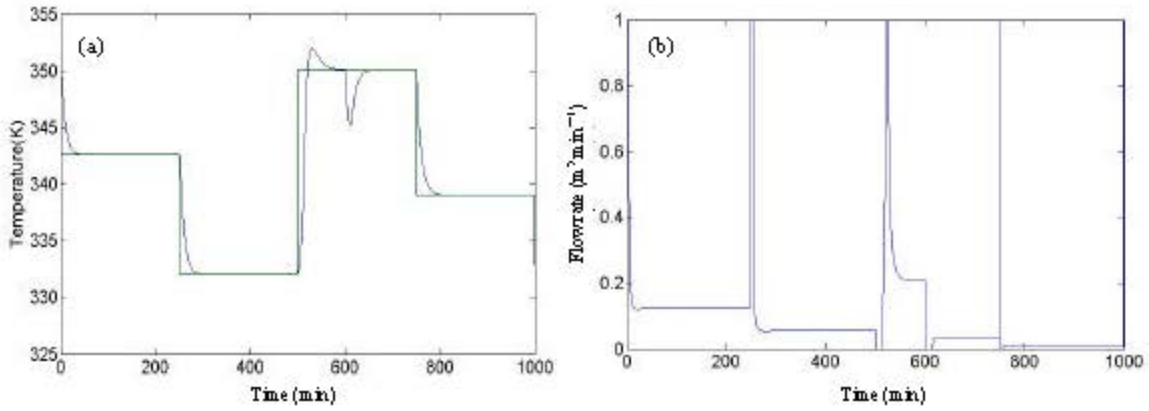


Fig. 16: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in absence of adaptation and 3% decreasing of feed temperature

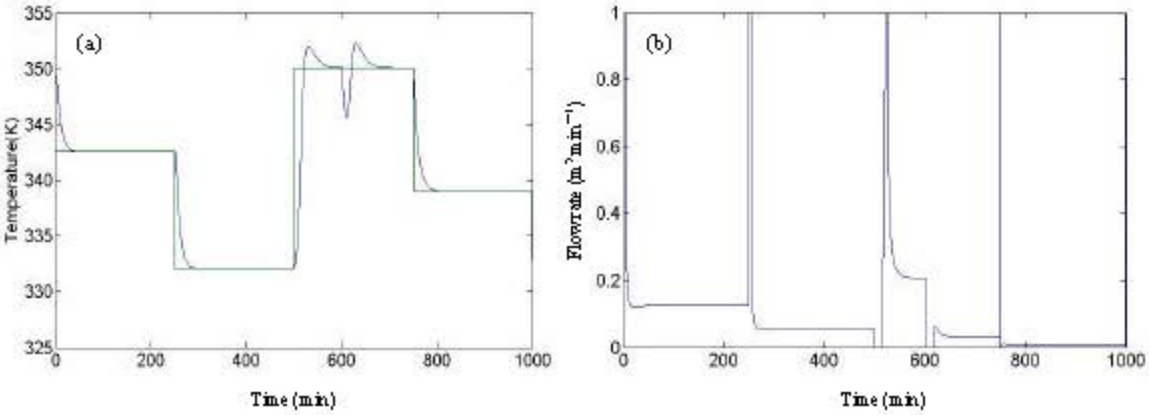


Fig. 17: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 3% decreasing of feed temperature

