

Comparative Study Of Direct Inverse Neural Controller With Conventional PI (Designed At Lower Space Velocity) Controller For An Isothermal Continuous Stirred Tank Reactor With Input Multiplicities

Ballekallu Chinna Eeranna* G. Prabhaker Reddy**

*Dept. of petroleum Engineering ,Lords institute of engineering and Technology

**Dept. of Chemical Engineering, University College of Technology,
Osmania University, Hyderabad,

Abstract—In the present work, the Neural network (NN) based controller design has been implemented for a non-linear continuous stirred tank reactor processes with input multiplicities. Multilayer feed forward networks (FFNN) were used as direct inverse neural network (DINN) controllers. The training as well as testing database was created by perturbing the open loop process with pseudo random signals (PRS). Direct inverse neural network controller is analyzed to a continuous stirred tank reactor (CSTR) carrying out series and parallel reaction: $A \rightarrow B \rightarrow C$ and $2A \rightarrow D$ (Van de Vusse reaction) and exhibiting input multiplicities in the space velocity (i.e., manipulated variable), on the product concentration (B), (i.e. the controlled variable). Continuous Stirrer Tank Reactor which exhibits input multiplicities in space velocity on concentration. i.e., two values of space velocity will give the same value of concentration. The Performance of proposed direct inverse neural network controller and linear PI controller has been evaluated at lower and higher input. As the Neural network controller provides always the two values of space velocity for control action and by selecting the value at higher and lower to the operating point, it is found to give stable and faster responses than linear PI controller. Thus, direct inverse neural network control is found to overcome the control problems due to the input multiplicities at lower and higher input space velocities. It is interesting to note that the present neural network controller is giving superior performance like previously proposed nonlinear controller by authors (Reddy, G.P. and Chidambaram, M (1995)) to overcome the control problems due to input multiplicities.

Keywords: Direct inverse neural network control, CSTR, Input Multiplicities, space velocity and lower inputs

1. INTRODUCTION

Generally in a single Input and Single Output (SISO) process, more than one value of input variable producing

same value of output is known as input multiplicity. As given in Fig. 1. The two inputs U_1 & U_2 will produce the same output Y. Input multiplicities occur due to the competing effects in the processes. Dynamic and steady state behavior of the process with input multiplicity will remain distinct at different input values for the same output. Processes with multiple reactions, multi reactors or recycle structures are shown to exhibit input multiplicities (Koppel, L.B. (1982 & 1983)). Conventional linear PI controller will have control problems like instability, oscillatory and less economical (Dash, S.K. and Koppel, L.B.(1989)) due to input multiplicities in the process. The inherent nonlinearity of the production of cyclopentenol process often renders control difficult (Henson, M.A. and Seborg, D.E. (1982) and Agrawal, P and Lim (1984)). In the last two decades, a new direction to control has gained considerable attention. This new approach to control is called 'Intelligent control'. The term 'intelligent control' addresses to more general control problems. It may refer to systems, which cannot be adequately described by differential equations framework. There are three basic approaches to intelligent control: knowledge-based experts systems, fuzzy logic and neural networks. The term 'conventional control' refers to theories and methods that are employed to control dynamic systems whose behavior is primarily described by differential and difference equations. Among these intelligent controllers, data based direct inverse neural control has become popular tool for control of dynamic process, demonstrating the ability of handling non linearity. Many neural network controllers are the data based type where the controller's output response is described by a series of data generation, training and validation of the control. The procedure used to perform the learning process is called a learning algorithm, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective. Neural networks

Use sub symbolic processing characterized by microscopic interactions that eventually manifest themselves as macroscopic symbolic intelligent behavior. It is a computing system made up of a number of simple, highly interconnected nodes or processing elements, which process information by its dynamic state response to external inputs. The goal of a neural network is to map a set of input patterns into a corresponding set of output patterns. The network accomplishes this mapping by first learning from a series of past examples defining sets of input and output correspondences for the given system. The network then applies what it has learned to a new input pattern to predict the appropriate output.

Neurologists and artificial intelligence researchers have proposed a highly interconnected network of neurons or nodes for this purpose. By using a computer, information is input into a network of artificial nodes. These nodes mathematically interact with each other in ways unknown by the user. Eventually, based on the input, the network produces an output that maps the expected, macroscopic input-output pattern. The microscopic, sub symbolic processing that occurs in neural networks manifests itself as a macroscopic, symbolic, intelligent behavior.

