

ANFIS based Distillation Column Control

R. Sivakumar

Department of Electronics and Instrumentation
Engineering
St. Joseph's College of Engineering
Chennai, India

K. Balu

Department of Chemical Engineering
Sri Venkateswara College of Engineering
Chennai, India

ABSTRACT

This paper presents a control strategy that combines the predictive controller and neuro-fuzzy controller type of ANFIS. An Adaptive Network based Fuzzy Inference System architecture extended to cope with multivariable systems has been used. The neuro-fuzzy controller and predictive controller are works parallel. This controller adjusts the output of the predictive controller, in order to enhance the predicted inputs. The performance of the control strategy is studied on the control of Distillation Column problem. The results confirmed the control quality improvement with MPC and multi-loop PID controller.

Keywords

ANFIS, Neural modeling, MPC, Distillation Column, PID controller.

1. INTRODUCTION

The controller always aims to achieve the process variable to the given set point value. This is the main task of the properly designed controller. The controller should also provide some flexibility in case of change in set point and disturbances. Today there are many methods for designing intelligent controllers, such as predictive controller, fuzzy control, neural networks and expert systems. Various combinations of these controllers give a number of design possibilities.

Artificial Neural Networks (ANNs) and Fuzzy Logic (FL) have been increasingly in use in many engineering fields since their introduction as mathematical aids by McCulloch and Pitts, 1943, and Zadeh, 1965, respectively. Being branches of Artificial Intelligence (AI), both emulate the human way of using past experiences, adapting itself accordingly and generalizing. While the former have the capability of learning by means of parallel connected units, called neurons, which process inputs in accordance with their adaptable weights usually in a recursive manner for approximation; the latter can handle imperfect information through linguistic variables, which are arguments of their corresponding membership functions.

Although the fundamentals of ANNs and FL go back as early as 1940s and 1960s, respectively, significant advancements in applications took place around 1980s. After the introduction of back-propagation algorithm for training multi-layer networks by Rumelhart and McClelland, 1986, ANNs has found many applications in numerous inter-disciplinary areas [1-3]. On the other hand, FL made a great advance in the mid 1970s with some successful results of laboratory experiments by Mamdani and Assilian [4]. In 1985, Takagi and Sugeno [5] contributed FL with a new rule-based modeling technique.

Obtaining a mathematical model for a system can be rather complex and time consuming as it often requires some assumptions such as defining an operating point and doing linearization about that point and ignoring some system parameters, etc. This fact has recently led the researchers to exploit the neural and fuzzy techniques in modelling complex systems utilizing solely the input-output data sets. Although fuzzy logic allows one to model a system using human knowledge and experience with IF- THEN rules, it is not always adequate on its own. This is also true for ANNs, which only deal with numbers rather than linguistic expressions. This deficiency can be overcome by combining the superior features of the two methods.

To achieve the most accurate set point, an appropriate extensions and improvements in the intelligent control is needed. The predictive control is added with neuro – fuzzy controller to get this task. The prediction is a future value of model. Using this states and suitable optimization criterion, it is possible to get more accurate values and the controlling becomes more effective.

This paper uses a control strategy that enhances a fuzzy controller with a self-learning capability for achieving prescribed control objectives. In this sense, an extended Adaptive-Network-based Fuzzy Inference System (ANFIS) architecture is employed [6, 7], so that a fuzzy inference system is built for achieving a desired input/output mapping. The learning method used allows the tuning of parameters both of the membership functions and the consequents in a Sugeno-type inference system.

2. MODEL PREDICTIVE CONTROL

2.1 Classical MPC

The main idea behind MPC-type controllers is illustrated in Figure 1 for a SISO system. At sampling time k , a set of m future manipulated variable moves (control horizon) are selected, so that the predicted response over a finite horizon p (prediction horizon) has certain desirable characteristics. This is achieved by minimizing an objective function based on the deviation of the future controlled variables from a desired trajectory over the prediction horizon p and the control energy over the control horizon m . The MPC optimization is performed for a sequence of hypothetical future control moves over the control horizon and only the first move is implemented [8]. The problem is solved again at time $k + 1$ with the measured output $y(k + 1)$ as the new starting point. Model uncertainty and unmeasured process disturbances are handled by calculating an additive disturbance as the difference between the process measurement and the model prediction at the current time step. For the measured disturbances

it is assumed that the future values will be equal to the current values [12].

