

LOCAL LINEAR MODEL TREE AND NEURO-FUZZY SYSTEM FOR MODELLING AND CONTROL OF AN EXPERIMENTAL pH NEUTRALIZATION PROCESS

G. Petchinathan^{1*}, K. Valarmathi², D. Devaraj¹ and T. K. Radhakrishnan³

¹Department of Instrumentation and Control Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India.
Pin Code - 626 126, Phone: + 91 90035 23312, Fax: + 91 4563 289322.
E-mail: gpetchi@gmail.com

²P. S. R. Engineering College, Sivakasi, Tamilnadu, India.
E-mail: krvalarmathi@yahoo.co.in

³Department of chemical Engineering, National Institute of Technology, Trichy, Tamilnadu India.
E-mail: radha@nitt.edu

(Submitted: September 19, 2012 ; Revised: April 4, 2013 ; Accepted: July 23, 2013)

Abstract - This paper describes the modelling and control of a pH neutralization process using a Local Linear Model Tree (LOLIMOT) and an adaptive neuro-fuzzy inference system (ANFIS). The Direct and Inverse model building using LOLIMOT and ANFIS structures is described and compared. The direct and inverse models of the pH system are identified based on experimental data for the LOLIMOT and ANFIS structures. The identified models are implemented in the experimental pH system with IMC structure using a GUI developed in the MATLAB -SIMULINK platform. The main aim is to illustrate the online modelling and control of the experimental setup. The results of real-time control of an experimental pH process using the Internal Model Control (IMC) strategy are also presented.

Keywords: LOLIMOT; ANFIS; Internal Model Control; pH process.

INTRODUCTION

Control of pH plays a very important role in chemical industries, such as wastewater treatment, polymerization reactions, fatty acid production, biochemical processes, etc. Modelling and control of a pH process is a very challenging and extremely complex task, because the slope of the process nonlinearity can be very steep around the neutralization region and small changes in the influent stream concentration can result in tremendous changes in pH (Rodrigues and Loparo, 2004). It is very difficult to achieve high performance and robust control using

conventional PID control schemes. Conventional PID controllers must be tuned with predictive capability in order to ensure closed-loop stability for the full range of operating conditions. This means that the plant will not operate at high efficiency. To improve the efficiency of the control scheme, some advanced control techniques like model-based control have been proposed. Some of the model-free control techniques like reinforcement learning (Syafiea *et al.*, 2007) and universal learning network (Han *et al.*, 2006) also have been proposed. Once the accurate modeling of a nonlinear process is completed (Billings, 1980), the internal model control

*To whom correspondence should be addressed

(Nahas *et al.*, 1992) and model predictive control (Richalet, 1993) schemes can be employed to improve the closed loop performance of the system. Due to the complexity of the industrial process, a first-principles model cannot be applied and hence a more flexible type of model needs to be used. In recent decades, numerous schemes have been used for modelling and control of nonlinear processes. The genetic algorithm (GA) approach is used to find the optimal values of the model parameters and controller parameters for modelling and control of complex nonlinear process (Mwembeshi and Kent, 2004; Valarmathi *et al.*, 2009). The response time is high in the case of the GA based approach. The multi-layer artificial neural network (ANN) was introduced in the 1980s. It allows fast processing of large amounts of information. A large repository of literature using ANNs and their application in many different fields like modelling and control are available. The ANNs are also used in system identification and control of pH processes (Sean, 1999; Wior *et al.*, 2010; Sivaraman and Arulsevi, 2011). Radhakrishnan *et al.* (2009) proposed a GA and ANN approach for system identification and control of a pH process. The main problem with the neural based approach is that no human expertise can be stored. The fuzzy logic-based modelling (Babuska and Verbruggen 2003) approach has been widely used in many industrial processes. The main difficulty related to fuzzy-based approach is that it needs prior knowledge about the operation of the process.

To overcome the problems associated with individually using fuzzy and neural approaches, a large variety of Neuro-Fuzzy system (NFS) approaches was introduced in the 1990s. Jang (1993) introduced the adaptive neuro-fuzzy inference system (ANFIS), which belongs to a class of NFS. ANFIS is an adaptive network which permits the usage of neural network topology together with fuzzy logic. It not only includes the characteristics of both methods, but also eliminates some disadvantages when used alone. Any linear or nonlinear function can be approximated using ANFIS. The NFS has the capacity to store human expertise and learning. In the last decade, NFS has been proposed for modelling and control of complex nonlinear industrial processes (Vieira *et al.*, 2004; Rezaeeian *et al.*, 2008; Wu *et al.*, 2008; Navghare *et al.*, 2011). The fast neural network LOLIMOT was introduced to determine the structure of a local linear input-output model description from experimental data by Nelles (1999). The learning phase of this network is quite fast and more deterministic than a classical

neural network. The response time of a LOLIMOT network is much less. It was proposed for modelling of a heat exchanger (Fischer *et al.*, 1998) and the combustion process of a diesel engine (Hafner *et al.*, 2000) and Van der Vusse reactor (Widjiantoro *et al.*, 2003).

In this study, LOLIMOT and NFS are used to ascertain its effectiveness for modelling and control of an experimental pH neutralization process using MATLAB. The direct and inverse models are obtained using both the techniques. The internal model control scheme is used for control of the process.

EXPERIMENTAL SYSTEM: pH NEUTRALIZATION PROCESS

The pH neutralization process considered in this work is technically realized in a constant volume mixing tank with two input streams and one output stream. The schematic diagram and experimental plant of the pH control system used in this study are shown in Figures 1 and 2, respectively. It consists of tanks for acid, base and water with the volume of 8.5 litres and also has one mixing tank with the volume of 3.5 litres with constant speed stirrer. A level switch has been provided to maintain constant level in the acid and base tank. The flows of acid and base streams are regulated by normally closed type equal percentage pneumatic control valves. In this setup, the water tank acts as a disturbance tank and an electrically operated solenoid valve is used to change the water flow. The pH sensor (ABB - PT100 100PSI 7-3) is located at the top of the mixing tank with a pH transmitter (ABB- AX400). The pH transmitter, signal conditioning circuit and Data Acquisition (DAQ) module with NI- VISA RS 232 serial interface are connected to a personal computer (PC). Through this interface, bi-directional real time information like the actuating signal supplied by the controller and pH data for the control operation are passed. The DAQ module shown in Figure 3 is composed of I/V and V/I converter, 12 bit ADC and DAC and two channel analog input and output. The input channel of the DAQ module receives pH data from the pH transmitter - AX 400 in terms of mA, which is converted into 0-5 V by the I/V converter. This voltage is sent to the PC with a 12 bit resolution of the ADC. The output channel of the DAQ module receives the control signal from the PC through the DAC and V/I converter. The current signal from the V/I converter will actuate the pneumatic control valve through a current to pressure (I/P) converter.