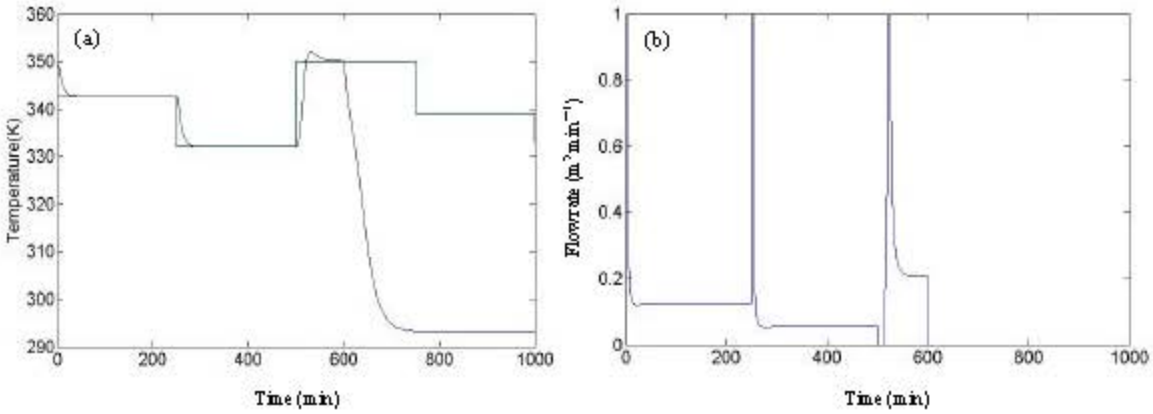


Fig. 18: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 5% decreasing of feed temperature

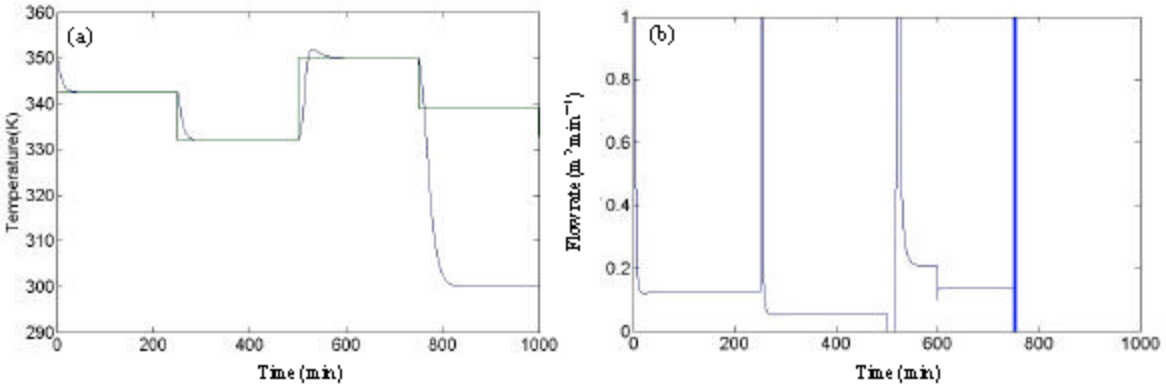


Fig. 19: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi- mechanistic modeling in absence of adaptation and 3% decreasing of inlet jacket temperature

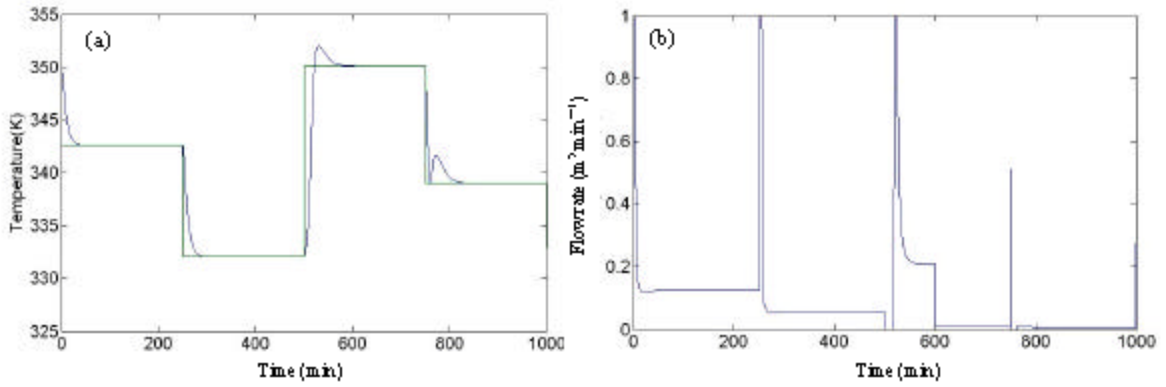


Fig. 20: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi- mechanistic modeling in presence of adaptation and 30% decreasing of inlet jacket temperature

