# Real-Time Output Feedback Neurolinearization

**Bahreini, Rabeheh; Bozorgmehry Boozarjomehry, Ramin*+**
*Department of Chemical and Petroleum Engineering, Sharif University of Technology, Tehran, I.R. IRAN*

**ABSTRACT:** *An adaptive input-output linearization method for general nonlinear systems is developed without using states of the system. Another key feature of this structure is the fact that, it does not need model of the system. In this scheme, neurolinearizer has few weights, so it is practical in adaptive situations. Online training of neurolinearizer is compared to model predictive recurrent training. Relationships between this controller and neural network based model reference adaptive controller are established. A CSTR reactor and pH control in a neutralization process illustrate performance of this method. Simulation studies show a superior performance with respect to a PI controller.*

**KEY WORDS:** *Feedback linearization, Neural network, pH control, Online training, Model reference adaptive control.*

## INTRODUCTION

The development of the feedback linearization technique offers a powerful tool for nonlinear system control. Control schemes based on feedback linearization provide larger dynamic operation range than the conventional Jacobian linearization ones. Even though making the closed loop system linear in an input/output sense using differential geometric control methods appears to be very efficient, it has not been used in many processes. Most of these methods such as GLC require all process states and exact model of the plant. To relax these necessities, different methods have been developed to use the advantage of neural networks in function approximation. Several methods have been proposed that rely on the affine model of the process [1,2].

To solve the problem of linearization of general nonlinear systems, *Boozarjomehry et al.* [3] have used neural networks. In this method, back propagation learning algorithm is used to train the neurocontroller in an offline manner. In the NN control design based on offline training, a process simulator and large number of data are required. To deal with this requirement, stable adaptive NN design methods based on Lyapunov stability theorem have been developed [4-7]. However, in these methods, obtaining adaptation laws need process model and its states. Online neural network controller, which has been proposed by *Krishnapura et al.* [8], can emulate inverse controller without using model of system. However, this method cannot be used for unstable systems.

In most methods described earlier, model of the system and its states must be known. In this paper, real-time linearization method has been developed without using model of system. The paper is organized as follows: Section 2 describes discrete time globally linearizing control. Section 3 gives details of real time output feedback neurolinearization. Section 4 compares new method with neural network based model reference

* *To whom correspondence should be addressed.*
+ *E-mail: brbozorg@sharif.edu*
*1021-9986/09/2/121 9/$/2.90*

adaptive control. In section 5 two case studies are illustrated, and performance of the new method is compared to a PI and NNBMRAC controller.

## DISCRETE TIME GLC STRUCTURE

In this section, Discrete-time input/output feedback linearization *Soroush et al.* [9] is described. In Feedback linearization control, the original nonlinear model can be transformed into a linear model through proper coordinate transformation. One can place a linear controller with integral action around the linearized system in order to eliminate offset, which leads in the GLC structure. Input-output feedback linearization allows compensating for the nonlinearities of the system using nonlinear state feedback and nonlinear state transformation. The state feedback transformation can be found after selecting stable linear model. The relative order of selected linear model must be equal or greater than process model. For a single-input single-output system, discrete time model can be assumes in a general form, as follows

$$\begin{cases} x(k+1)=\Phi[x(k),u(k)] \\ y(k)=h[x(k)] \end{cases} \tag{1}$$

It is straightforward to show that the exact sampled-data representation of a dead-time free continuous system with finite relative order r always has r=1. Thus, if a discrete time nonlinear system of the form of Eq. (1) has r >1, (r-1)$\Delta$t represents the plant dead-time, whereas the additional delay $\Delta$t is the delay due to sampling [3,9].

$$h^{r-1}\{\Phi[x(k),u(k)]\}=\beta_0 x(k)-\sum_{i=1}^{r}\beta_i y(k+r-i) \tag{2}$$

Where r is relative order and $\beta_i$ are tuning parameters. Linear model equation can be obtained as Eq. (3).

$$y(k+r)+\sum_{l=1}^{r}\beta_l y(k+r-1)=\beta_0 v(k) \tag{3}$$

### *Real-time output feedback neurolinearization (RTOFN)*

To linearize system under the state feedback of Eq. (2), all states must be known and measurable. In the case that, these states cannot be measured, they can be estimated with an observer. Estimation of states with an observer need exact system model, which in the most cases is not available. *Boozarjomehry et al.* [3] proposed a structure that does not require availability of process states and its model. They combined a neural network state estimator and neural network linearizer, and made a concise structure, which does not need state estimation. Fig.1 shows neural networks to approximate state estimator and state feedback linearizer and Fig. 2 shows structure of combined networks. Structure of the Linearizer that has been used in this paper is shown in Fig. 2.