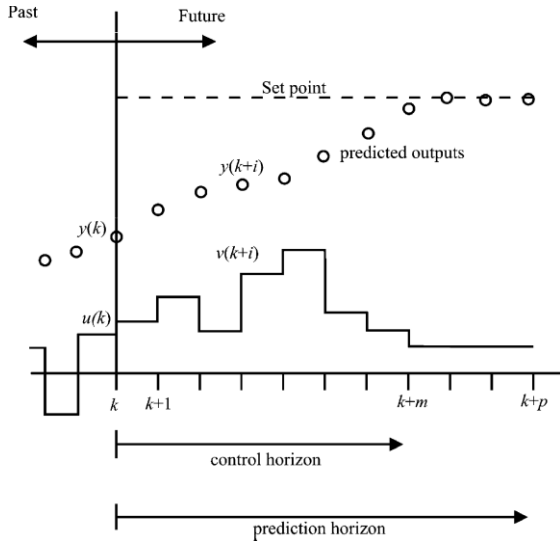


Figure 1 Schematic view of MPC

2.2 Neural Networking Modeling

Neural network needs the system input and output data (Figure 2). Neural network is connected parallel to the system and they share input. Second input to neural network is error between system and neural network output. Based on this error, new parameters of neural network are adjusted. Sampling period of input and output data, network architecture, training algorithm and train periods number are affect the quality of trained neural network.

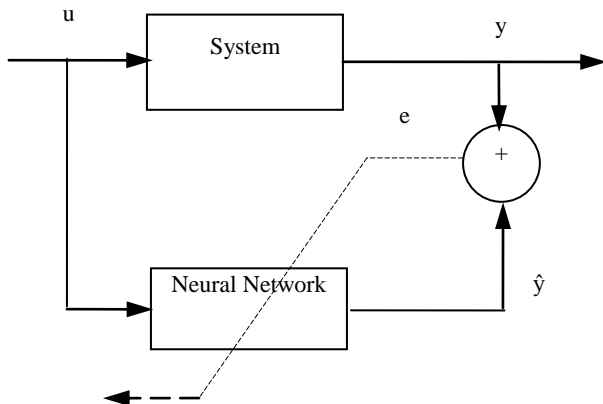


Figure 2 Generation of neural model

The neural model of the system is given by:

$$y_k = f(u_{k-1}, u_{k-2}, u_{k-3}, y_{k-1}, y_{k-2}, y_{k-3})$$

3. NEURO – FUZZY CONTROLLER

The neural predictive controller can be extended with neuro-fuzzy controller, connected in parallel (Figure 3). Neuro-fuzzy

systems, which combine neural networks and fuzzy logic, have recently gained a lot of interest in research and application. A specific approach in neuro-fuzzy development is the ANFIS (Adaptive Network-based Fuzzy Inference System).

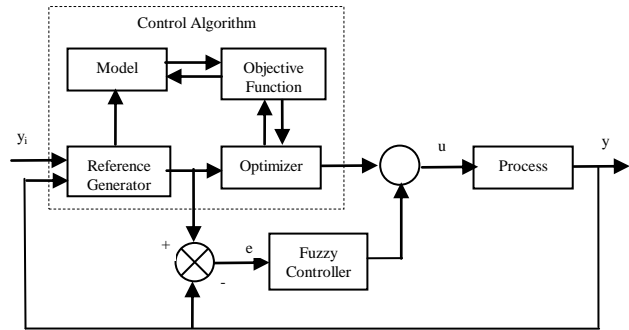


Figure 3 Neuro – Fuzzy Control scheme

ANFIS uses a feed forward network to search for fuzzy decision rules that perform well on a given task. Using a given input-output data set, ANFIS creates an Fuzzy Inference System for which membership function parameters are adjusted using a combination of a backpropagation and least square method. The ANFIS architecture of the first-order Takagi-Sugeno inference system is shown in Figure 4.