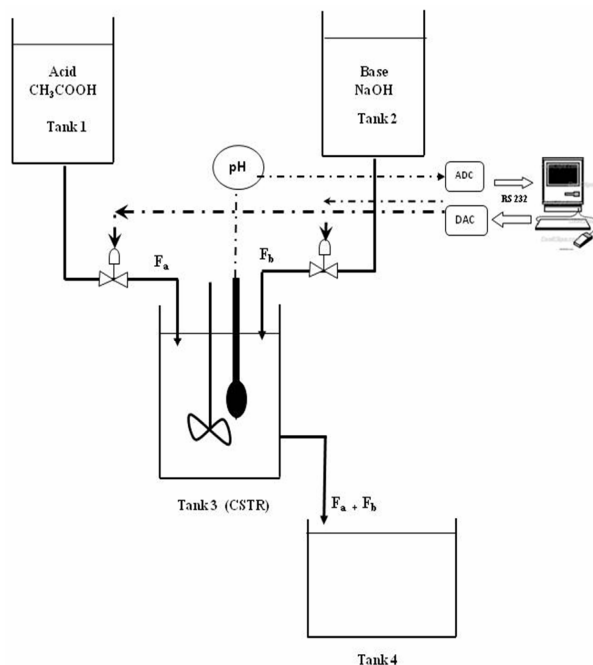


Figure 1: Schematic diagram of the pH neutralization process.

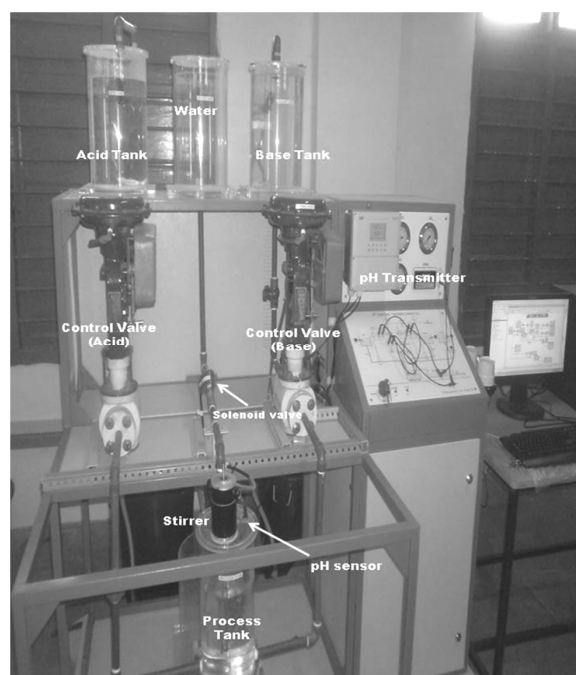


Figure 2: Experimental pH neutralization process.

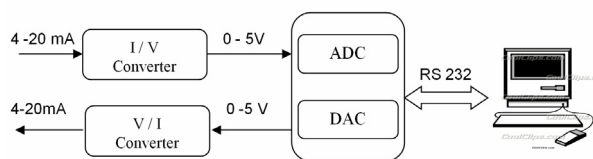


Figure 3: DAQ Module.

The 0.2 g mol L^{-1} acid stream of acetic acid (CH_3COOH) is pumped into the mixing tank at a constant flow rate (F_a) with 30% opening of the control valve. The 0.1 g mol L^{-1} base stream of sodium hydroxide (NaOH) is pumped into the mixing tank by a pneumatic control valve at a flow rate of F_b . The stirrer inside the mixing tank is used for mixing of the two streams. A neutral product and salt are produced by reaction between acid and base. The acid flow rate, F_a , introduces the disturbance in the process and the base stream flow rate, F_b , is used as a manipulated variable for control of the pH in the mixing tank. In this study, the maximum flow of acid and base streams is maintained around 1.2 L/min . The percentage of opening of the control valve is calibrated as flow rate for acid and base streams.

MODELLING OF THE pH NEUTRALIZATION PROCESS

Theoretical modelling of a pH process is too complex because of the severe nonlinear behaviour of the system. As to the characteristics of the experimental pH process, the nominal operating range of pH is from 5 to 12 and the nominal operating range of the base flow rate (F_b) is from 20% to 100% because the minimum value of the base flow rate is only obtained at 20% of the opening of the control valve. Here, the base and acid flow rate are measured in terms of % of the respective control valve opening. For modelling of the pH neutralization process, the acid is pumped into the mixing tank at a constant flow rate (F_a) of 30% opening of the control valve and the control signal (base flow rate F_b) is varied in the range of 20% to 70% opening of the control valve by a variable amplitude pseudo-random function. For more than 70% opening of the base flow control valve, the pH value of the solution in the mixing tank reaches around 13 and does not reduce further upon variation of the base flow rate for a long period of time. Hence, the operating range of base flow rate F_b was fixed in the range of 20% to 70% opening of control valve. The training and validation data sets for this study were obtained in an open loop using perturbations on the base flow-rate (F_b) as shown in Figure 8, from which variation of the pH was obtained from 5 to 12. This region exhibits the nonlinear behaviour of the process.

The data set for identification contained 10000 samples with a sampling time of 10 sec. From the training and validation data sets, it can be seen that the rise and fall times are different and the time delay

of the system is constant and equal to one sampling period. The structures of the direct and inverse models for training are illustrated in Figures 4 and 5. These structures are the main solutions for training models illustrated in the literature (Deshpande and Ash, 1988; Hunt and Sbarbaro, 1991). The representation of a nonlinear dynamic system using the black-box approach at discrete time $k + 1$ is a nonlinear function of past inputs (n_u) and past outputs (n_y) and noise and is in the form of a NARX (Nonlinear Auto Regressive with Exogenous inputs) model, given by Eq. (1)

$$y(k) = f\left(y(k-1), y(k-2), \dots, y(k-n_y), u(k-1), u(k-2), \dots, u(k-n_u)\right) + e(k) \quad (1)$$

In the above equation, n_y and n_u represent the number of previous output samples used and the number of previous control (input) signal samples used, respectively. The direct and inverse models of the pH process are obtained with $n_y = n_u = 2$. The values of n_y , n_u and time delay (n_k) are evaluated using the system identification toolbox of MATLAB based on statistical criteria, viz., the Akaike information criterion (AIC) and Final Prediction Error (FPE) value, as shown in Table 1.