**Adaptation for any change in $T_j$ :** In the absence of adaptation, by decreasing the feed temperature of the jacket more than 3% at time = 600, the system becomes unstable (Fig. 19a, b). By applying adaptation, even when the feed temperature of the jacket decreases 100%, the system remains stable (Fig. 20a, b). It should be noted that by increasing $T_j$ more than 5%, even in presence of adaptation and having no constraint for manipulated variable, the system goes to unstable mode (Fig. 21a, b). This is because of the existence of $T_j$ in denominator of manipulated variable (u). The physical explanation is due to the fact that if the temperature of jacket increases more than a specific amount, the water in the jacket can not absorb the heat generated during the reaction with any possible flow rate.

**Estimation of heat of reaction by neural network:** In Eq. 33, fNN is the model of heat generated during the reaction which can be described by:

$$f_{NN} = \frac{H_r}{\rho c_p} k_0 \exp\left(\frac{-E_a}{Rx_1}\right) x_1 \quad (39)$$

By using NN, generated heat in the reaction can be estimated without any need to online measuring of concentration. By using this approach, any small changes in the parameters of heat equation can be considered and also there is no need to have the exact values of the involved parameters ($H_r$, $k_0$, $E_a$).

It can be seen from Fig. 22 that the neural network would be trained according to the error between the grey-box model output and the actual plant output. Since the other parameters changes would be compensated instantaneously by the controller, it can be expected that the NN reflects the heat of reaction with a good accuracy and its online training would be mostly influenced by changes in heat of reaction parameters and not the other plant parameters.

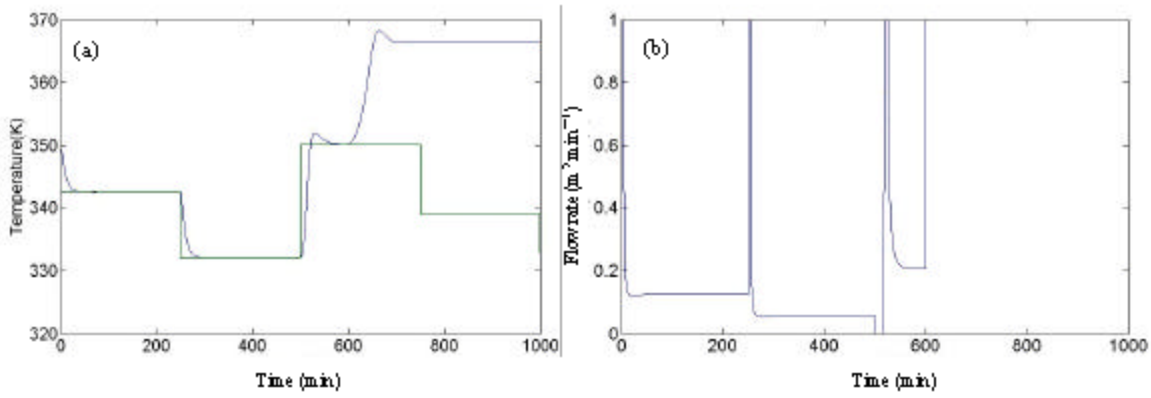


Fig. 21: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi- mechanistic modeling in presence of adaptation and 6% increasing of inlet jacket temperature