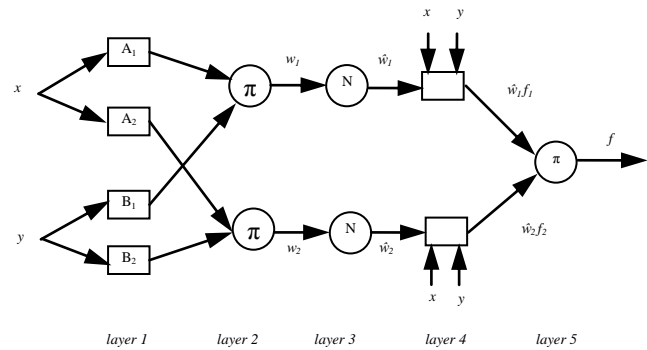


Figure 4 System Architecture of ANFIS

The entire system consists of five layers, and the relationship between the input and output of each layer is summarized as follows:

Layer 1: Every node i in this layer is an adaptive node with a node output defined by

$$o_{1,f} = \mu_{A_i}(x) \text{ for } i = 1, 2 \text{ or}$$

$$o_{1,i} = \mu_{B_{i-2}}(y) \text{ for } i = 3, 4$$

where x or y is the input to the node, A_i or B_{i-2} is a fuzzy set associated with this node, characterized by the shape of the membership function in this node. Parameters in this layer are referred to as premise (antecedent) parameters.

Layer 2: Every node in this layer is a fixed node labeled Π , which multiplies the incoming signals and output product.

$$o_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \text{ , } i = 1, 2$$

each output node represents the firing strength of a rule

Layer 3 : Every node of this layer is a circular node labeled N. the i th node calculates the ratio of the i th rule's firing strength to the sum of all rule's firing strengths.

$$o_{3,i} = \hat{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2$$

output of this layer is called as normalized firing strengths.

Layer 4 : Node i in this layer computes the contribution of the i th rule towards the model output, with the following node function

$$o_{4,i} = \hat{w}_i \hat{f}_i = \hat{w}_i (p_i x + q_i y + r_i)$$

where \hat{w}_i is the output of layer 3 and (p_i, q_i, r_i) is the parameter set. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ that computes the overall outpour as the summation of all incoming signals.

$$\text{Overall output} = O_5 = \sum_i \hat{w}_i f_i$$

4. DISTILLATION COLUMN

The distillation column is probably the most popular and important process studied in the chemical engineering literature. Distillation is used in many chemical processes for separating feed streams and for the purification of final and intermediate product streams. Most column handle multicomponent feeds, but many can be approximated by binary or pseudobinary mixtures. The objective is to split a liquid two component mixture into its fractions throughout stripping and rectifying processes.

Figure 5 shows the most important loops of a binary distillation. Acceptable operation of a binary distillation column normally requires the following control objects:

- Control of the composition of the distillate
- Control of the composite of the bottom products
- Control of the liquid hold-up in reflux drum
- Control of the liquid hold-up at the base of the column

The first two control objectives characterize the two product streams, where the other two objects are required for operational feasibility. A mathematical model of a binary distillation column based on various simplifying assumptions was used in our studies. This multivariable control strategy may be implemented for either time domain or Laplace transform models. For multivariable systems such as distillation columns having multiple delays, a commonly employed linear model takes the form:

$$y(s) = G_p(s)u(s) + G_d(s)d(s) \quad (1)$$

where y is a vector of outputs, u a vector of controls and d a vector of disturbance variable.