### *Linearized model and external controller structure*

A popular type of closed-loop response is first- order-plus-dead-time one. This can be achieved by setting the parameters of Eq. (3) as follows:

$\beta_l = 0 \qquad l=2,...,r$

Therefore, the closed loop response simplifies into:

$$\frac{\nu(z)}{v(z)}=z^{-r}\frac{1+\beta_1}{1+\beta_1 z^{-1}} \tag{4}$$

Assuming unit gain for linearized system, relation between parameters simplifies as follows:

$$\beta_0 = 1+\beta_1 \tag{5}$$

Using a PI controller as linear controller shown in Eq. (6), results in closed loop model described by Eq. (7)

$$\frac{\nu(z)}{e(z)}=\frac{(1+\lambda)\left(1+\beta_1 z^{-1}\right)}{\beta_0\left(1-z^{-1}\right)} \tag{6}$$

$$\frac{y(z)}{y_{sp}(z)}=\frac{(1+\lambda)z^{-r}}{(1+(\lambda-1)z^{-1}+z^{-r})} \qquad \beta_1=\lambda \tag{7}$$

The following methods have been proposed for online training of the neurolinearizer:

### *Training algorithm 1: model predictive recurrent training (Indirect Training)*

In this method at first, a neural network is trained offline based on input-output data in order to predict the output of plant. At each sample point, using previous neurolinearizer parameters, control action is calculated and this signal along with previous plant outputs are fed to NN process model. Future output of the plant is

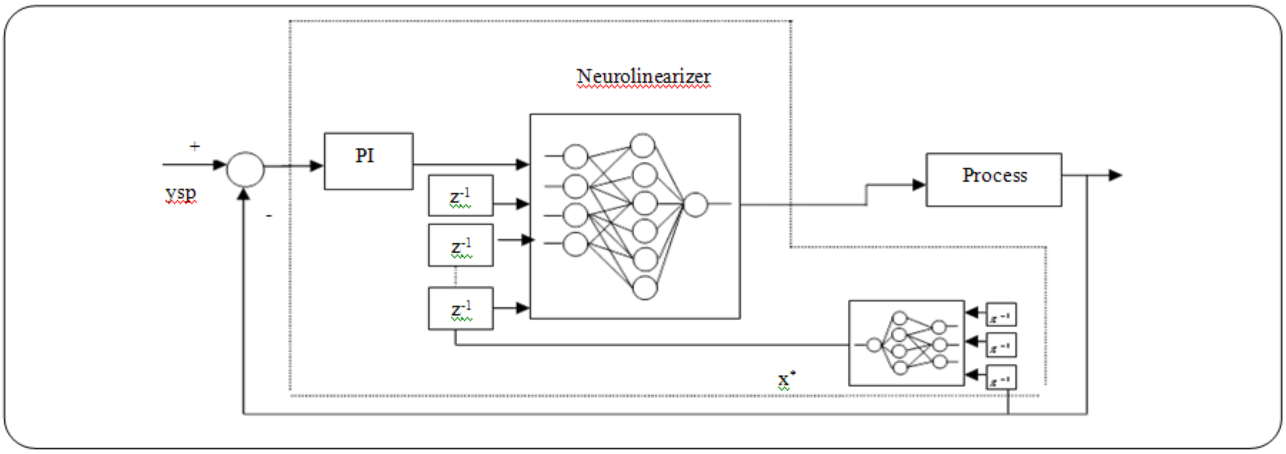


*Fig. 1: Neural networks to approximate state estimation and state feedback linearizer.*

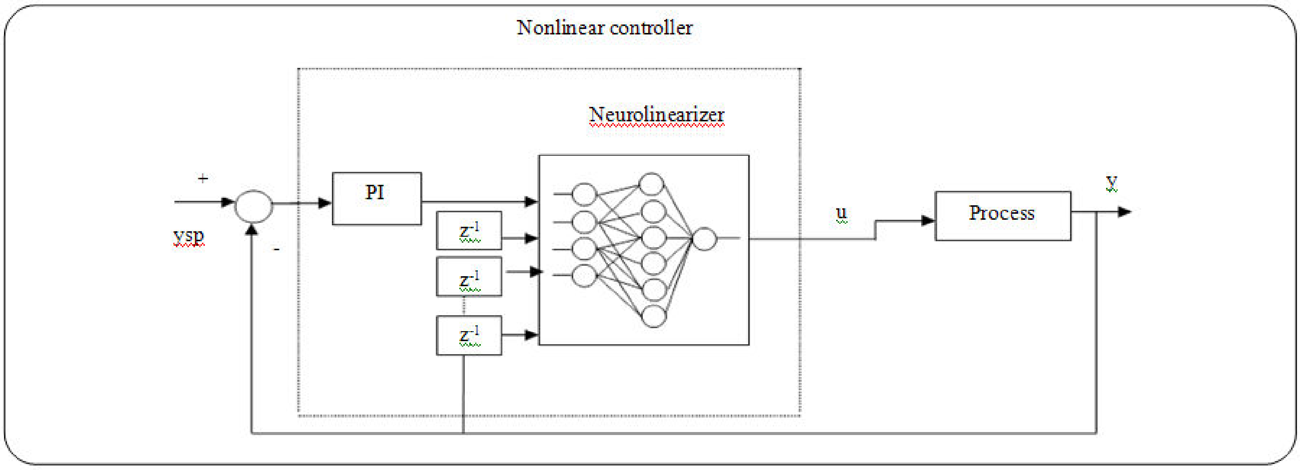


*Fig. 2: Structure of control loop in real-time output feedback neurolinearization (RTOFN).*

predicted and is compared to output of linearized model. The difference between predicted outputs of linearized system and plant output is used as network error and is back propagated through linearizer network. This task is repeated until this error is minimized or maximum epochs are reached. Input structure and number of neurons in hidden layer of neurolinearizer is calculated through a trial and error procedure. Structure of this method is indicated in Fig. 3. This configuration incorporating a process model is called 'indirect' .Although this method has an appropriate performance, it has some drawbacks:

1- To guarantee an efficient control, it requires large number of input-output data to train the neural network process model.