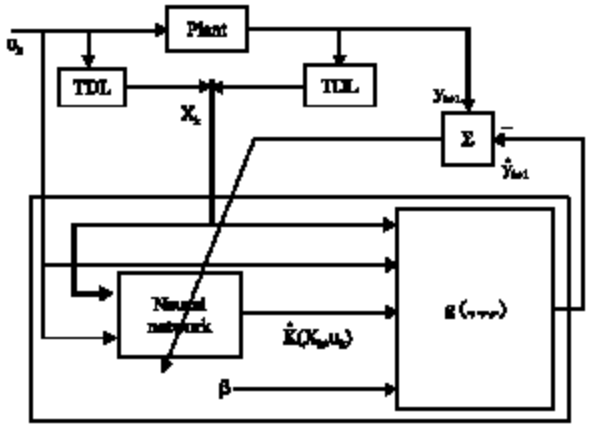


Fig. 22: Training diagram for Neural Network (the box is shown inside the dotted line)

The NN should be able to converge so as to make the following estimation error equal to zero:

$$\begin{aligned} e &= T_{plant} - T_{estimate} \\ T_{estimate} &= T_{plant} - e \\ \dot{T}_{estimate} &= \frac{d(T_{plant} - e)}{dt} \end{aligned} \quad (40)$$

From the structure of the model, the output of NN is:

$$f_{NN} = \frac{d(T_{plant})}{dt} - \frac{d(e)}{dt} - \frac{Q}{V}(T_r - T_{estimate}) - \frac{UA}{\rho c_p V}(Tj - T_{estimate}) \quad (41)$$

Theoretically, the difference between the plant output and the model output should be zero. So, by inserting de/dt = 0 and $T_{plant}$ =T estimate in Eq. 41, the desired output of the neural network should be:

$$f_{NN} = \frac{\Delta T_{plant}}{\Delta t} - \frac{Q}{V}(T_r - T_{plant}) - \frac{UA}{\rho c_p V}(T_j - T_{plant}) \quad (42)$$

For initial training of the neural network in off-line mode, the appropriate input-output dada should be collected.

In this case, the input of the neural network is temperature of the reactor and its output is the reaction heat which can be obtained from Eq. 42.

The results of using NN for reaction heat estimation are shown in Fig. 23a and b. In this case, the controller can detect up to 29% fall in $k_0$ and keep the process in the stable mode.

**Determining the changes of $U/\rho c_p$:** Fouling phenomena can change the value of U (i.e., heat transfer coefficient). In addition, the values of $\rho$ and $c_p$ are somehow dependent on the temperature and also the concentration of the product inside the reactor. So, assumption of changing the term $U/\rho c_p$ is not unrealistic.

In addition, it can be supposed that the deviation of this parameter is not too much and even negligible. But, for considering the generality of the study, a method for online determination of term $U/\rho c_p$ is presented here.

Up to now, it has been assumed that the semi-mechanistic model and some of the system equations of our system are accessible. One of these equations is heat balance equation of the Jacket which is as follows:

$$\frac{dT_j}{dt} = \frac{UA}{\rho c_p V_j}(T - T_j) + \frac{u}{V_j}(T_{jr} - T_j) \quad (43)$$

By defining $UA/\rho c_p = \alpha$ and $t_s$ as the sampling time, it can be written:

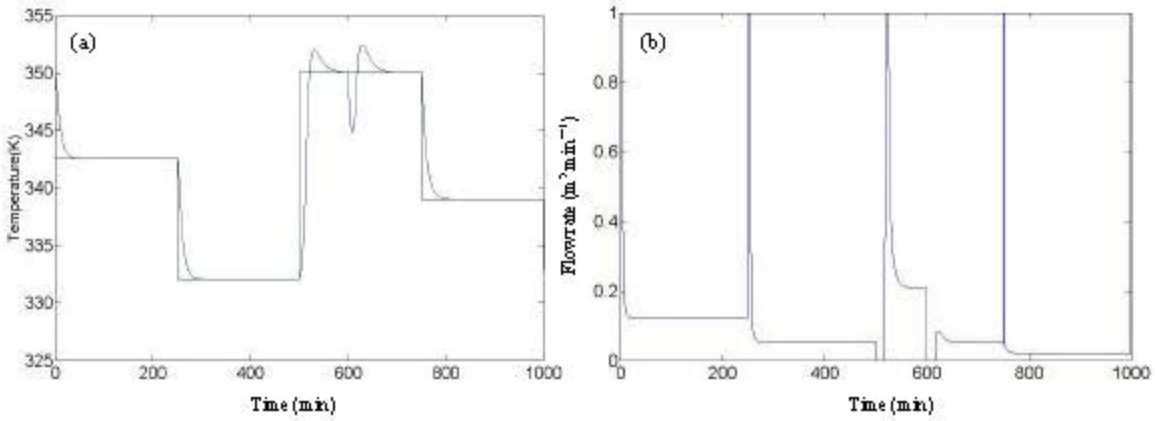


Fig. 23: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of neural network for estimation of heat of reaction when $k_0$ falls to 80% of its initial value

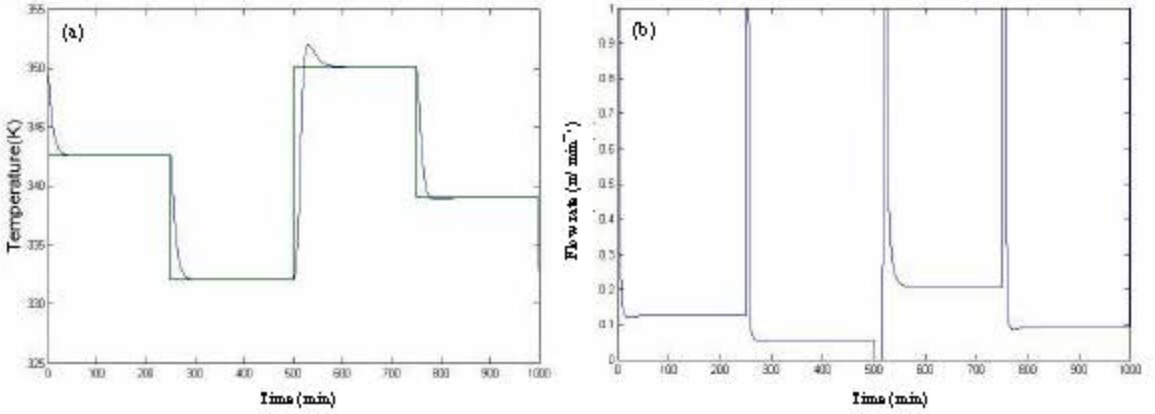


Fig. 24: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi-mechanistic modeling in presence of adaptation and 10% decreasing of in the model

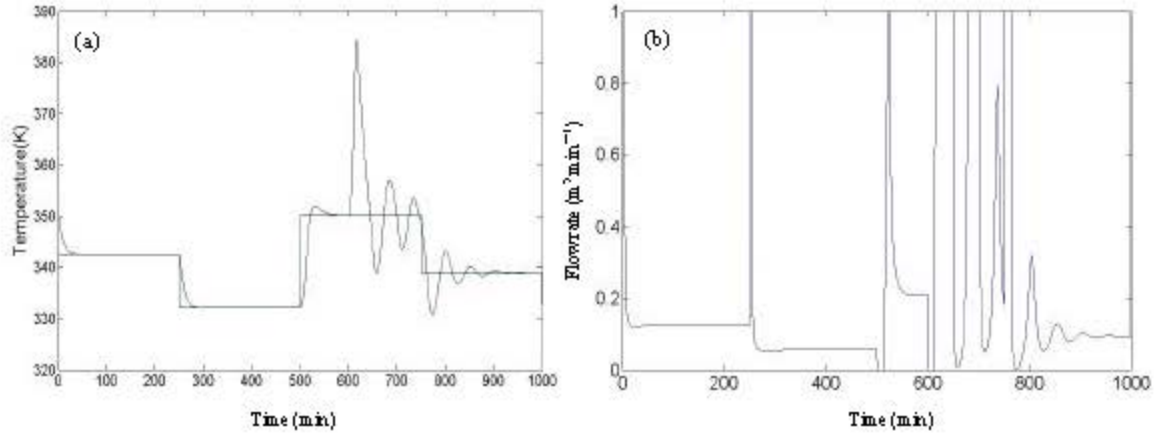


Fig. 25: (a) Temperature and (b) jacket flow rate of CSTR obtained by semi- mechanistic modeling in presence of adaptation and 15% change in $k_0$ and U and also 10% change in $E_a$ 15%