Neural network derives its computing power through first it's massively distributed structure and second its ability to learn and therefore generalize. Generalization refers to the neural network producing reasonable output for inputs not encountered during training. These two information processing capabilities make it possible for neural networks to solve complex problems that are currently intractable. However neural network cannot provide the solution by working individually. Rather they need to be integrated into a consistent system engineering approach.

There are many advantages of neural networks like Information is distributed over a field of nodes, Neural networks have the ability to learn, Neural networks allow extensive knowledge indexing, Knowledge indexing is the ability to store a large amount of information and access it easily, The network stores knowledge in two forms a) the connection between the nodes b) the weight factors of these connections, Neural networks are better suited for processing noisy, incomplete, or inconsistent data and Neural networks mimic human learning processes.

In this work, the design and evaluation of, unlike model based nonlinear controller, the lesser computationally involved direct inverse neural network controller for an isothermal CSTR is presented to overcome the control problems associated with conventional PI controller due to input multiplicities.

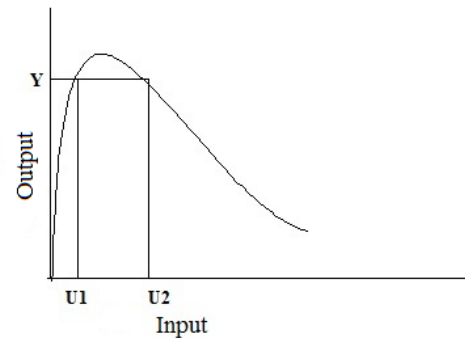
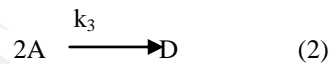
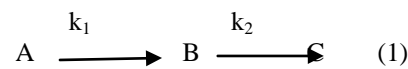


Fig.1.Steady state Behavior of Input Multiplicity process

2. DESCRIPTION OF CSTR WITH INPUT MULTIPLICITIES.

We consider here a continuous stirred tank reactor (CSTR) with the following isothermal series and parallel reactions (Van de Vusse, 1966);



The product B is the desired one. The mass balance equations for A and B are given by (Kravaris1990):

$$\begin{aligned} dX_1/dt &= -k_1 X_1 - k_3 X_1^2 + (C_{A,u} - X_1)u \\ dX_2/dt &= k_1 X_1 - k_2 X_2 + X_2 u \end{aligned} \quad (4)$$

Where,

$$X_1 = C_A, X_2 = C_B, u = F/V \quad (5)$$

and F is the flow rate (l/min), C_A and C_B are the concentration of A and B in the reactor (mol/l) and $C_{A,u}$ is the feed concentration of A (mol/l). The steady state solutions of equations (3) and (4) are given by

$$X_{1,s} = \{-b + [b^2 + 4k_3 C_{A,0} U_s]^{0.5}\} / 2k_1 \quad (6)$$

$$u_s = \{f_2 \pm [f_2^2 - 4 * f_1 * f_3]^{0.5}\} / (2 * f_1) \quad (7)$$

Where

$$b = k_1 + U_s$$

$$f_1 = d_2^2 - 1$$

$$f_2 = -2 * d_2^2 d_1 = 2 * k_1 + d_3$$

$$f_3 = d_1^2 - k_1^2$$

$$d_1 = (2k_1 k_2 X_2 + k_1^2) / k_1$$

$$d_2 = (2k_3 X_2 + k_1) / k_1$$

$$d_3 = 4k_3 C_{A,0}$$

The parameters considered for the present work are given by $k_1 = 0.8333 \text{ (min}^{-1}\text{)}$, $k_2 = 1.6667 \text{ (min}^{-1}\text{)}$, $k_3 = 0.16667 \text{ (mol}^{-1} * \text{min}^{-1}\text{)}$,

$C_{A,0} = 10 \text{ (mol/l)}$. The values of X_2 vs u_s is shown in Fig 1. Shows steady state input multiplicities in u_s on the product concentration ($X_{2,s}$). That is two same values of $X_{2,s}$ for example, $X_2 = 1.117$ can be obtained at $u = 0.5714$ and also at $u = 2.8746$. The gain is $+0.5848$ at $U = 0.5714 \text{ Min}^{-1}$ where as the gain is -0.1208 at $U = 2.8746 \text{ Min}^{-1}$.