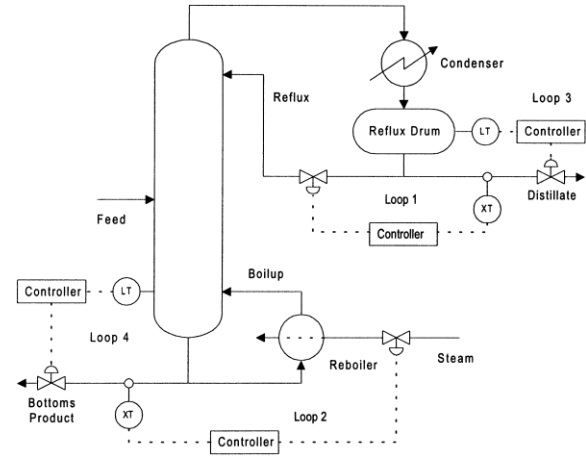


Figure 5 Control of distillation column

The first 2 x 2 MIMO process is presented by Wood and Berry (1973). The study was performed on a 9 inch diameter, 8 tray column equipped with a total condenser and a basket type reboiler. The required control action for the manipulative variables in the composition loops, reflux and steam flow, were cascaded to the set points of the appropriate flow controllers. The transfer function characterizing the column dynamics were established by pulse testing. Parameters of the assumed first order plus time delay transfer function were determined from the transient data. The time delays were established and the gains and time constants are determined by least squares fit employing Rosenbrok's direct search technique. The process transfer function matrix of the distillation process is given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \frac{12e^{-s}}{16.7s+1} & \frac{-18.9e^{-2s}}{12s+1} \\ \frac{6.6e^{-7s}}{19.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2)$$

5. DESIGN OF ANFIS CONTROLLER

The basic idea behind the design of neuro-adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modeling procedure to learn information about data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input-output data. ANFIS constructs an input-output mapping based on both human knowledge (in the form of fuzzy if-then rules) and simulated input-output data pairs. It serves as a basis for building the set of fuzzy if-then rules with appropriate membership functions to generate the input output pairs.

The parameters associated with the membership functions are open to change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the ANFIS is modeling the input-output data for a given parameter set. Once the gradient vector is obtained, back propagation or hybrid learning algorithm can be applied in order to adjust the parameters.

ANFIS can be used in modeling, estimating and controlling studies in chemical engineering processes similar to other artificial intelligence methods such as NNs and Fuzzy Logic (FL). In this paper, the designed ANFIS is utilized as an estimator and a controller. Estimation is done for compositions from the temperature measurements in distillation column whereas adaptive control strategy that needs no separate process network model with ANFIS controller is utilized in a control system for set point tracking problem and disturbance rejection problem.

5.1 ANFIS as an Estimator

ANFIS can be used for the estimation of some dependent variables in chemical process. The designed ANFIS estimator is used to infer the compositions from measurable tray temperatures distillation column. In estimator design process, different ANFIS structure are constructed and trained to find the architecture that gives the best performance as an estimator.

As a first step to design an estimator, training data sets should be generated to train the estimator networks. These data sets consist of estimator inputs and desired output values. They are produced from the process input output data. Since, ANFIS is a data processing method, it is important that the input-output data must be within the sufficient operational range including the maximum and minimum values for both input and output variables of the system. If this is not provided, estimator performance cannot be guaranteed and thus the designed estimator will not be accurate. Having generated the training data, estimators that have different architectures are trained with the obtained data sets.

Performances of the trained estimators are evaluated with model simulations and best estimator architecture is obtained. These simulations are made to verify and to generalize the ANFIS structures. Verification is done to show how good the estimator structure learned the given training data. This is carried out by simulating the column models with specific initial process inputs used in obtaining training data sets. Generalization capabilities of the estimators are found with other simulations in which input process variables are in operational range but not used in training data formation.

ANFIS estimator design consists of two parts: constructing and training. In constructing part, structure parameters are determined. These are type and number of input Membership Functions (MFs), and type of output MF. Any of several MFs such as Triangular, Trapezoidal and Gaussian can be used as an input MF. Frequently used MFs in literature are the Triangular and Gaussian. For this reason, they are chosen as input MF type in this study. Number of MFs on each input can be chosen as 3, 5, and 7 to define the linguistic labels significantly. Effective partition of the input space is important and it can decrease the rule number and thus increase the speed in both learning and application phase. Output MFs can be either a constant or in linear form. Both of these two forms are used for the output MF in this study. Having described the number and type of input MFs, the estimator rule base is constituted. Since, there is no standard method to utilize the expert knowledge; automatic rule generation (grid partition) method is usually preferred (Castillo and Melin 2001). According to this method, for instance, an ANFIS model with two inputs and three MFs on each input would result in $32 = 9$ Takagi-Sugeno fuzzy if-then rules automatically. Although this method can require much computational knowledge especially in