$$T_j^K = t_s \times \frac{\alpha}{V_j}(T^{k-1} - T_j^{k-1}) + t_s \times \frac{u}{V_j}(T_{jf} - T_j^{k-1}) + T_j^{k-1} \quad (44)$$

So, $\alpha$ can be estimated as:

$$\alpha = \frac{T_j^K - t_s \times \frac{u}{V_j}(T_{jf} - T_j^{k-1}) - T_j^{k-1}}{\frac{t_s}{V_j}(T^{k-1} - T_j^{k-1})} \quad (45)$$

Therefore, $\alpha$ can be updated at each sampling time. By this technique up to 19% reduction in $\alpha$ would be tolerable and the stability of the system has been verified by simulation studies. As a typical demonstration, the result for 10% reduction in $\alpha$ has been shown in Fig. 24a and b.

Although, by this method, $UA/\rho c_P$ can be estimated very accurate and fast, since its variation is very slow due to the nature of $UA/\rho c_P$, the observations show that it is not necessary to measure this term in each sample for practical applications.

In Madar *et al*. work (2005), the results have been reported for the uncertainty of 3% in $k_o$, 2.5% in $E_a$ and 10% in U. Whereas, in our proposed method, the controlled system remains stable, as shown in Fig. 25a and b, even for 15% change both in $k_0$ and U and 10% change in $E_a$ all being exercised at t = 600.

## CONCLUSIONS

In this study, two different methods for adaptive control of nonlinear processes using neural networks were presented. The main benefit of these techniques is the online capability in order to incorporate any changes of the plant's parameters in the controller implementation.

The first method is adaptive feedback linearizing control technique which is based on black-box modeling of the plant. The main advantages of this method are:

- Implementation is simple
- It can use a nonlinear model of the system without a priori knowledge

The second method is adaptive feedback linearizing control technique which is based on a semi-mechanistic modeling of the plant. It was practically observed that the power of set-point tracking of this method is more efficient than the first one, giving a better characteristic as well.

It should be mentioned again that semi-mechanistic modeling approach for feedback linearization allows the user to combine black-box modeling with white-box modeling in such a way that a posteriori modeled element replaces the uncertain part of a priori model. In the proposed semi-mechanistic model, a neural network replaces the difficult-to-model part of the priori model.

When precise knowledge about some parts of the uncertain plant exists, it may be used in forming partial structure of the model and the modeling capability of the NN can be focused on the unknown parts. This can be achieved using a serial neuro-gray-box model which is more accurate than the black-box one while it takes shorter training time. Although, serial gray-box schemes can only be used for the plants with some a priori partial knowledge, the resulting validity domain is larger than that of the black-box method.

For the methods in which a black-box model replaces the plant, any changes in parameters can be identified by neural network. But, in the semi-mechanistic feedback linearization beside the use of a neural network for modeling the difficult-to-model part of the plant, extra sensors and transmitters are needed for online estimation of the plant parameters.

So, although the performance of the semi-mechanistic feedback linearization is better than the other methods, its implementation can be more expensive and so a trade-off compromise should be taken to select the proper method for practical applications.

In the semi-mechanistic modeling technique, although the plant parameters' changes of the plant can be distinguished accurately and fast so as to be used in the controller implementation, the plant output may go to an unstable mode mainly due to the manipulated variable saturation.

## REFERENCES

Bazaei, A. and V.J. Majd, 2003. Feedback linearization of discrete-time nonlinear uncertain plants via first-principles-based serial neuro-gray-box models. J. Process Control, 13: 819-830.