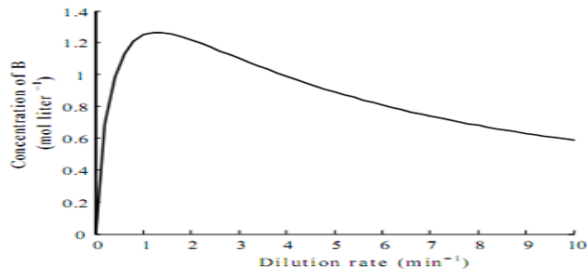


Fig.2. Steady state response space velocity of CSTR

3. DESIGN OF A DIRECT INVERSE NEURAL NETWORK CONTROLLER

The various steps of neural network based inverse model controller for the CSTR process are presented here. The Input-Output data is generated at lower and higher input put space velocities with pseudo random signal (PRS) shown in Fig.3 &4 for input signal of space velocities (u) of the CSTR system, the output responses shown in Fig.5&6 for the concentration of B, C_B is generated using the simulink model of the process with a sampling time of 0.2s for 1000 samples.

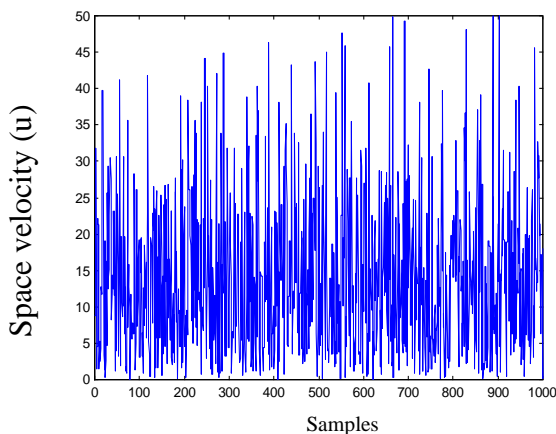


Fig 3. Pseudo Random Signal for Space velocity, u (process input) to CSTR process at lower input.

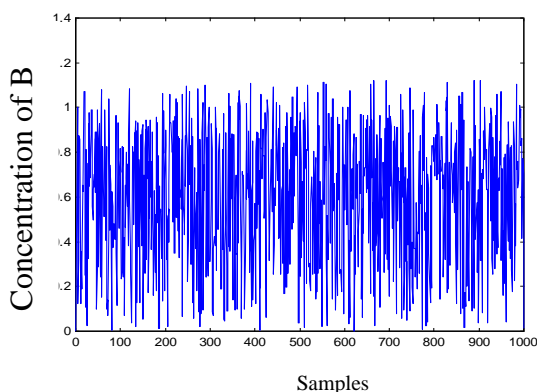


Fig 4. Process response/ output in C_B for the PRS input (u) shown in Fig 3

Inverse Neural network model shown in Fig.5 is basically the neural network structure representing the inverse of the system dynamics at the completion of training. The training procedure in this case is called

inverse modeling. Here the network is fed with past inputs, past outputs, present outputs. The network then predicts the controller output, $u(t)$ to make the output to reach the set point.

The final network representation of the inverse is given by $u(t) = f^{-1}[y(t+1), y(t), y(t-1), u(t-1)\dots]$ (8)

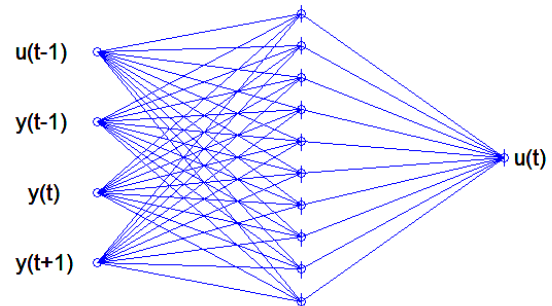


Fig 5. Structure of inverse neural network model

Training a neural network involves feeding the network with a set of known input-output patterns, & adjusting the network parameters until each input produces the appropriate output. In general, to train the neural networks, the weight factors are adjusted until the output pattern is calculated from the given input reflects the desired relationship. In this work, Levenberg – Marquardt method, a version of the back propagation algorithm is used for training the neural network. The objective of this algorithm is to minimize the sum of the squares of the errors. The most frequently used are two concepts of inverse neural network model architecture: (i) the ‘general training’ architecture and (ii) the ‘specialized training’ architecture. Often general training can be used to provide an initialization of the network so that on-line approach is only used for a fine-tuning of the controller. This is a highly recommended procedure.