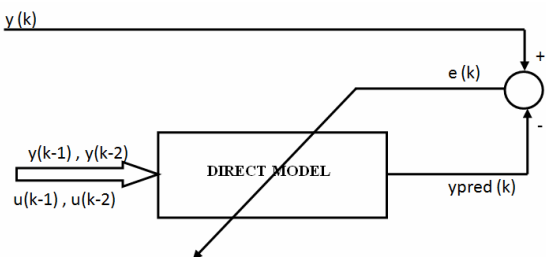


Figure 4: Structure of the direct model for training.

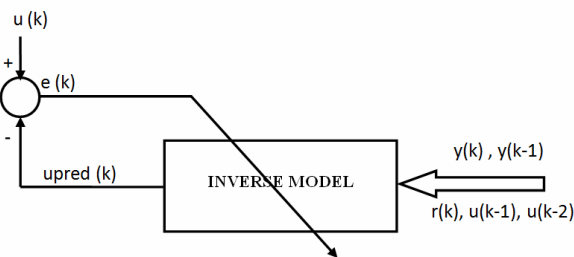


Figure 5: Structure of the inverse model for training.

The prediction of the pH in the direct model of the process at time k is given by Eq. (2).

$$ypred(k) = f(y(k-1), y(k-2), u(k-1), u(k-2)) \quad (2)$$

The prediction of the control signal at time instant

k in the inverse model is given by Eq. (3)

$$upred(k) = g(y(k), y(k-1), r(k), u(k-1), u(k-2)) \quad (3)$$

where $r(k)$ is the reference signal at time k . The local linear model tree (LOLIMOT) and neuro-fuzzy systems approaches like ANFIS were used to obtain the direct and inverse models of the nonlinear pH process.

Table 1: FPE and AIC for different values of n_y , n_u and n_k .

Model	n_u	n_y	n_k	FPE	AIC
M1	1	1	1	0.0023	-6.0731
M2	2	2	1	0.0017	-6.3762
M3	3	2	1	0.0017	-6.3520
M4	3	3	1	0.0017	-6.3485
M5	4	3	1	0.0018	-6.3421
M6	4	4	1	0.0017	-6.3704
M7	5	2	1	0.0018	-6.3096
M8	5	5	1	0.0017	-6.3096

Local Linear Model Tree

Local Linear Model Tree (LOLIMOT) is one kind of fast local linear neural networks, because the learning phase is quite fast and more deterministic than a classical neural network. This network was introduced by Nelles (1999). The main idea of this network is to approximate the non-linear function with multiple piecewise linear models. It is an extended radial basis function network and it is obtained by replacing the output layer weights with a linear function of the network inputs. Thus, each neuron represents a local linear model with its corresponding validity function. In addition to this, the radial basis function network is normalized, that is, the sum of all validity functions for a specific input combination sums up to one. The local linear models are interpolated by Gaussian functions as the weighing functions. The Gaussian weighing functions determine the regions of the input space where each neuron is active. The input space of the net is divided into M hyper-rectangles, each represented by a linear function (Hafner *et al.* 2000).

The output y of the LOLIMOT network with p inputs u_1, u_2, \dots, u_p is calculated by summing up the contributions of all M local linear models.

$$\hat{y} = \sum_{i=1}^M (w_{i0} + w_{i1}u_1 + w_{i2}u_2 + \dots + w_{ip}u_p) \Phi_i(u, c_i, \sigma_i) \quad (4)$$

where w_{ik} are the parameters of the i^{th} linear regression model, u is the input vector and Φ_i is the normalized Gaussian weighing function for the i^{th} model with center coordinate c_i and standard deviations σ_i .

$$\Phi_i(u, c_i, \sigma_i) = \frac{\Psi_i}{\sum_{j=1}^M \Psi_j} \quad (5)$$

with

$$\Psi_i = \exp \left(-\frac{1}{2} \left(\frac{(u_1 - c_{i1})^2}{\sigma_{i1}^2} + \frac{(u_2 - c_{i2})^2}{\sigma_{i2}^2} + \dots \right) \right) \quad (6)$$

The procedure of this network is split into two loops. In the outer loop, the structure of the local model network is optimized for the number of neurons and the partitioning of the input space. The network structure is optimized by a tree construction algorithm that determines the centers and standard deviations of the weighing functions (Figure 6). The inner loop estimates the parameters and structure of the local linear models by a local modelling technique (Pedram *et al.*, 2006).

The LOLIMOT network partitions the input space in hyper rectangles. The weighting function of the corresponding linear model is placed in the

center of each hyper rectangle. The standard deviations are selected proportional to the size of the hyper rectangle. This creates the size of the validity region of a local linear model proportional to its hyper rectangle extension. A model may be valid over a wide operating range of one input variable but only in a small range of another one.

At the outer loop of the each iteration, the local linear model with the worst local error in Eq. (7) is bisected into two new ones. Local error is the sum of the squared errors weighted with the corresponding weighing function Φ_i over all the data samples N .

$$J_{local} = \sum_{j=1}^N \Phi_i(u(j))(y(j) - \hat{y}(j))^2 \quad (7)$$

The possible cuts in all dimensions are tested and the one with the highest performance is selected as a new hyper rectangle. In the inner loop, the parameters of the local linear models are calculated by a local weighted least-squares technique. Some unique features of the LOLIMOT network are, local estimation of the local linear model parameters, considering only the worst local linear model for a split and freezing of the validity function values for extrapolation. This network is very fast and robust. In addition, the computational demand increases only linearly with the number of local models due to the local estimation approach (Widjiantoro *et al.*, 2003; Deshpande and Ash, 1988).

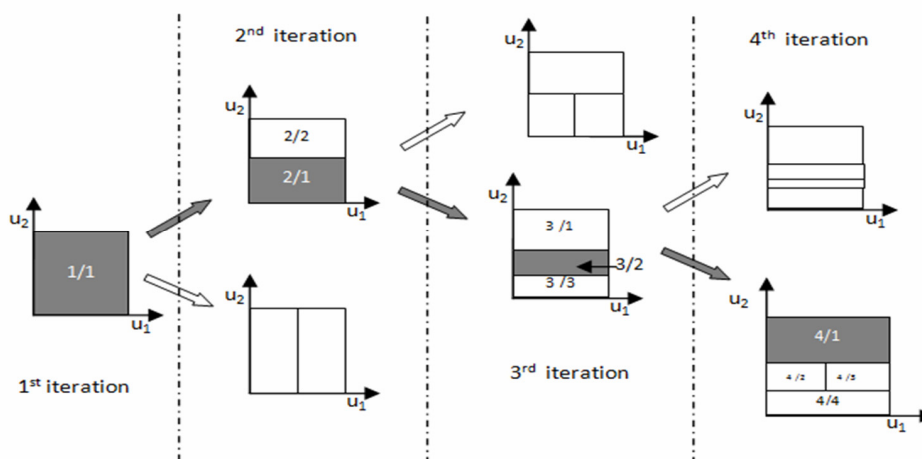


Figure 6: Operation of the LOLIMOT training algorithm in the first four iterations.