2- An optimization problem is solved at each sample time, which has large computation effort.

3- This method cannot apply to unstable systems.

***Training algorithm 2: Specialized learning (Direct training)***

**Neurolinearizer structure**

In recent years, linear in parameter networks have been found a lot of attention in control [6,10,11]. However, such networks have some drawbacks because of the following reasons:

1- Good performance of such neural networks in function approximation depends on suitable selections of activation functions and their thresholds which is a difficult task [10].

2- RBF neural networks have the property of locality in data representation. This means that each unit is responsible for a relatively small subset of the network input space. Appropriate choice of activation functions would have to account for input space dimensionality. Sufficient number of RBF units necessary to represent a function of (n)

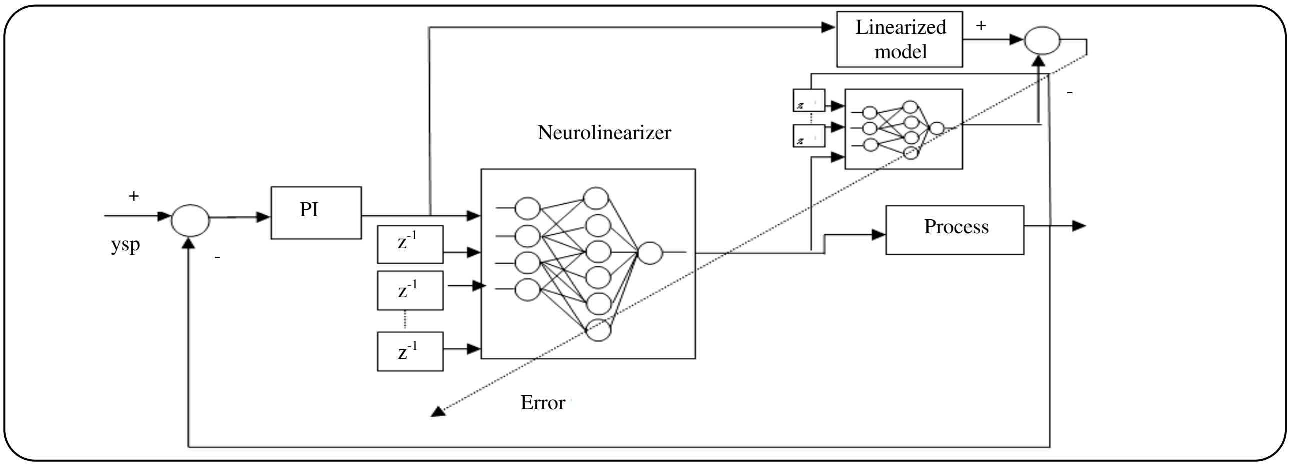


*Fig. 3: Indirect training of neurolinearizer.*

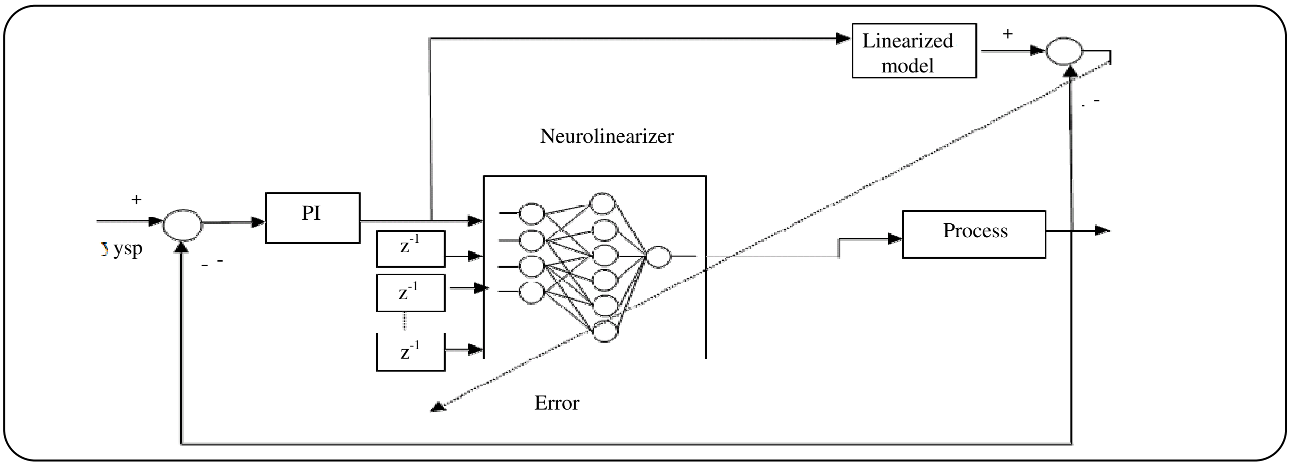


*Fig. 4: Direct training of neurolinearizer.*

arguments, grows exponentially with (n) [12].

