Direct Inverse Neural Network Control of A Continuous Stirred Tank Reactor (CSTR)

D.B.Anuradha, G.Prabhaker Reddy*, J.S.N.Murthy

Abstract— **In recent years, there has been a significant increase in the number of control system techniques that are based on nonlinear concepts. One such method is the nonlinear inverse model based control strategy. This method is dependent on the availability of the inverse of the system model. Since neural networks have the ability to model any nonlinear system including their inverses and their use in this control scheme is promising. In the present paper, direct inverse neural network control strategy for controlling the CSTR with van de vusse reaction is studied. The direct inverse NN control strategy utilizes the process inverse model as controller. For training the neural network, the process input-output data is generated by applying a pseudo random signal on a simulink model of the CSTR process. Then, the input-output data is divided into two parts for training & validation. Training is performed using the Levenberg-Marquardt method. Based on the SSE, the optimum number of hidden nodes is taken as ten. The model obtained after the training is inverse NN model, which is taken as NN based controller. The performance of proposed NN based controller is evaluated for servo and regulatory control problems through simulation studies. Through the closed loop simulation studies, it is found that neural network based direct inverse control strategy gives superior performance to PID controller for setpoint changes. To improve the performance of direct inverse NN controller for regulatory problem, the IMC structured with forward and inverse NN models are included in the closed loop system.**

***Index Terms*—Neural Network Control, CSTR, IMC structure, Van de Vusse Reaction.**

I. INTRODUCTION

Neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest. The network is usually implemented by using electronic components or is simulated in software on a digital computer. To achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as neurons or processing units. A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge & making it available for use [1] - [6]. It resembles the brain in two ways: (i) Knowledge is acquired by the network from its environment through a learning process and (ii) Inter neuron connection strengths, known as synaptic weights are used to store the acquired knowledge.

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 onto 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 its 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

Manuscript received January 7, 2009. G.Prabhaker Reddy is thankful to UGC and MHRD, Govt. of India for the financial support under TEQIP to present this work in ICCA-2009 in Hong Kong.
D.B.Anuradha is with the Dept. of Chemical Eng., University College of Technology, Osmania University,Hyderabad,-500007,India. (e-mail: anu1214@gmail.com)
G.Prabhaker Reddy is with the Dept. of Chemical Eng., University College of Technology, Osmania University, Hyderabad,-500007, India. (Phone: 91-9866619899;Fax:91-4027098472; e-mail: gpreddy_ouct@yahoo.com)
J.S.N.Murthy is with the Dept. of Chemical Eng., University College of Technology, Osmania University,Hyderabad,-500007,India. (e-mail: jsnmurthy_ou@yahoo.com)
.* Corresponding Author

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 recent years there has been a significant increase in the number of control system techniques that are based on nonlinear concepts. One such method is the nonlinear inverse model based control strategy [7] – [11]. This method is dependent on the availability of the inverse of the system model. Since neural networks have the ability to model any non linear system including their inverses and their use in control scheme is promising. Hence, In the present paper, the design and evaluation of the neural network based inverse model controller performance to a continuous stirred tank reactor with van de vusse reaction scheme is presented.

II. Description and Model of The CSTR Process

The system considered for the control studies is an isothermal continuous stirred tank reactor with the following series-parallel reaction known as Van de Vusse reaction is considered [12].The schematic diagram of CSTR is shown in Fig.1

$A \xrightarrow{K_1} B \xrightarrow{K_2} C$ and $2A \xrightarrow{K_3} D$;

Here the desired product is B. The molar rate of formation of each component is: $r_A=-k_1C_A-k_3C_A^2$, $r_B=k_1C_A-k_2C_B$ $r_c=k_2C_B$ and $r_D=(1/2)\,k_3C_A^2$

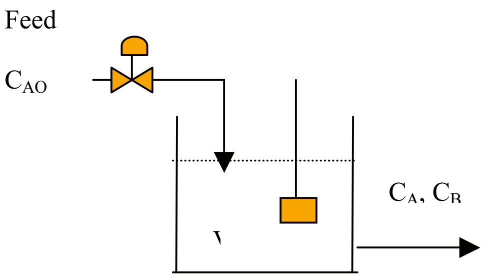

Fig.1 Schematic of the CSTR process

The model of process is obtained from the component mass balance equations for the species A & B are given by

$$dx_1/dt=-k_1*x_1-k_3*x_1*x_1+(C_{Ao}-x_1)*u \qquad (1)$$

$$dx_2/dt=k_1*x_1-k_2*x_2-x_2*u \qquad (2)$$

Where $x_1=C_A$, $x_2=C_B$, u=F/V. F=flow rate in lit/min; V is the volume of the reactor; C_A and C_B are the concentrations respectively of A&B. The parameter values considered are k_1=50 1/h, k_2=100 1/h, k_3=10 l/mol h C_{Ao}=10 mol/l. At steady state, the concentration of B is 1.117 mol/l for space velocity, u = 34.284 1/h .