Neuro-Fuzzy System

A fuzzy inference system (FIS) can use human knowledge by storing its crucial components in a rule base, and perform fuzzy reasoning to infer the overall output value. The source of if-then rules and its membership function largely depend on *a priori* knowledge about the process. On the other hand, there is no logical way to transfer the knowledge of human experts to the knowledge base of a FIS. So, there is a need for adaptation of the rule base to generate the output within the specified error. The combination of FIS and ANN results in the NFS. The NFS takes advantage of the capacity that FIS has to store human expertise and the capacity of learning of the ANN (Vieira *et al.*, 2004). The learning algorithm is applied to a FIS by forming a special ANN-like architecture known as an adaptive neuro-fuzzy inference system (ANFIS). In this paper, ANFIS was taken as the NFS, because of its robustness and fast convergence.

The architecture of the ANFIS (Jang, 1993) is illustrated in Figures 7(a) and 7(b). Assume that the FIS under consideration has two inputs u, x and one output z . For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is as follows in Eq. (8).

Rule 1:

If u is $A1$ and x is $B1$, then $z_1 = p_1u + q_1x + r_1$,

Rule 2:

If u is $A2$ and x is $B2$, then $z_2 = p_2u + q_2x + r_2$. (8)

The reasoning mechanism for the Sugeno model is illustrated in Figure 7(a) and the corresponding equivalent ANFIS architecture is shown in Figure 7(b).

The output z in Figure 7(b) can be written as

$$\begin{aligned} z &= \frac{w_1}{w_1 + w_2} z_1 + \frac{w_2}{w_1 + w_2} z_2 \\ &= \bar{w}_1(p_1u + q_1x + r_1) + \bar{w}_2(p_2u + q_2x + r_2) \\ &= (\bar{w}_1u)p_1 + (\bar{w}_1x)q_1 + (\bar{w}_1)r_1 \\ &\quad + (\bar{w}_2u)p_2 + (\bar{w}_2x)q_2 + (\bar{w}_2)r_2 \end{aligned} \quad (9)$$

The ANFIS architecture has five layers, as shown in Figure 7(b). The first layer has the membership functions for each input and rule, the second layer determines the firing strength of each rule, and the third layer normalizes the value of the firing strength. The nodes in the fourth layer are adaptive and perform the consequence of the rules. The parameters in this layer are referred to as consequent parameters. Finally, the overall output can be expressed as a linear combination of the resultant parameters in the fifth layer.

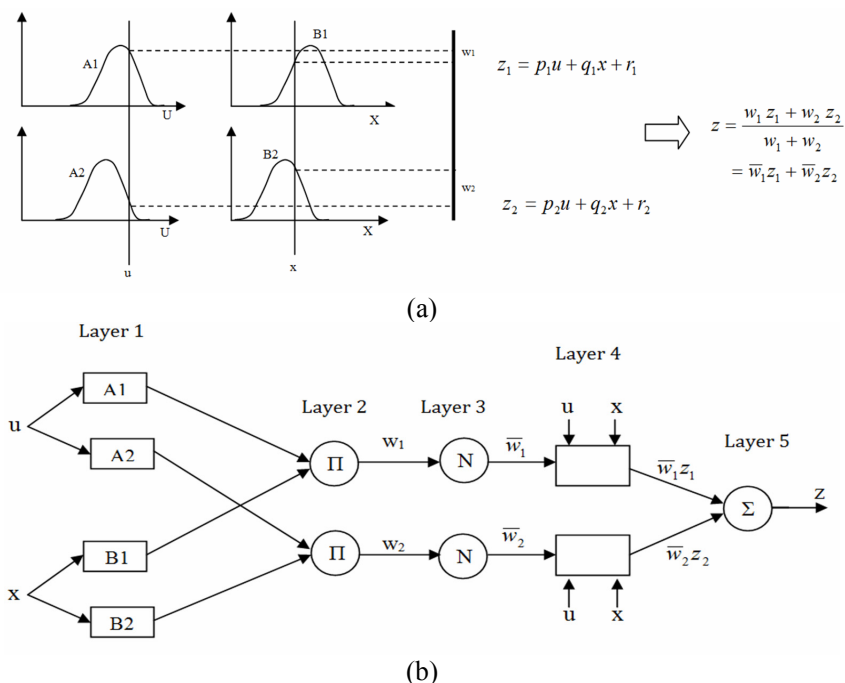


Figure 7: (a) A two-input first-order Sugeno fuzzy model with two rules; (b): ANFIS architecture for a two-input first-order Sugeno fuzzy model with two rules.

The learning algorithm of ANFIS is composed of two phases. In the forward phase, the node output values go forward until layer 4 and the resultant parameters are identified by the least squares method. In the backward phase, the output errors are propagated backward and the parameters are updated by the gradient descent method.

LOLIMOT and ANFIS Structures for Modelling

The ability of a LOLIMOT network to identify nonlinear systems was ascertained by Nelles (1999) and Widjiantoro *et al.* (2003). In general, the choice of model structure depends on the projected use of the model. In this study, the model is used for prediction purposes. The local linear models are chosen to be the nonlinear ARX (NARX) model. The NARX requires past input signals and past real output signals as an input to form the predicted output. The structure of NARX is very simple and allows the use of linear optimization techniques.

The standard ANFIS structure is used to obtain the direct and inverse model. The structure contains sixteen rules, four inputs with two bell-shaped membership functions and one output that is a linear function of the resultant parameters

The good regressor for LOLIMOT and ANFIS based modelling is obtained via the process of iteration with initial values. The regressor is taken as a row vector $[n_u \ n_y \ n_k]$. Here, n_u is the number of past input signals that affects the next output. n_y is the

number of past output signals and n_k is the time delay. The initial value of the regressor is taken as $[2 \ 2 \ 1]$ for both techniques, similar to the orders used in the NARX model. The direct and inverse models are obtained with training and test data sets using the LOLIMOT and ANFIS techniques. The identification procedures use a fuzzy control toolbox for use with MATLAB for the LOLIMOT and ANFIS models (Molsa, 2007).