3- Significant overlap among the basis functions makes the learning more difficult [13].

Multilayer nonlinear feed forward NN's are theoretically capable of representing arbitrary mappings [14], but these conventional networks have a large number of connection weights. For online learning methods, the learning is computationally demanding because all the weights should be updated in each learning cycle, and this makes the training speed one of the crucial issues of online neuromorfic control algorithms. To fulfill the short training time requirements of online neural networks, network structure should have the minimum of synaptic weights. Hence, neural network that has been used in this paper is a nonlinear feed forward network with one hidden layer, which consists of one neuron. Nonlinear structure of feed forward network with sigmoid activation functions forms a strong compound, which is capable of compensating plant nonlinearities.

***Input structure***

The controller input vector elements are often chosen intuitively without any formal justification. In this paper,the controller network has just two input terms, output of external linear controller and output of process at the previous sampling instant. prevoius outputs of controller is not needed because GLC is a static mapping with respect to control action [15]. For online training, *Psaltis et al* [16] proposed a learning algorithm called specialized learning that allows the neural controller to learn in an adaptive way. They used the difference between the actual output of the plant and the desired output to change the weights, and considered the plant as an additional, unmodifable layer of the neural controller.

The Training steps of the networks used in this paper are as follows:

1- Initial weights of neurolinearizer have to be set to small random numbers.

2- Sum of square error of some of previous sampling times are back propagated through neurolineaizer network. This error is the difference between process output and the output of linear model. This short error function length provides sufficient information to adapt neurolinearizer parameters in the presence of time delay or processes with relative order greater than one.

3- The inputs to the network should be normalized into a small region, depending on their magnitudes.

4- Process gain is substituted with its sign using prior knowledge of the plant.

5- The learning rate has to be chosen carefully to prevent sluggish or unstable response.

6- Parameters of Linear model in Eq. (3) should be selected such that this model represents the linearization of nonlinear system in high gain operating ranges.

7- An important feature of this method is that only one epoch is required for the weights updating algorithm.

***Neural network Based model reference adaptive control (NNBMRAC)***

The model reference adaptive control scheme consists of four blocks: process, controller, reference model, and adaptor. The objective of a model reference adaptive control (MRAC) is to obtain a control law and an updating law, such that the closed loop response tracks a reference model. For nonlinear systems, controller design procedure is based on input-output linearization [17]. An alternative approach for nonlinear MRAC is to adapt the weights of a neural network controller to achieve input-output linearization [18, 19]. Although real-time output feedback neurolinearization (RTOFN) seems to be similar to neural network model reference adaptive control (NNBMRAC), there are several dissimilarities between these two methods, which are as follows:

1- In NNBMRAC controller, neural network is the only controller. In RTOFN, neural network is used to compensate process nonlinearity.

2- For highly nonlinear systems, whose gain changes several orders of magnitude, performance of RTOFN is superior with respect to NNBMRAC controller because of external integrator, which is used in RTOFN structure.

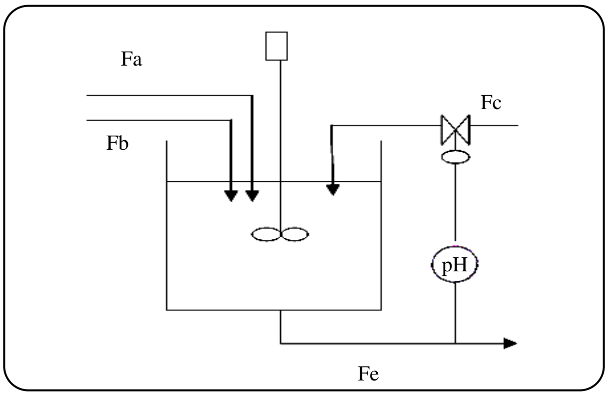


***Fig. 5: pH control system.***

***Simulation studies***

Two standard benchmarks frequently used by control research community are selected to demonstrate the performance of the proposed methods. These systems are a pH neutralization system and a CSTR reactor. The pH neutralization process is a highly non-linear process in which the process gain varies over several orders of magnitude depending on the buffer flow rate and operating condition. The model of the pH neutralization process which is used in this study follows that proposed by *Henson et al.* [20]. The other system is a CSTR reactor, with steady state multiplicities [9]. The control objective in this benchmark is temperature control of the reactor. The performance of RTOFN, NNBMRAC and PI controllers are compared against each other.