III. Design of a Direct Inverse Neural (DIN) 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 with pseudo random signal (PRS) shown in Fig.2 for input signal of space velocity (u) of the CSTR system, the output response shown in Fig.3 for the concentration of B, CB is generated using the simulink model of the process with a sampling time of 0.2s for 1000 samples.

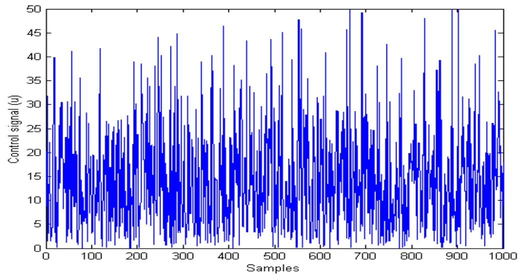

Fig. 2 Pseudo Random Signal for process input, space velocity, u to CSTR process

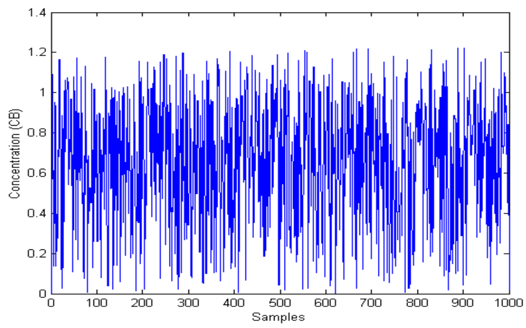

Fig. 3 Process response/ output in C_B for the PRS input, u, shown in Fig.2

Inverse Neural network model shown in Fig.4 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

$$u(t) = f\text{-}1[\,y(t+1), y(t), y(t-1), u(t-1)\ldots] \qquad (3)$$

ISBN: 978-988-17012-7-5

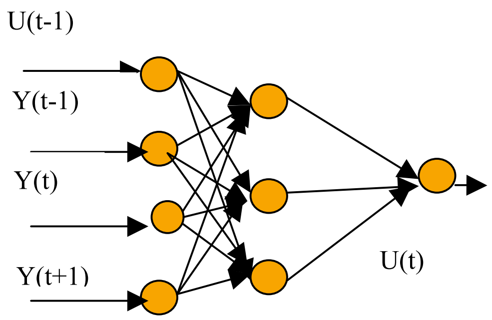

Fig. 4 Structure of Inverse neural network model

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 uN is the error eN 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 W1 contains weights from input to hidden layer & weight matrix W2 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 lambda is the regularization factor which is chosen initially as 1. Based on the SSE the regularization factor is also updated. Lambda 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 is increased and SSE is observed if it is decreasing then the nodes are increased till the value again starts decreasing. The number of nodes are chosen where it gives minimum SSE. Number of input nodes are taken as four 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.

IV. SIMULATION RESULTS & DISCUSSION

The performance of the direct inverse neural network controller for CSTR with van de vusse reaction scheme is evaluated using the closed loop block diagram shown in Fig.5.

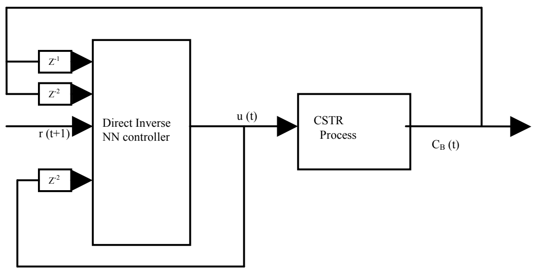

Fig. 5 Direct Inverse Neural Network Control

The closed loop results of direct inverse neural network controller are compared with the PID based controller below for servo & regulatory problems. In the present studies, the concentration of B, C_B is controlled using the space velocity, F/V as manipulated variable. The disturbance considered here is feed concentration of A i.e., C_{AO}. The normal operating conditions considered for simulation studies are : Feed concentration C_{A0}=10mol/l, space velocity,u = 34.25 1/h and concentration of desired product, B, CB=1.117 mol/l. The PID controller parameters are taken as K_c=1.87, τ_I=1.1 and τ_D =0.26 [12].