Comparison of the LOLIMOT and ANFIS Models

The input – output data collected from the experimental pH process are shown in Figure 8. Out of the 10000 samples collected, 5000 samples of data are used for training and the remaining 5000 samples of data are used for validation. The number of training epochs is fixed. The stopping criterion for training is a minimum value of the mean square error (MSE). In this study, the minimum value of MSE was taken as $1e^{-6}$. The MSE was used as the criterion for the training and test data sets to compare the accuracy of the model. The simulation results of the direct and inverse model are shown in Tables 2 and 3, respectively. The simulation output of LOLIMOT and ANFIS based direct and inverse models are shown in Figures 9(a)-12(a). The enlarged versions of the simulation output of the LOLIMOT and ANFIS based direct and inverse models are presented in Figures 9(b)-12(b) to illustrate the difference between the mentioned variables.

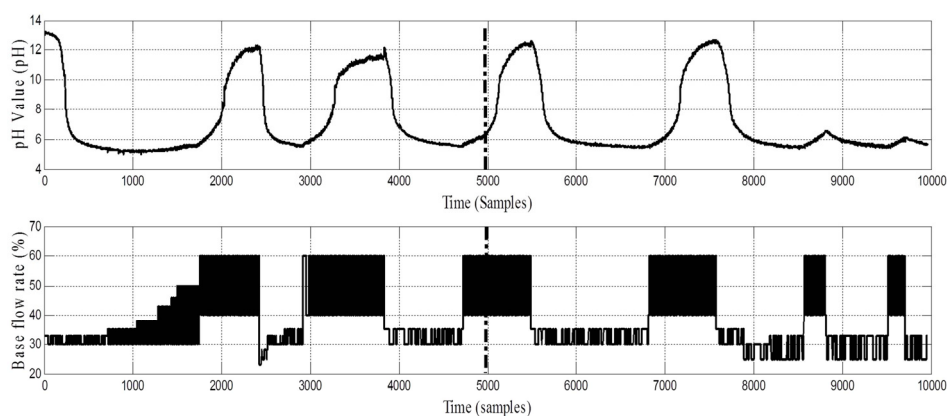


Figure 8: Training and Validation Data.

Table 2: MSE for the Direct Model.

Direct Model	MSE	
	Training	Testing
LOLIMOT	$1.620e^{-3}$	$1.589e^{-3}$
ANFIS	$1.744e^{-3}$	$1.739e^{-3}$

Table 3: MSE for the Inverse Model.

Inverse Model	MSE	
	Training	Testing
LOLIMOT	10.11	10.24
ANFIS	11.97	12.21

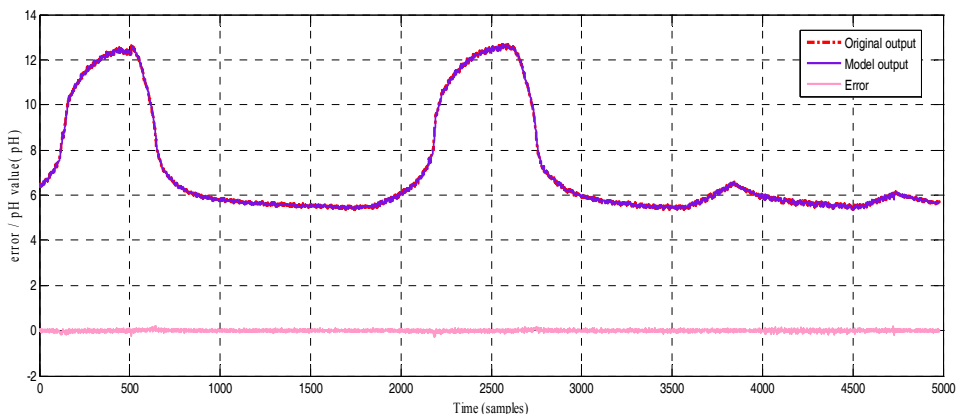


Figure 9(a): LOLIMOT direct model output after validation.

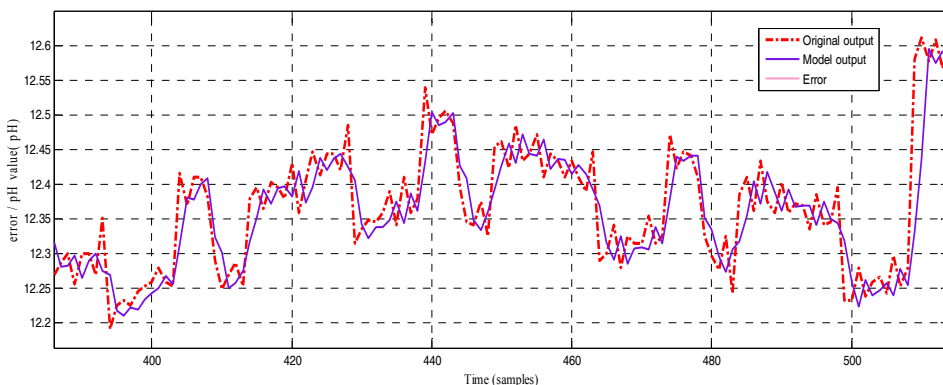


Figure 9(b): Enlarged View of the LOLIMOT direct model output after validation.

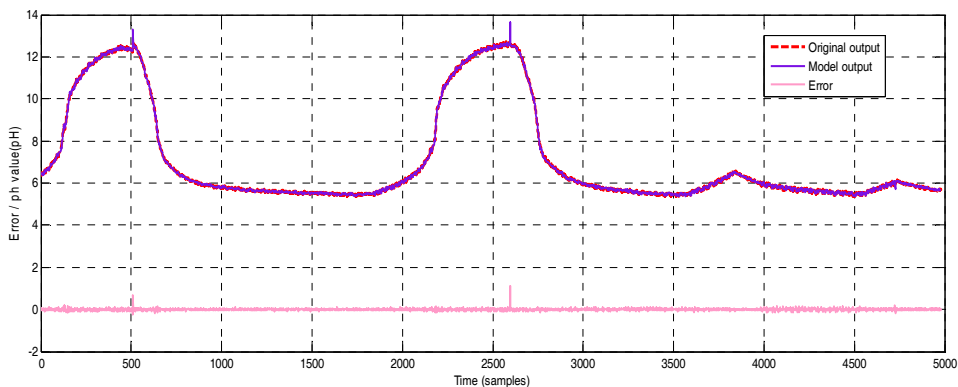


Figure 10(a): ANFIS direct model output after validation.