***pH control in a neutralization process***

pH neutralization is quite common in the chemical process industry. Due to its highly nonlinear behavior, pH neutralization processes have been used as the benchmarks in nonlinear process control studies [20-23]. As shown in Fig. 5, this system has three input stream, including acid($HNO_3$), buffer ($NaHCO_3$) and base stream (NaOH), respectively. The operating conditions and parameters of the process model are shown in table 1. PID controller parameters are those used by *Nahas et al.* [24]. Structure of neurolinearizer for this system is shown in table 2. Set point tracking of this controller is shown in Fig. 6. Regulatory performance of two controllers is shown in Fig. 7. Since this process has a delay of two-sample points [24], and its relative order is one, the length of error function which has the best performance is 3.

*Table 1: Parameters of neutralization process.*

| | | | |
|---|---|---|---|
| $(C_{HNo3})_a$=0.003 M | $(C_{HNo3})_b$=0 M | $W_{1a}$ = 0.003 M | $pK_1$ = 6.35 |
| $(C_{NaHNo3})_a$=0 M | $(C_{NaHNo3})_b$=0.03 M | $W_{2a}$ = 0 M | $pK_2$ = 10.25 |
| $(C_{NaOH})_a$=0 M | $(C_{NaOH})_b$=0.03 M | $W_{1b}$ = 0 M | pHe=7.00 |
| $(C_{HNo3})_c$=0 M | $F_a$=16.6 mL/s | $W_{2b}$ = 0.03 M | V = 2900 mL |
| $(C_{NaHNo3})_c$=0.0005 M | $F_b$=0.55 mL/s | $W_{1c}$ = -0.00305 M | |
| $(C_{NaOH})_c$=0.003 M | $F_c$=15.5 mL/s | $W_{2c}$ = 5’ $10^{-5}$ M | |

*Table 2: Structure of networks used in each method for pH system.*

| Neural network | Structure (input×hidden×output) | Inputs | outputs |
|---|---|---|---|
| Process model | 4 ×10×1 | pH(k-1), pH(k-2), pH(k-3), $Q_b$(k) | pH(k+1) |
| Neurolinearizer | 2×1×1 | pH(k), v(k-2) | $Q_b$(k) |

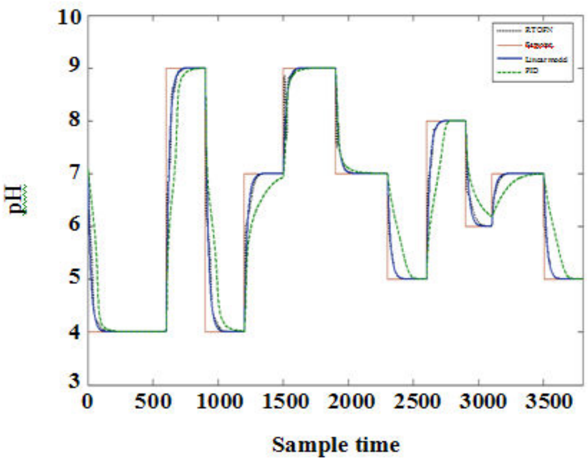


***Fig. 6: Set point-tracking capability of controller for pH system.***

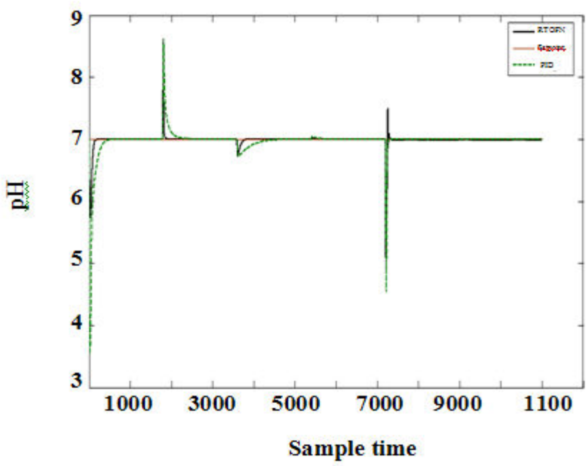


***Fig. 7: Closed loop response of pH system to disturbance given in table 3.***

Since variation of process gain near neutralization point (6<pH<8) is considerable, Peaking phenomena, which is the characteristic of high gain systems, is observed in NNBMRAC, while it is not observed in RTOFN. Fig. 8 represents this comparison.

$$V\frac{dW_e}{dt}=F_aW_a+F_bW_b+F_cW_c-(F_a+F_b+F_c)W_e \qquad (8)$$

$$W=\begin{bmatrix}W_1\\W_2\end{bmatrix}=\left[C_{H^-}C_{OH}-C_{HCO_3^-}-2C_{CO_3^{2-}}\right]$$

$$W_{1e}+10^{PH-14}+W_{2e}\frac{1+2\times10^{PH-PK_2}}{1+10^{PK_2-PH}+10^{PH-PK_2}}-10^{-PH}=0$$

***Temperature control of a CSTR reactor***

This method is applied to a CSTR reactor shown in Fig. 9 in which the following reactions take place.

$$\begin{cases}A\xrightarrow{k_1}U_1\\A\xrightarrow{k_2}U_2\\A\xrightarrow{k_3}D\end{cases} \qquad (9)$$