Braake, H.A.B., E.J.L. Van-Can, J.M.A. Scherpen and H.B. Verbruggen, 1998. Control of nonlinear chemical processes using neural models and feedback linearization. Comput. Chem. Eng., 22: 1113-1127.

Chen, C.T. and C.S. Dai, 2001. Robust controller design for a class of nonlinear uncertain chemical processes. J. Process Control, 11: 469-482.

Cote, M., B.P.A. Grandjean, P. Lessard and J. Thibault, 1995. Dynamic modeling of the activated sludge process: improving prediction using neural network. Wat. Res., 29: 995-1004.

Fourati, F., M. Chtourou and M. Kamoun, 2008. Stabilization of unknown nonlinear systems using neural networks. Applied Soft Comput., 8: 1121-1130.

Hagan, M.T., H.B. Demuth and M.H. Beale, 1996. Neural Network Design. 1st Edn., PWS Publishing Co., Boston, MA, USA., ISBN: 0-53494332-2.

Henson, M.A. and D.E. Seborg, 1990. Input-output linearization of general nonlinear processes. AIChE J., 36: 1753-1757.

Henson, M.A. and D.E. Seborg, 1994. Adaptive nonlinear control of a pH neutralization process. IEEE Trans. Control Syst. Technol., 2: 169-182.

Henson, M.A. and D.E. Seborg, 1997. Nonlinear Process Control. Prentice Hall, Upper Saddle River, NJ., ISBN-13: 978-0136251798.

Kar, I. and L. Behera, 2009. Direct adaptive neural control for affine nonlinear systems. Applied Soft Comput., 9: 756-764.

Kravaris, C. and C.B. Chung, 1987. Nonlinear state feedback synthesis by global input-output linearization. AIChE J., 33: 592-603.

Lee, P.L. and G.R. Sullivan, 1988. Generic model control (GMC). Comput. Chem. Eng., 12: 573-580.

Lee, D.S., C.O. Jeon, J.M. Park and K.S. Chang, 2002. Hybrid neural network modeling of a full-scale industrial wastewater treatment process. Biotechnol. Bioengin., 78: 670-682.

Madar, J., J. Abonyi and F. Szeifert, 2005. Feedback linearizing control using hybrid neural networks identified by sensitivity approach. Eng. Appli. Artif. Intell., 18: 343-351.

Marino, R. and P. Tomei, 1995. Nonlinear Control Design: Geometric, Adaptive and Robust. Prentice-Hall, London, ISBN-13: 978-0133426359.

McLellan, P.J., T.J. Harris and D.W. Bacon, 1990. Error trajectory descriptions of nonlinear controller designs. Chemical Eng. Sci., 45: 3017-3034.

Narendra, K.S. and S. Mukhopadhyay, 1997. Adaptive control using neural network and approximation model. IEEE Trans. Neural Networks, 8: 475-485.

Nikolaou, M. and V. Hanagandi, 1993. Control of nonlinear dynamical systems modeled by recurrent neural networks. AIChE J., 39: 1890-1894.

Su, H.T., N. Bhat, P.A. Minderman and T.J. McAvoy, 1992. Integrating neural networks with first principles models for dynamic modeling. Proceedings of the IFAC Symposium on Dynamics and Control of Chemical Reactors, 1992, Maryland, USA., pp: 327-332.

Van Can, H.J.L., C. Hellinga, K.C.A.M. Luyben, J.J. Heijnen and H.A.B. Te-Braake, 1996. Strategy for dynamic process modeling based on neural network in macroscopic balances. AIChE J., 42: 3403-3418.

Widrow, B., D.E. Rumelhart and M.A. Lehr, 1994. Neural networks: Application in industry, business and science. Commun. ACM, 37: 93-105.

Zhang, T. and M. Guay, 2005. Adaptive control of uncertain continuously stirred tank reactors actuator with unknown nonlinearities. ISA Trans., 44: 55-68.