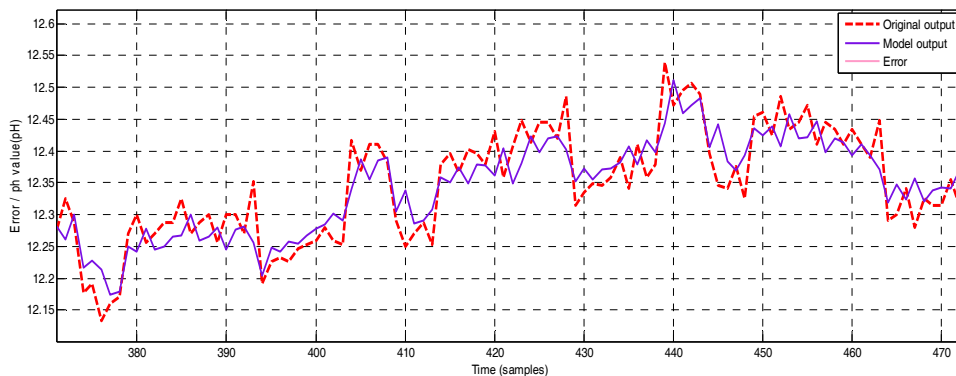


Figure 10(b): Enlarged View of the ANFIS direct model output after validation.

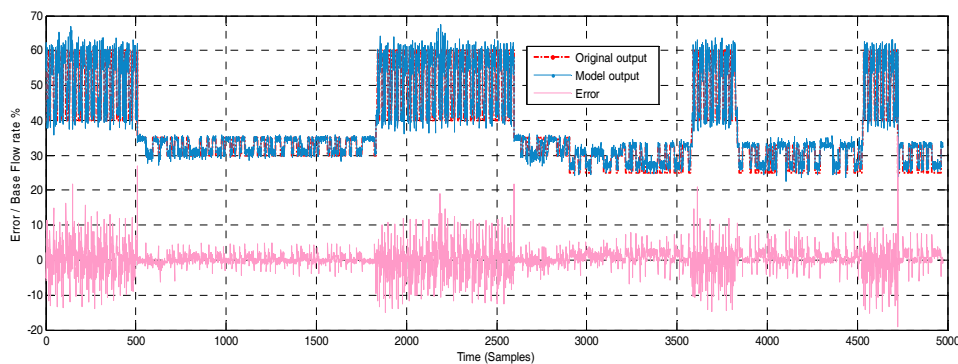


Figure 11(a): LOLIMOT inverse model output after validation.

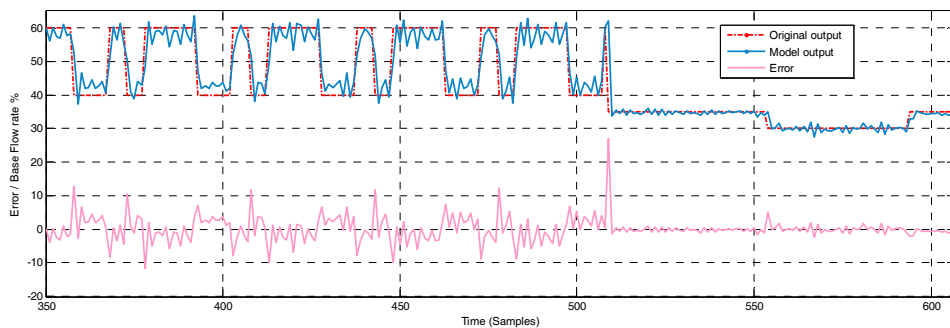
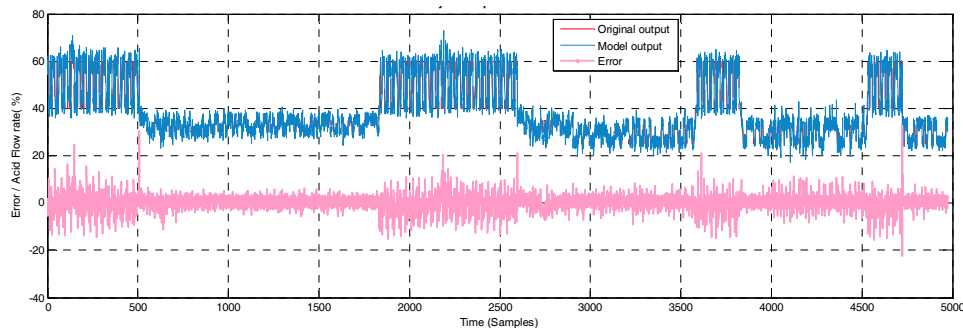
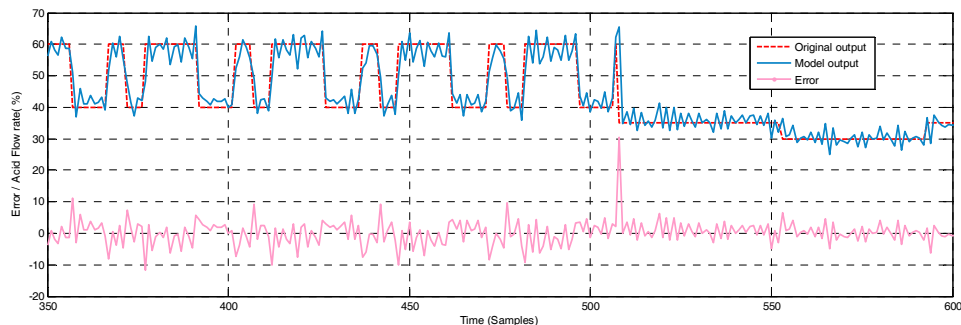


Figure 11(b): Enlarged view of the LOLIMOT inverse model output after validation.



(a)

Figure 12(a): ANFIS inverse model output after validation.



(b)

Figure 12(b): Enlarged view of the ANFIS inverse model output after validation.

From Tables 2 and 3, it can be seen that the LOLIMOT structure achieved a smaller MSE compared to the ANFIS structure. The obtained models can also be compared in terms of complexity by measuring the number of parameters, number of training epochs and training time. The results are summarized in Table 4 for the direct and inverse models. From Table 4, it can be seen that the number of parameters and training time are less in LOLIMOT compared to ANFIS because of the standard ANFIS structure.

Table 4: Complexity comparison.

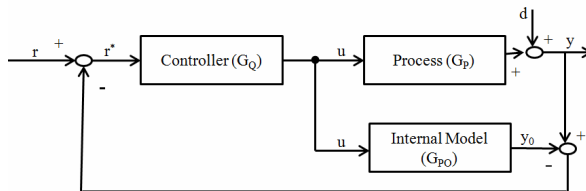
Properties	Direct Model		Inverse Model	
	LOLIMOT	ANFIS	LOLIMOT	ANFIS
No. of parameters	19	104	37	104
No. of training epochs	100	50	50	50
Training time (sec)	5	9	8	17

CONTROL OF A pH NEUTRALIZATION PROCESS

To test the obtained direct and inverse models for both the LOLIMOT and ANFIS architectures, the internal model control (IMC) strategy was used. The internal model control (IMC) method was introduced by Garcia and Morari (1985) and thorough research and development took place during the past decades (Garcia and Morari, 1985; Morari and Zafiriou, 1989). The IMC method relies on the internal model principle, which states that if any control system contains within it, implicitly or explicitly, some representation of the process to be controlled, then a perfect control is simply achieved. In particular, if the control scheme has been developed based on the exact model of the process, then perfect control is

theoretically possible. The IMC method includes an internal model and an internal model controller, which consists of the inverse internal model and a filter. It provides good tracking and robust performance.