systems that have to be defined with many inputs, it is used in this study due to advantage of MATLAB software. Therefore, rule bases of the estimators are formed automatically with the number of inputs and number of MFs. After the ANFIS structure is constructed, learning algorithm and training parameters are chosen. As mentioned in the earlier in this chapter, backpropagation or hybrid learning can be used as a learning algorithm. The hybrid learning algorithm is used in this study. Parameters in the algorithm are epoch size (presentation of the entire data set), error tolerance, initial step size, step size decrease rate, and step size increase rate. Since there is no exact method in literature to find the optimum of these parameters a trial and error procedure is used.

MATLAB fuzzy logic toolbox is used to design ANFIS estimators' structures. Using the given training data set, the toolbox constructs an ANFIS structure using either a backpropagation algorithm alone, or in combination with least squares type of method (hybrid algorithm). ANFIS model can be generated either from the command line, or through the ANFIS editor GUI. In this study, ANFIS Editor GUI is used to generate the ANFIS models with the chosen design parameters in construction phase. Written MATLAB code is used to train the ANFIS structure in the training step.

The steps in ANFIS estimator design in this study utilizing the MATLAB fuzzy logic toolbox are as follows:

1. Generated training data is loaded to the Editor GUI.
2. Design parameters, number of input MF, type of input and output MF, are chosen. Thus, initial ANFIS structure is formed.
3. The code for the training is run with the initial structure.
4. ANFIS structure constituted after training is saved to use as an estimator.

5.1.1 Generation of Training Data

If the operating input-output data are outside their training data range, estimator will not operate accurately. As a result, the training data set should possess sufficient operational range including the maximum and minimum values for input-output variables. The data set should include data for each process variable, evenly distributed throughout the range for which estimation is desired. The maximum and minimum values of reflux and boilup rates in the column were determined by Bahar (2003) by looking at the closed-loop responses of the system. Thus, model simulations are done to obtain the input-output data by using these values. Then, tray temperature values and corresponding top and bottom product compositions are collected.

5.1.2 Training of ANFIS Estimators

Estimator structure design and training are realized using MATLAB software. First, generated training data is loaded using the GUI Editor. Then, with chosen design parameters, initial estimator structure is constructed. For example, if three triangular MFs are used for each input and constant output MF is chosen, GUI Editor determines the initial parameters of triangular MFs automatically using loaded data and constructs the initial Tri3con (three triangular MFs for each input and constant output MF) ANFIS structure. Trainings of the structures are done by MATLAB. All structures are trained in the same way only by changing the training data.

5.2 Simulations Results

After the training of the ANFIS structures, performances of estimators are investigated through the model for both the verification and generalization tests. These tests are made by utilizing the different estimator structures. The responses of the compositions to a 4% increase in reflux rate and a 5% increase in feed rate are obtained by simulations to verify the estimator's learning performances. Also, reflux rate and feed rate are increased by 5% and 7% respectively to see the generalization capabilities of the estimators. In all simulations, the Integral of the Absolute Error (IAE) are calculated as the performance criteria. Simulation results are given Tables 1 and 2. In Verification capabilities of the estimators can be followed considering the IAE scores giving how well these different estimator structures can generalize what they have learned.