Where $U_1$ and $U_2$ are undesired side products, and D is desired product. The feed contain only A component. Temperature control is obtained by manipulating the heat input. This process is an unstable system with steady state multiplicities so, model predictive recurrent training

***Table 3: Disturbance patterns on acid and buffer flow rates.***

| Sample time | Buffer flow rate(ml/s) | Acid flow rate(ml/s) |
|---|---|---|
| 0 | 0.55 | 16.6 |
| 1800 | 1.2 | 14.6 |
| 3600 | 2.0 | 18.6 |
| 5400 | 1 | 16.6 |
| 7200 | 0.05 | 16.6 |
| 9000 | 0.55 | 16.6 |

***Table 4: Parameters of the CSTR reactor.***

| | |
|---|---|
| $K_{10} = 2\times10^3$ $m^6.Kmol^{-2}.s^{-1}$ | $E_1 = 4.90\times10^4$ $Kj.Kmol^{-1}$ |
| $K_{20} = 3.4\times10^6$ $Kmol^{0.5}.m^{-1.5}.s^{-1}$ | $E_2 = 6.50\times10^4$ $Kj.Kmol^{-1}$ |
| $K_{30} = 2.63\times10^5$ $s^{-1}$ | $E_3 = 5.70\times10^4$ $Kj.Kmol^{-1}$ |
| $-\Delta H_1 = 4.50\times10^4$ $Kj.Kmol^{-1}$ | $n_1 = 3.00$ |
| $-\Delta H_2 = 5\times10^4$ $Kj.Kmol^{-1}$ | $n_2 = 0.5$ |
| $-\Delta H_3 = 6\times10^4$ $Kj^6.Kmol^{-1}$ | $n_3 = 1.00$ |
| R= 1000 $Kg.mol^{-3}$ | $c = 4.2$ $Kj.Kg^{-1}.K^{-1}$ |