The standard structure of the IMC scheme, where the process model plays an explicit role in the control structure compared to the standard control loop, is shown in Figure 13. The IMC structure has some advantages over a conventional feedback control loop. For the nominal, the case plant (G_p) is equal to the plant model (G_{p0}); for instance, the feedback is affected only by the disturbance 'd', such that the system is effectively under open loop and hence no stability problems can arise. Also, if the process transfer function (G_p) is stable, the closed loop system is stable for any stable controller. Moreover, the controller can simply be designed as a feed-forward controller in the IMC scheme.

**Figure 13:** Standard structure of the IMC scheme.

In IMC design, controller G_Q can be designed as an open loop controller, the ideal choice of the controller is the inverse model of the process, which provides good tracking performance. The IMC structure used is shown in Figure 14, in which the inverse model is connected in series with the plant and the direct model (internal model) is connected in parallel with the plant. The differences between the model output and plant output is considered as error, $e(k)$. The error, $e(k)$ is fed back and subtracted from the set point or reference signal, $r(k)$.

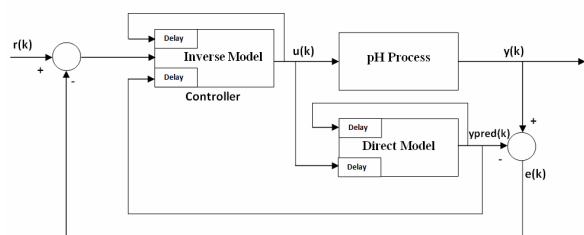


Figure 14: Structure of the Internal Model Control.

Implementation of IMC in Real-Time Control Action

The controllers are directly tested in the experimental pH neutralization process, after the simulation results are considered satisfactory. Figures 15

and 16 show the results of LOLIMOT and ANFIS internal model controllers for the experimental pH neutralization process. The real-time control action is performed for four different step changes in set point such as pH values of 7, 8, 9 and 11. The LOLIMOT and ANFIS based IMCs track the reference signal satisfactorily and produce reasonable control inputs.

Minimization of the mean square error (MSE) was used as a metric measure function for comparing the results of these two approaches. In LOLIMOT based IMC, as shown in Figure 15, the control actions for the changes from 7 to 8, 8 to 9 and 9 to 11 are smooth and track the set point with small overshoot, but the pH change from 11 to 9 takes more time to settle down.

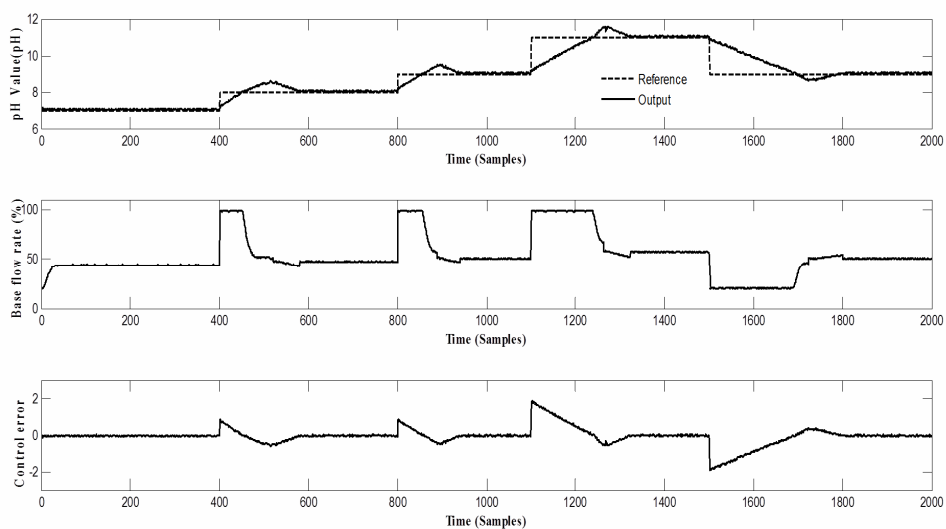


Figure 15: IMC using the LOLIMOT based model.

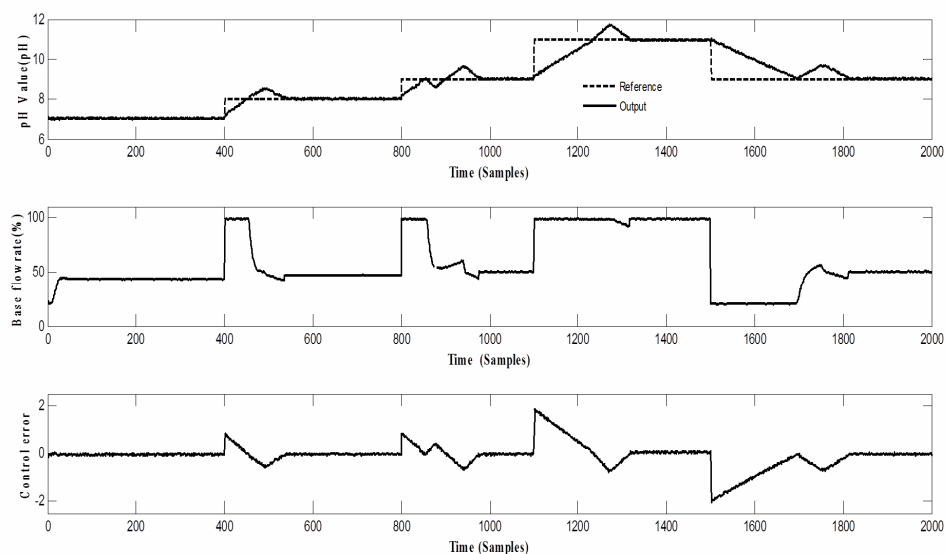


Figure 16: IMC using the ANFIS based model.

Overall results of the LOLIMOT based IMC are reasonable. For the ANFIS based IMC, as shown in Figure 16, the control actions for the changes from 7 to 8 and 9 to 11 are reasonably good, but for changes from 8 to 9 and 11 to 9 some oscillations occur in output and also the settling time is high. From Figures 15 and 16, it can be seen that the output is close to the set point, which shows that the model can capture dynamics of the nonlinear process very well. From Table 5, it can be concluded that the LOLIMOT based IMC gave better performance than the ANFIS based IMC.