Table 1 Verification: 5% increase in Feed Flowrate

Input MF	No. of input MF	Output MF	IAE score Top product	IAE score Bottom product
Triangular	3	Constant	0	0
Triangular	3	Linear	0	0
Triangular	5	Constant	0	0
Triangular	5	Linear	0	0
Triangular	7	Constant	0	0
Triangular	7	Linear	0	0
Gaussian	3	Constant	0	0
Gaussian	3	Linear	0	0
Gaussian	5	Constant	0	0
Gaussian	5	Linear	0	0
Gaussian	7	Constant	0	0
Gaussian	7	Linear	0	0

Table 2 Verification: 4% increase in Reflux Rate

Input MF	No. of input MF	Output MF	IAE score Top product	IAE score Bottom product
Triangular	3	Constant	0.00049	0.0038
Triangular	3	Linear	0.00017	0.0011
Triangular	5	Constant	0.00026	0.0035
Triangular	5	Linear	0.00015	0.0007
Triangular	7	Constant	0.00025	0.0032
Triangular	7	Linear	0.00019	0.0008
Gaussian	3	Constant	0.00172	0.0038
Gaussian	3	Linear	0.00014	0.0014
Gaussian	5	Constant	0.00037	0.0026
Gaussian	5	Linear	0.00012	0.0006
Gaussian	7	Constant	0.00027	0.0021
Gaussian	7	Linear	0.00048	0.0023

5.3 Control of the Distillation Column

The control of Distillation column is often considered to be the benchmark for nonlinear process control because of the highly nonlinear behavior exhibited by composition dynamics. In this study, it is aimed to use ANFIS as a controller in a composition control system. For this purpose, ANFIS controller is designed and used in an adaptive way in the distillation column control scheme. Figure 6 illustrates the adaptive control scheme for the distillation column control system under study. Developed model is used as a real plant in this scheme. The objective of the system is to control the composition by manipulating feed flowrate. Inputs to the controller at each sampling instant are plant and controller output, $X_i(k-1)$ and $F(k-1)$, respectively at previous sampling instant. Controller output is the new plant input, $F(k)$.

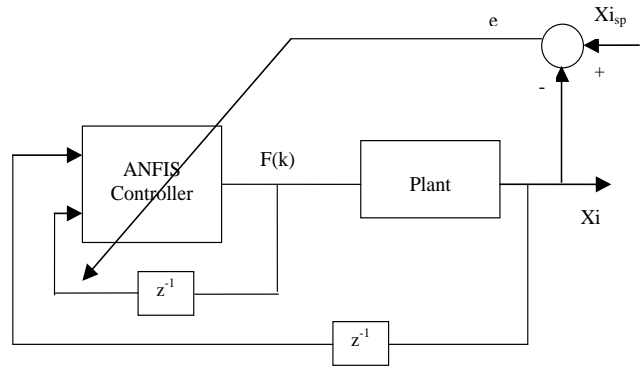


Figure 6 Adaptive composition control scheme

Table 3 Rule base of the controller

Rule	Controller rule base
1	If $X_i(k-1)$ is S and $F(k-1)$ is S then $F(k)$ is f1
2	If $X_i(k-1)$ is S and $F(k-1)$ is M then $F(k)$ is f2
3	If $X_i(k-1)$ is S and $F(k-1)$ is B then $F(k)$ is f3
4	If $X_i(k-1)$ is M and $F(k-1)$ is S then $F(k)$ is f4
5	If $X_i(k-1)$ is M and $F(k-1)$ is M then $F(k)$ is f5
6	If $X_i(k-1)$ is M and $F(k-1)$ is B then $F(k)$ is f6
7	If $X_i(k-1)$ is B and $F(k-1)$ is S then $F(k)$ is f7
8	If $X_i(k-1)$ is B and $F(k-1)$ is M then $F(k)$ is f8
9	If $X_i(k-1)$ is B and $F(k-1)$ is B then $F(k)$ is f9

The performance of ANFIS controller is also compared with that of the NN controller for the same system under study. The process response with NN controller is presented in Figure 11 and ANFIS response is illustrated in Figure 12. The other controller responses like conventional multiloop PI controller and MPC is seen from the Figures 7 to 10. Although system responses follow the same trajectory, the process controlled with ANFIS controller is faster and reaches the steady state values with minimum oscillations in both top and bottom product composition control. Actually, both controllers, ANFIS and NN, use the backpropagation algorithm with constant learning rate, but performance of ANFIS is better

than that of NN. This point out that classical backpropagation algorithm is effective for ANFIS controller.