In the present NN design work, the Levenberg-Marquardt training method is used. This method uses the past values of input, u & output, y , the control signal required for producing the desired output is found. The difference between expected u and the neural model output u_N is the error e_N which can be utilized for network learning. The input- output data obtained is divided into two parts each containing 500 data. The first 500 data are taken for training. Weights are initialized from input to hidden layer & hidden to output layer. Weight matrix $W11$ & $W12$ contains weights from input to hidden layer & weight matrix $W21$ & $W22$ contains weights from the hidden to output layer. The input matrix is chosen such that it contains the values of past input & output.

The weights obtained after training are used in validation & control. Here λ is the regularization factor which is chosen initially as 1. Based on the SSE the regularization factor is also updated. λ is increased if SSE has increased & decreased if the SSE has decreased. Once the

training is complete the final weights are stored & these are used for validating the network.

The criteria for choosing these values for the training parameters are SSE (Sum of the squares of the error). Initially number of nodes in hidden layer is taken as 1 & SSE computed. Then the nodes are increased and SSE is observed if it is decreasing then the nodes are increased till the value again starts decreasing. The numbers of nodes are chosen where it gives minimum SSE. Number of input nodes is taken as four and 10 hidden nodes based on selection of number of past input & output values. Number of outputs is taken as process outputs. Here only one output i.e. desired concentration of B is considered.

After training is completed the remaining 500 data are taken for validation. In this case, the NN model obtained is called inverse NN model. The network is next validated on the remaining set of data to evaluate the model. After suitable training model is obtained then the network is validated using the remaining data. This inverse model after training & Validation is taken for control. Here the inverse model itself acts as the controller.

4. RESULTS AND DISCUSSION

The performance of proposed direct inverse neural network controller and conventional PI controller to the CSTR with input multiplicities at lower and higher space velocities is evaluated using the closed loop block diagrams as shown in Figs 8 & 9. During the identification and control tasks the NNSYSID (M. Nørgaard, 1996) and NNCTRL (K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, 2000) toolboxes for MATLAB are used.

The simulation studies for servo and regulatory problem have been presented below at lower and higher space velocities. The parameters of conventional PI controller used in the simulation studies are, $K_c=1.25$, $\tau_i=0.5848$ min (Chidambaram M and Reddy, G.P. (1995))

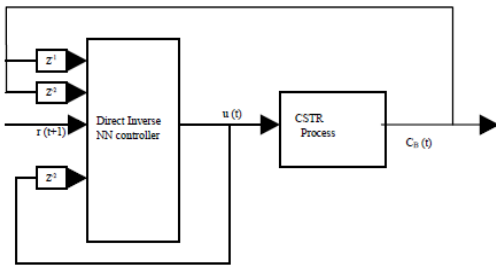


Fig. 6. Closed loop block diagram of Direct Inverse Neural Network Control of CSTR

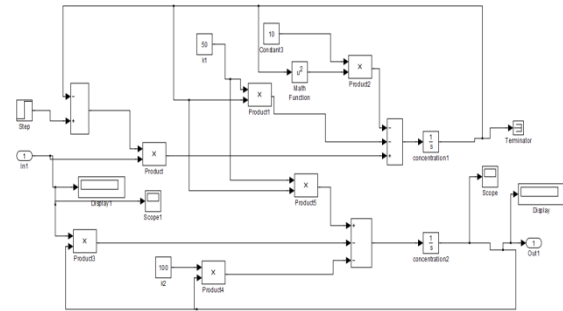


Fig7. Closed loop Simulink diagram of dynamic CSTR process

4.1.1 Servo problem:

The servo response has been studied by giving a step change in set point of concentration of B (C_B) with direct inverse neural network and PI controller.