Table 5: Performance comparison of the controllers.

Controller	MSE
IMC - LOLIMOT	0.2235
IMC - ANFIS	0.2475

CONCLUSIONS

In this study, modeling of the experimental pH process is based on LOLIMOT and ANFIS structures and real-time control action is performed using the IMC structure for both the models. In the direct and inverse modelling of the pH process, the MSE of the training and testing are close in the LOLIMOT and ANFIS structures. The error achieved is smaller when using a LOLIMOT than with ANFIS. The time taken for the training is also higher in ANFIS than LOLIMOT because the standard ANFIS structure used is not very flexible. The real time control action using IMC structure with both the approaches produced good results. However, the control action with the LOLIMOT model gives a slightly better result than the ANFIS model. In general, both the approaches are valid options for modelling of complex and non-linear real-time systems.

ACKNOWLEDGEMENT

This work was performed at the Advanced Process Control Laboratory (DST- FIST sponsored), Kalasalingam Academy of Research and Education, Krishnankoil, Srivilliputtur.

REFERENCES

Babuska, R., Verbrugge, H., Neuro-fuzzy methods for nonlinear system identification. *Annual Reviews in Control*, 27, 73-85 (2003).

- Billings, S. A., Identification of nonlinear systems – a survey. *Proceedings of IEE, Part D: Control Theory and Applications*, 127(6), 272-285 (1980).
- Deshpande, P. B., Ash, R. H., *Computer Process Control with Advanced Control Applications*. 2nd Revised Edition, ISA (1988).
- Doherty, S. K., Control of pH in chemical processes using artificial neural networks. Ph.D Thesis, Liverpool John Moores University (1999).
- Fink, A., Nelles, O., Isermann, R., Nonlinear internal model control for MISO systems based on local linear neuro-fuzzy models. In *Proceedings of 15th Triennial World Congress International Federation of Automatic Control*, Barcelona, Spain (2002).
- Fischer, M., Nelles, O., Isermann, R., Adaptive predictive control of a heat exchanger based on a fuzzy model. *Control Engineering Practice*, 6(2), 259-269 (1998).
- Garcia, C. E. and Morari, M., Internal model control-2: Design procedure for multivariable systems. *Ind. Eng. Chem. Process Des. Develop.*, 24(3), pp. 427-484 (1985).
- Hafner, M., Schuler, M., Nelles, O., Isermann, R., Fast neural networks for diesel engine control design. *Control Engineering Practice*, 8, 1211-1221 (2000).
- Han, M., Han, B., Guo, W., Process control of pH neutralization based on adaptive algorithm of universal learning network. *Journal of Process Control*, 16, 1-7 (2006).
- Hunt, K. J., Sbarbaro, D., Neural networks for linear internal model control. *IEEE Proceedings-D, Control Theory and Applications*, 138(5), 431-438 (1991).
- Jang, J. S. R., ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on System, Man, and Cybernetics*, 23(3), 665-685 (1993).
- Molsa, J., Limited toolbox for fuzzy identification and control for use with MATLAB. Tampere University of Technology (2007).
- Morari, M., Zafiriou, E., *Robust Process Control*. Prentice Hall, Englewood Cliffs, USA (1989).
- Mwembeshi, M. M., Kent, C. A., Salhi, S., A genetic algorithm based approach to intelligent modelling and control of pH in reactors. *Computers & Chemical Engineering*, 28(9), 1743-1757 (2004).
- Nahas, E. P., Henson, M. A., Seborg, D. E., Non-linear internal model control strategy for neural network models. *Computers & Chemical Engineering*, 16, 1039-1057 (1992).
- Navghare, S. R., Bodhe, G. L., Shruti, Design of Adaptive pH Controller using ANFIS. *International Journal of Computer Applications*, 33(6), 41-48 (2011).

- Nelles, O., Nonlinear System Identification with Local Linear Neuro-Fuzzy Models. Dissertation, Technische Universität Darmstadt, Shaker Verlag, Aachen, (1999).
- Pedram, A., Jamali, M. R., Pedram, T., Fakhraie, S. M., Lucas, C., Local linear model tree (LOLIMOT) reconfigurable parallel hardware. *International Symposium on Parallel Computing in Electrical Engineering*, 13(17), 198-201 (2006).
- Radhakrishnan, T. K., Valarmathi, K., Devaraj, D., Intelligent techniques for system identification and controller tuning in pH process. *Brazilian Journal of Chemical Engineering*, 26(01), pp. 99-111 (2009).
- Rezaeeian, A., Yousefi-Koma, A., Shasti, B., Doosthoseini, A., ANFIS modeling and feed forward control of shape memory alloy actuators. *International Journal of Mathematical Models and Methods in Applied Sciences*, 2(2), 228-235 (2008).
- Richalet, J., Industrial applications of model based predictive control. *Automatica*, 29(5), 1251-1274 (1993).
- Rodriguez, J. L., Loparo, K. A., Modelling and identification of the pH processes. In *Proceedings of the American Control Conference*, 6, 5483-5488 (2004).
- Sivaraman, E., Arulselvi, S., Neuro modeling and control strategies for a pH process. *International Journal of Engineering Science and Technology*, 3(1), 194-203 (2011).
- Syafie, S., Tadeo, F., Martinez, E., Model-free learning control of neutralization processes using reinforcement learning. *Engineering Applications of Artificial Intelligence*, 20, 767-782 (2007).
- Valarmathi, K., Devaraj, D., Radhakrishnan T. K., Real-coded genetic algorithm for system identification and controller tuning. *Applied Mathematical Modelling*, 33, 3392-3401 (2009).
- Vieira, J., Dias, F. M., Mota, A., Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study. *Engineering Applications of Artificial Intelligence*, 17, 265-273 (2004).
- Widjiantoro, B. L., Liong, T. H., Nazaruddin, Y. Y., Purnama B. S., LOLIMOT Based Model Predictive Control. *Jurnal Teknologi*, 17(1), 37-44 (2003).
- Wior, I., Boonto, S., Abbas, H. S., Modeling and control of an experimental pH neutralization plant using Neural Networks based Approximate Predictive Control. In *Proceedings of 1st Virtual Control Conference*, Denmark (2010).
- Wu, X. J., Zhu, X. J., Cao, G. Y., Tu, H. Y., Non-linear modeling of a SOFC stack based on ANFIS identification. *Simulation Modelling Practice and Theory*, 16, 399-409 (2008).