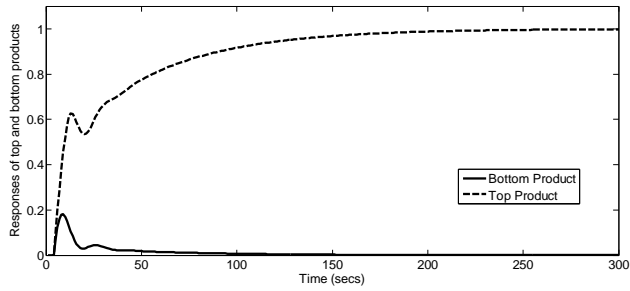


Figure 7 PI controller response of WB column with unit step change in bottom flow rate

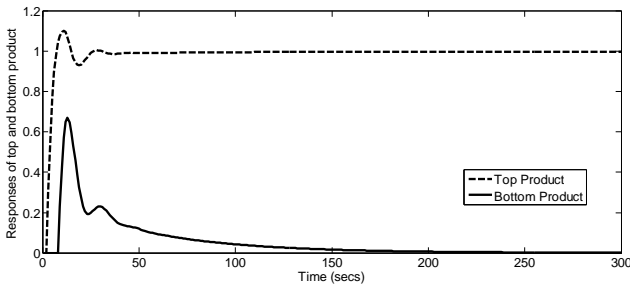


Figure 8 PI controller response of WB column with unit step change in distillation flow rate

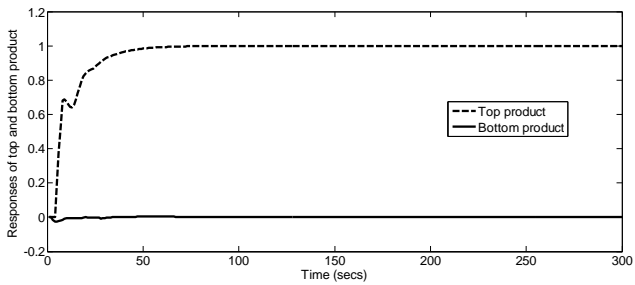


Figure 9 MPC response of WB column with unit step change in Bottom flow rate

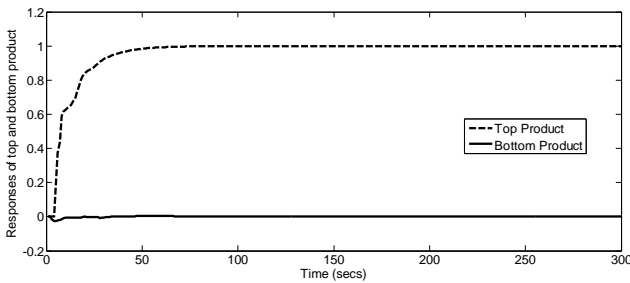


Figure 10 MPC response of WB column with unit step change in distillation flow rate