At lower space velocity the servo problem has been analyzed by giving step change in set point of concentration B from 1.117 to 1.22 and the corresponding responses are shown fig.8. Direct inverse control shows stable response at about 3 min but it gives 0.03 offset where as PI reaches after 6 min. To overcome offset in direct inverse neural network control, integral action ($T_i=1.25$ min) is introduced. It works as a hybrid control and this concept is now applicable to all control studies. The hybrid control response is shown in fig 9 and its corresponding control action in terms of space velocity is shown in fig. 10

Fig 11 shows the step change in the set point of concentration B from 1.117 to 1.2. In this response the NNDIC+I reaches the set point at around 3min without any offset whereas PI is reaching the set point at 6min. The corresponding manipulated variable in terms of space velocity versus time behavior is shown fig 12

Fig 13 shows the step change in the set point of concentration B from 1.117 to 1.0. In this response the NNDIC+I reaches the set point before 2 min without any offset whereas PI is reaching the set point at 4 min. The corresponding manipulated variable in terms of space velocity versus time behavior is shown fig 14

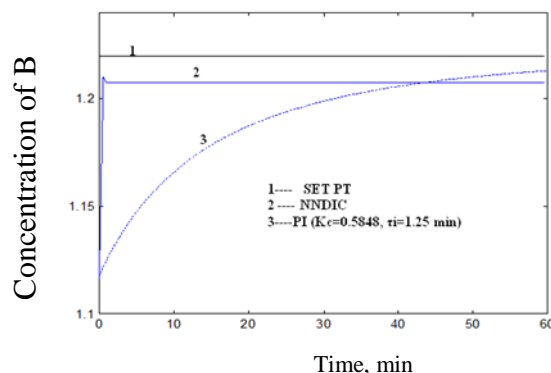


Fig.8. Closed loop response of concentration C_B for step change in set point from 1.117 to 1.22 at lower input

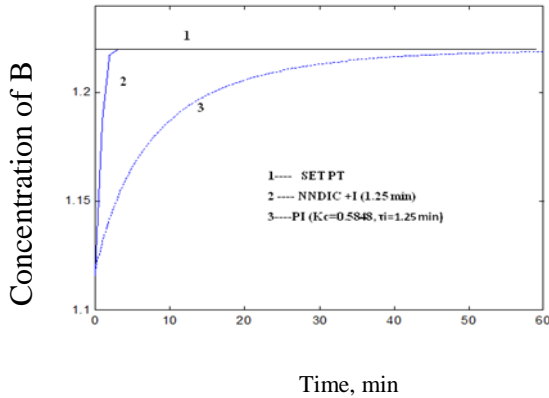


Fig.9. Closed loop response of concentration C_B for step change in set point from 1.117 to 1.22 at lower input

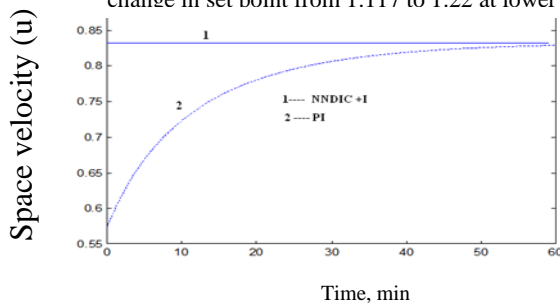


Fig 10. Control action in Space velocity Vs Time for the response shown in fig 7.9

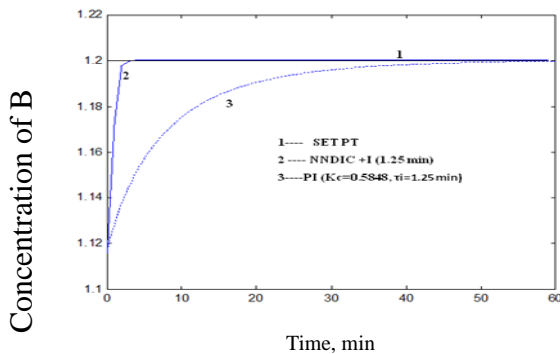


Fig.11. Closed loop response of concentration C_B for step change in set point from 1.117 to 1.2 at lower input