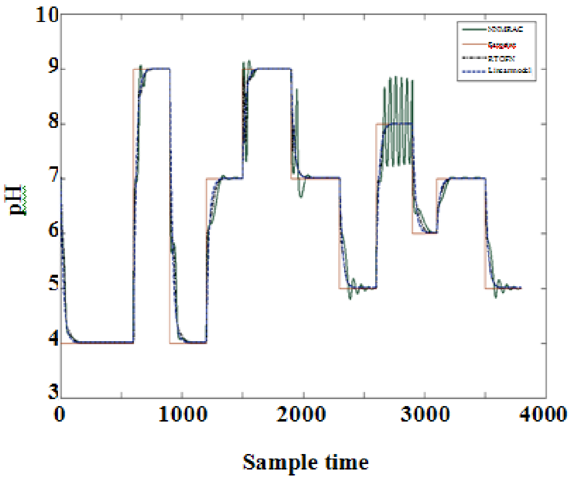


***Fig. 8: Performance of RTOFN with respect to NNMRAC.***

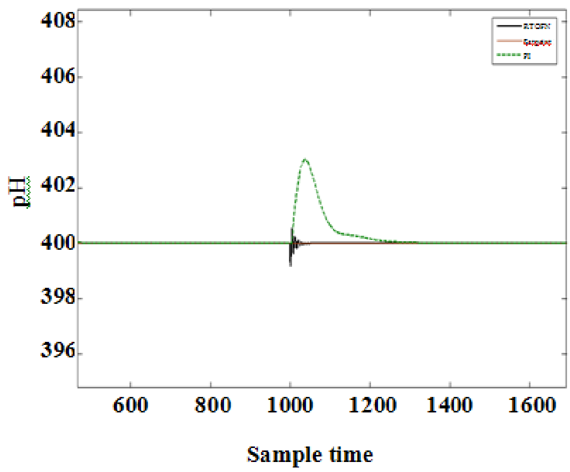


***Fig. 10: Disturbance rejection of the CSTR reactor.***

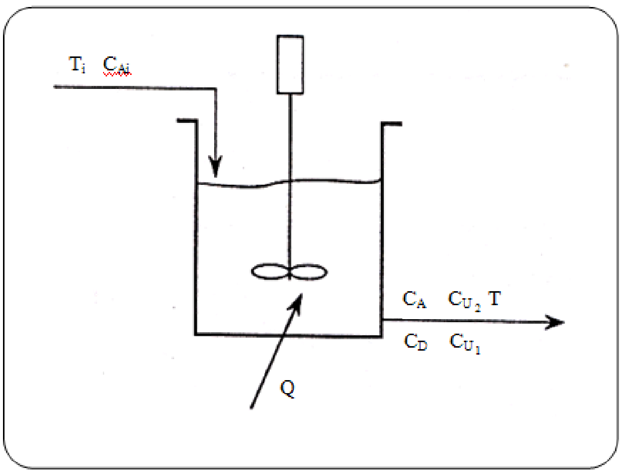


***Fig. 9: CSTR reactor.***

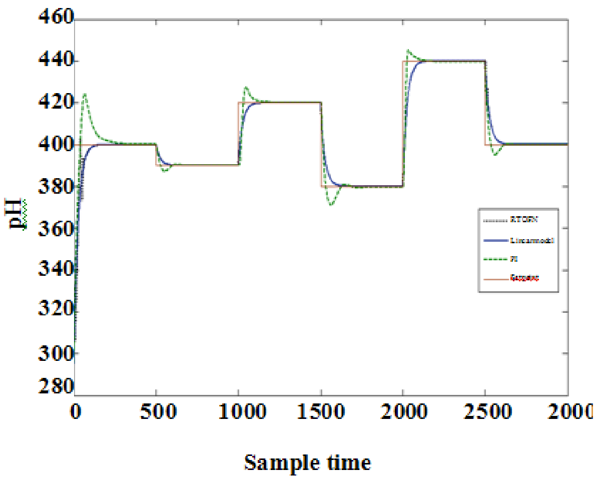


***Fig. 11: Closed loop response of the CSTR system to setpoint changes.***

*Table 5: Operating conditions of CSTR reactor.*

| | | |
|---|---|---|
| $C_{Ai}$ = 10 Kmol/m$^3$ | $C_{Dss}$ = 4 Kmol/m$^3$ | T(0) = 295.2 K |
| $C_A(0)$ = 0.1 Kmol/m$^3$ | V=0.01 m$^3$ | $T_{ss}$ = 400 K |
| $C_D(0)$ = 0 Kmol/m$^3$ | t = 300 s | $T_i$ = 295.2 K |
| $C_{Ass}$ = 1.3204 Kmol/m$^3$ | $Q_{ss}$ = -1.0303 kj / s | |

*Table 6: Structure of neurolinearizer for CSTR system.*

| Neural network | Structure (input×hidden×output) | Inputs | outputs |
|---|---|---|---|
| Neurolinearizer | 2×1×1 | T(k-1),v(k) | Q(k) |

cannot be used with this system. Parameters of this system is shown in tables 4 and 5. Structre of neurolinearizer is summmerized for this system is summerized in table 6. Performance of RTOFN controller for this process is not differing from NNMRAC. The parameters of PI controller ($k_c$=1.446, $\tau_i$=80) are those used by *Boozarjomehry et al.* [3].

Fig. 10 presents the regulatory performance of controller when an unmeasured step change occurs in feed temperature 1000 s after the plant startup. Set point tracking of this controller is shown in Fig. 11. rror function length which is selected for this system is one because there is no delay and its relative order is one. RTOFN outperform PI controller in both set point tracking and disturbance rejection.

$$\begin{cases} \dfrac{dC_A}{dt} = K_1 C_A^{n_1} - K_2 C_A^{n_2} - K_3 C_A^{n_3} + \dfrac{C_{A_2} - C_A}{\tau} \\ \dfrac{dT}{dt} = \\ \dfrac{(-\Delta H_1) K_1 C_A^{n_1} + (-\Delta H_2) K_2 C_A^{n_2} + (-\Delta H_3) K_3 C_A^{n_3}}{\rho C} \\ + \dfrac{T_i - T}{\tau} + \dfrac{Q}{\rho C V} \\ K_i = K_{i0} \exp\left(\dfrac{-E_i}{RT}\right) \quad i = 1,2,3 \end{cases} \tag{10}$$

## CONCLUSIONS

On-line neuromorphic input-output linearization of general nonlinear systems has been proposed. In this method, there is no requirement to know process model and its states. All the othe alternative methods used for neuromorphic I/O linearization have an offline manner, which requires a large network and tarining data set for its training. In the proposed method this drawback has been resolved, and there is no need to have a large data set obtained from excitation of the system. In real-time output feedback neurolinearization, the number of network synaptic weights is severely lower than those of the network used on Offline neurolinearization and this results in the ease of training.

The performance of the proposed method has been compared to neural network model reference adaptive control and an optimal PI controller. RTOFN outperforms both NNMRAC and PI controller in both set point tracking and disturbance rejection in the case of higly nonlinear system pH. performance of RTOFN is similar to NNMRAC for The CSTR system.

**Nomenclatures**

| | |
|---|---|
| c | Heat capacity of the mixture |
| $C_A$ | Concentration of reactant A in the ractor |
| $C_{Ass}$ | Concentration of reactant A in the reactor at steady state |
| $C_{A_i}$ | Concentration of reactant A in the reactor feed |
| $C_D$ | Concentration of product (D) in the reactor |
| $C_{D_{ss}}$ | Concentration of desired product at steady state |
| $E_i$ | Activation energy of reaction i |
| $F_a$ | Acid flow rate |
| $F_b$ | Buffer flow reate |
| $F_c$ | Base flow rate |
| $K_{0i}$ | Frequency factor of the reaction i |
| $n_i$ | Order of the reaction i |
| $pK_i$ | Logarithm of equilibrium dissocition constant for reaction i |
| Q | Heat input to the reactor |
| $Q_{ss}$ | Steady-state heat input to the reactor |
| r | Relative order of the process output with respect to its input |
| $Q_b(k)$ | Base flow rate at k sample point |
| R | Universal gas constant |
| T | Reactor temprature |
| t | Time |
| $T_i$ | Feed temprature |
| $T_{ss}$ | Steady state temprature of the reactor |
| u | Manipulating varaible |

| | |
|---|---|
| $U_1,U_2$ | Undersired products |
| v | External controller output |
| V | Reactor valume |
| x | Vector of process states |
| $x^*$ | Vector of estimated process states |
| y | Process output |
| $y_{sp}$ | Output set point |
| $\beta_i$ | Controller tuning parameters |
| r | Mixture density |
| $DH_i$ | Tuning parameter heat of reaction of reactant i |
| pH (k) | pH at k sample point |