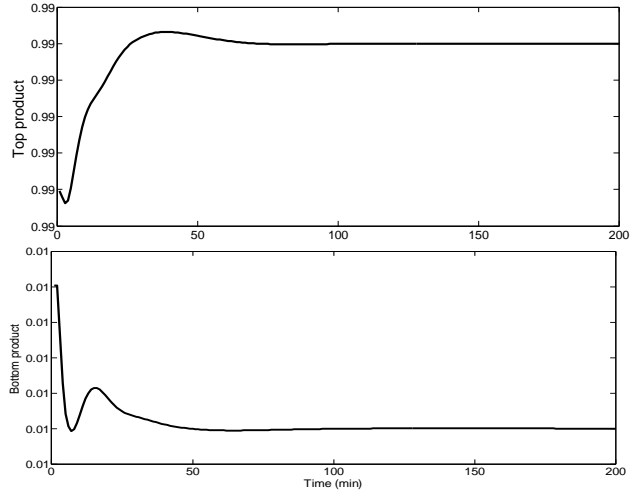


Figure 11 Composition control of distillation column using NN

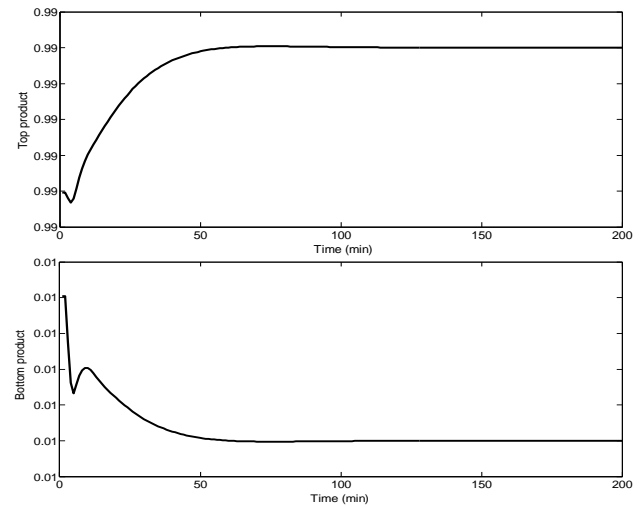


Figure 12 Composition control of distillation column using ANFIS

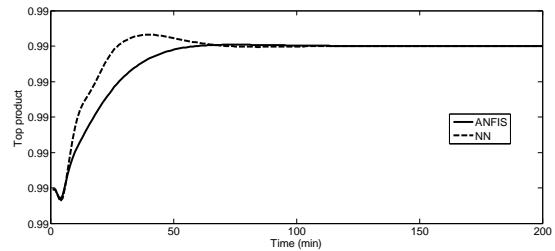


Figure 13 Comparison of ANFIS and NN controller performance of top product Composition control of distillation column

6. REFERENCES

- [1] Patterson, D. W. *Artificial Neural Networks – Theory and Applications*, Prentice Hall, 1996.
- [2] Engin, S. N., and Gulez, K. "A Wavelet Transform – Artificial Neural Networks (WT-ANN) based Rotating Machinery Fault Diagnostics Methodology", *IEEE NSIP' 99*, Falez Hotel, Antalya, Turkey, 1-3 June 1999
- [3] Staszewski, W. J. and Worden, K. "Classification of Faults in Gearboxes – Preprocessing Algorithms and Neural Networks," *Neural Computing and Applications* 5(3): 160-183, 1997
- [4] Mamdani, E. H. and Assilian, S., An Experiment in Linguistic Synthesis with a Logic Controller, *Int. J. Man – Machine Studies* 8, pp. 1 – 13, 1975
- [5] Takagi, S. and Sugeno, M, Fuzzy identification of fuzzy systems and it's application to modelling and control, *IEEE Trans. Systems Man Cybern.*, 15 pp 116-132, 1985
- [6] R. Jang, Self-learning fuzzy controllers based on temporal backpropagation, *IEEE Transactions on Neural Networks*, 1992, vol. 3(5), pp. 714–723
- [7] R. Jang, ANFIS: Adaptive-Network-Based fuzzy inference system, *IEEE Transactions on System, Man and Cybernetics*, 1993, vol. 23(3), pp. 665–684
- [8] Piotr Tatjewski and Maciej Ławryn' Czuk (2006), Soft computing in model-based predictive control', *International Journal of Applied Maths and Computer Science*, Vol. 16, No. 1, pp. 7–26, 2006
- [9] J. Fernandez de Canete , An Adaptive Neuro-Fuzzy Approach to Control a Distillation Column, *Neural Computing and Applications*, 2000, pp. 211–217.
- [10] Morteza Mohammadzaheri, Intelligent Modeling of MIMO Nonlinear Dynamic Process Plants for Predictive Control Purposes, *Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea*, 2008, pp 12401 - 12406
- [11] Suresh Manic K., Sivakumar R., Nerthiga V., Akila R. and Balu K., "Design of Plant Estimator Model using Neural Network", *International Journal for Computer Science and Network Security*, Vol. 9, No. 6, pp. 142-147, 2009.
- [12] Sivakumar R., Suresh Manic K., Nerthiga V., Akila R. and Balu K., "Application of Fuzzy Model Predictive Control in Multivariable Control of Distillation Column", *International Journal of Chemical Engineering and Applications*, Vol. 1, No. 1, pp. 39-42, 2010.