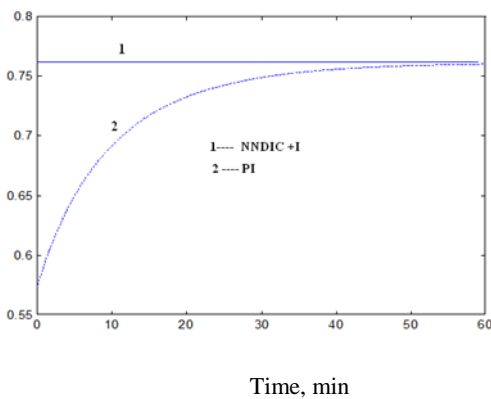


Fig 12. Control action in Space velocity Vs Time for the response shown in fig 7.11

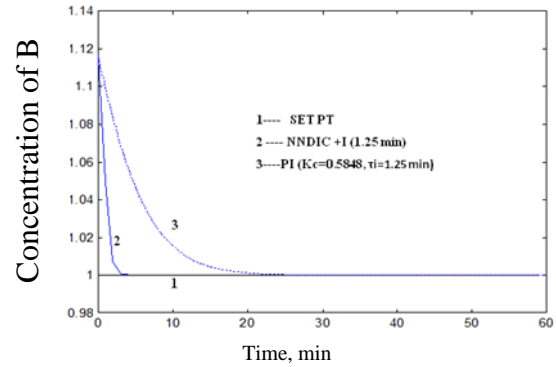


Fig 13. Closed loop response of concentration C_B for step change in set point from 1.117 to 1.0 at lower input

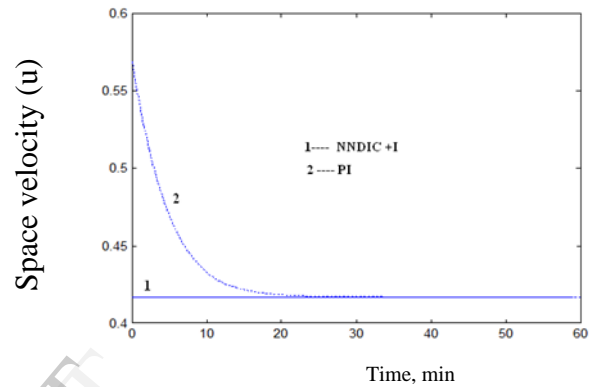


Fig 14. Control action in Space velocity Vs Time for the response shown in fig 7.13

4.1.2 Regulatory problem:

The regulatory response in concentration B of direct inverse neural network controller and PI controller for space velocity input of disturbance in feed concentration have been studied and they are stated below:

The regulatory response in concentration B of direct inverse neural network and conventional PI is shown fig 15 for a step change in feed concentration from 10 to 11(+10%). This fig shows that the response of the Direct inverse neural network controller is faster than that of the linear PI. Proposed neural network control has less deviation 1% whereas conventional PI controller has a larger deviation of about 8%. Direct inverse neural network controller has low settling time than the PI controller. The corresponding control actions for manipulated variable in terms of space velocity versus time behavior are shown in fig 16.

The regulatory response in concentration B of direct inverse neural network and conventional PI is shown fig 17 for a step change in feed concentration from 10 to 9(-10%). This fig shows that the response of the Direct inverse neural network controller is faster than that of the linear PI. Proposed neural network control has less deviation 2% whereas conventional PI controller has a larger deviation of about 8%. Direct inverse neural network controller has low settling time than the PI controller. The corresponding

control actions for manipulated variable in terms of space velocity versus time behavior are shown in fig 18.

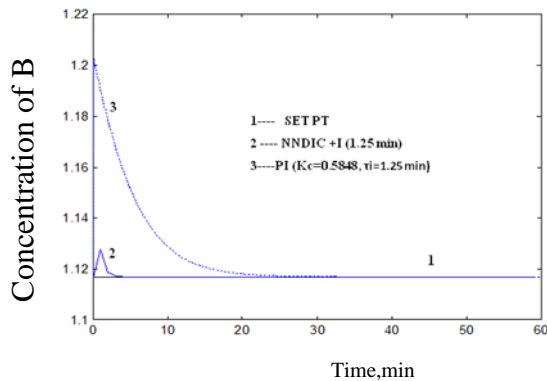


Fig 7.15 Closed loop response of C_B for a disturbance change in C_{AO} from 10 to 11 mol/l at lower input