*Received : 25th November 2007 ; Accepted : 22nd July 2008*

## REFERENCES

[1] Peel, C., Willis, M. J., Tham, M. T. and Manchanda, S., Globally Linearizing Control Using Artificial Neural Networks, Proc. of IEEE Inter. Conf. on Cont., 967 (1994).

[2] Braake, A.A. B., Can, E. J. L.V., Scherpen, J. M. A., Verbruggen, H.B., Control of Nonlinear Chemical Processes Using Neural Models and Feedback Linearization, *Computers Chem. Eng.*, **22**, 1113 (1998).

[3] Boozarjomehry, R.B., Svercek, W.Y., Output Feedback Neurolinearizations *ISA Transactions*, **40**, 139 (2001).

[4] Polycarpou, M.M., Stable Adaptive Neural Control Schemes for Nonlinear Systems and Neural Networks, *IEEE Trans. Automat. Control*, 447 (1996).

[5] Ge, S.S., Hang, C. C., Zhang, T, Nonlinear Adaptive Control Using Neural Networks and its Application to CSTR Systems, *Journal of Process Control*, **29**, 313 (1998).

[6] Ge, S.S., Hang, C. C., Zhang, T., Adaptive Neural Network Control of Nonlinear Systems by State and Output Feedback, *IEEE Transactions on Systems, Man, and Cybernetics part B,* 818 (1999).

[7] Hovakimyan, N., Yang, B. J., Calise, A. J., Adaptive Output Feedback Control Methodology Applicable to Non-Minimum Phase Nonlinear Systems, *Automatica,* **42**, 513 (2006).

[8] Krishnapura, V.G. Jutan, A., A Neural Adaptive Controller, *Chemical Engineering Science*, **55**, 3803(2000).

[9] Soroush, M., Kravaris, C., Discrete-Time Nonlinear Controller Synthesis by Input/Output Linearization, *AICHE*, **38**, 1923(1992).

[10] Jagannathan, S., Discrete Time CMAC NN Control of Feedback Linearizable Nonlinear Systems Under Persistence of Excitation, *IEEE Transactions on Neural Networks*, **10**, 128 (1999).

[11] Venkateswalu, Ch., Rao, K.V., Dynamic Recurrent Radial Basis Function Network Model Predictive Control of Unstable Nonlinear Processes, *Chemical Engineering Science*, **60**, 6718 (2005).

[12] Hrycej, T., "Neuro Control Toward an Industrial Control Methodology," John Wiley & sons, *Inc*. (1997).

[13] Ahmed, M. S., Neural Net Based MRAC for a Class of Nonlinear Plants, *Neural Networks*, 112 (2000).

[14] Cybenko, G., Approximation by Superposition of a Sigmoid Function, *Mathematics of Control, Signals and Systems*, **2(4)**, 303(1989).

[15] Henson, M.A., Feedback Linearization Strategies for Nonlinear Process Control, PhD. Dissertation, University of California, Santa Barbara (1992).

[16] Psaltis, D., Sideris, A., Andmamura, A Neural Cntroller, Proc. IEEE 1st Int. Conf. on Neural Networks, 217 (1987).

[17] McLain R.B., Henson M.A., Principle Component Analysis for Nonlinear Model Reference Adaptive Control, *Computers and Chemical Engineering*, 99 (2000).

[18] Cheng, J., Yi, J., Zhao, D., Neural Network Based Model Reference Adaptive Control for Ship Steering System, *International Journal of Information Technology,* **11(6)***,* 75(2005).

[19] Douratsos, I., Gomm, J.B., Neural Network Based Model Reference Adaptive Control for Processed with Time Delay, *International Journal of Information and Systems Sciences,* **3(1)**, 161(2007).

[20] Yeo, Y., K., Kwon, T. I., A Neural PID Controller for the pH Neutralization Process, *Process Design and Control*, 978(1999).

[21] Chen, J., Huang, T. C., Applying Neural Networks to On-Line Updated PID Controllers for Nonlinear Process Control, *Journal of Process Control,* 211(2004).

[22] Boozarjomehry, R.B., Svercek, W.Y., Aoutomatic Design of Neural Network Structures, *Computers and Chemical Engineering*, **25**, 1075(2001).

[23] Boling, J.M., Seborg, D.E., Hespanha, J. P., Multi-Model Adaptive Control of a Simulated pH Neutralization Process, *Control Engineering Practice,* **15**, 663 (2007).

[24] Nahas, E. P., Henson, M.A., Seborg, D.E., Nonlinear Internal Model Control Strategy for Neural Network Models, *Comput. Chem. Eng.*, **16**, 1039(1992).