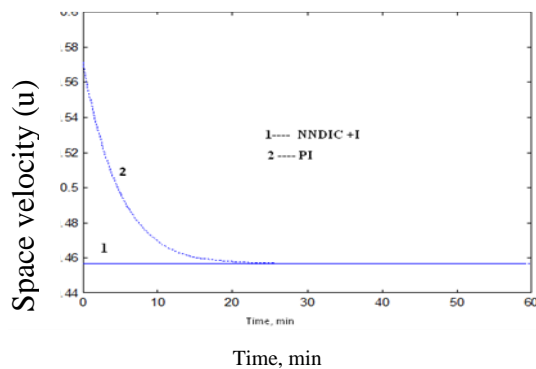


Fig 16. Control action in Space velocity Vs Time
For the response shown in fig 7.15

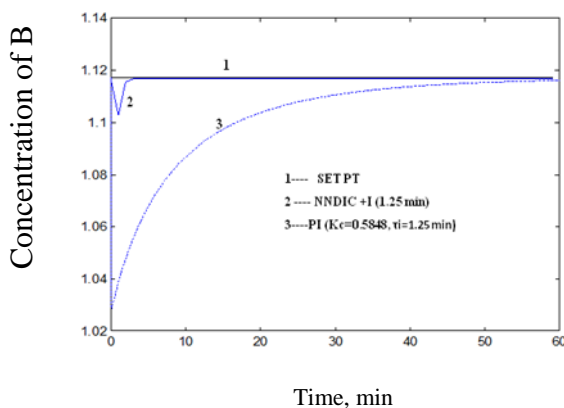


Fig 7.17 Closed loop response of C_B for a disturbance change in C_{AO} from 10 to 09 mol/l at lower input

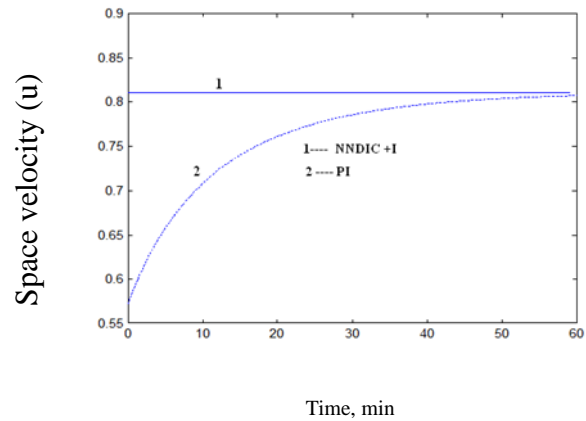


Fig 18. Control action in Space velocity Vs Time For the response shown in fig 7.17

5. CONCLUSION

For a continuous stirrer tank reactor with input multiplicities in space velocity, the performance of present Direct inverse neural network controller at lower input space velocities is found to much superior to that of the conventional PI controller at lower space velocity.

REFERENCES

- Chidambaram, M and Reddy, G.P. (1995) nonlinear control of systems with input multiplicities, Computers and Chemical Engineering, 19 pp249-252.
- Dash, S.K. and Koppel, L.B.(1989) Sudden destabilization of controlled chemical Processes Chemical Engineering Communications, 84 , pp 129-157.
- Koppel, L.B. (1982) Input multiplicities in nonlinear multivariable control systems AIChE Journal.28 pp935 -945.
- Koppel, L.B.(1983) Input multiplicities in process control, Chemical Engineering Education, pp58-63, & 89-92.
- Dr. Baughman and Y.Liu, "Neural Networks in Bio processing and Chemical Engineering," Academic press, 1995.
- Robert E. King, "Computational Intelligence in Control Engineering" (pg No. 153-166), Marcel Dekker Inc., NY, 1999.
- Mohamed Azlan Hussain, Pisan kittisupokorn & Wachira Daosud, "Implementation of Neural Network based inverse model control strategies on an exothermic reactor", Science Asia 27, 2001, pp. 41-50.
- Furong Gao, Fuli Wang & Mingzhong Li, "Neural controller based optimal iterative controller for non linear processes," The Canadian Journal of Chemical Engineering, Volume 78, 2000.