The closed loop responses of direct inverse neural network controller and PID controller for the step changes in set point is shown in Fig.6 From these result. it is found that the NN based DIC is found to be faster than PID controller. The corresponding control action is plotted in Fig.7

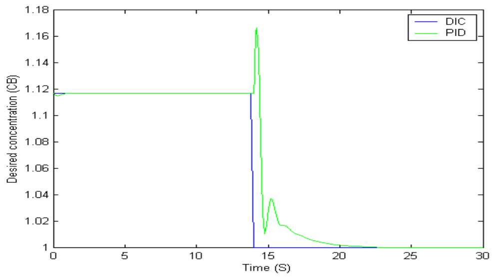

Fig.6 Closed loop response of Direct inverse NN controller & conventional PID control for set point change in CB from 1.117 to 1 mol/l.
Legend: 1- NN controller and 2-PID controller

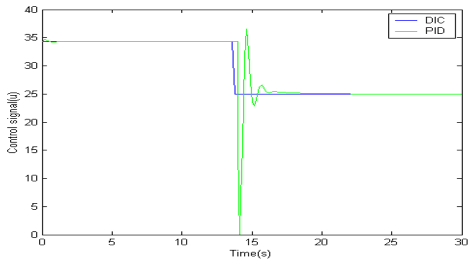

Fig. 7 Manipulated variable/controller action, u, versus time For the closed loop response shown in Fg.14.
Legend: 1- NN controller and 2-PID controller

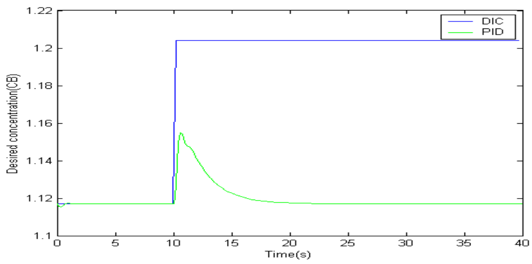

Fig. 8 Closed loop response of Direct inverse NN controller & PID controller for the disturbance in C_{A0} from 10 to 11 mol/l.
Legend: 1- NN controller and 2-PID controller

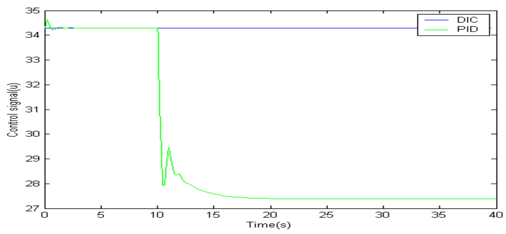

Fig. 9 Manipulated variable/control action, u, versus time For the closed loop response shown in Fg.8.
Legend: 1- NN controller and 2-PID controller

The regulatory problem is studied by giving the disturbance in the feed concentration, C_{AO}. The step disturbance is given at 10th sampling instant from 10 to 11 mol/lit. in C_{AO}. The set point is kept constant at 1.117 mol/lit. The closed loop responses of the NN based DIC for the above changes are shown in the Fig.8. The corresponding control signal action is plotted in Fig.9. From these results it is found that NN based DIC is giving offset. In order to improve the performance of the NN controller NN forward model is placed across the process as shown in Fig.10. This block diagram becomes NN based IMC structured.

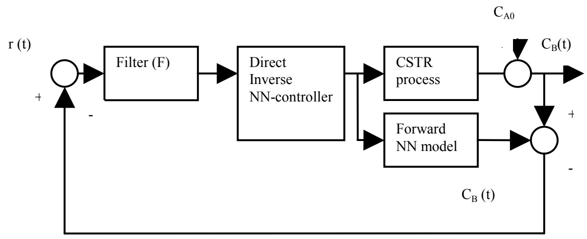

Fig.10 Closed loop block diagram of IMC structured Neural Network controller for a CSTR process.

The closed responses of the NN- IMC is shown in Fig.11. The corresponding control signal action is shown in Fig.12. It is found that NN based IMC is able to give offset free response.

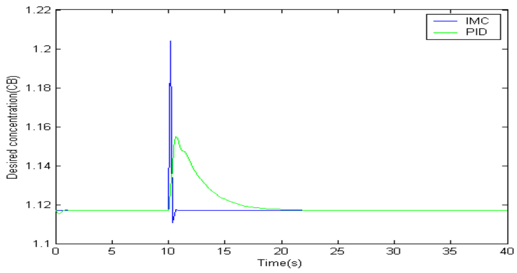

Fig. 11 Closed loop response of IMC structured NN controller & PID controller for disturbance in C_{AO} from 10 to 11mol/l. Legend: 1- NN controller and 2-PID controller

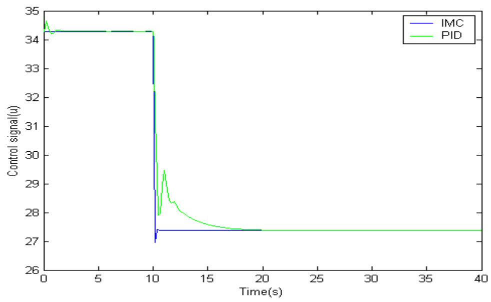

Fig. 12 Manipulated variable/controller action, u, versus tim For the closed loop response shown in Fg.11.
Legend: 1- NN controller and 2-PID controller

V. Conclusion

In the present work, the forward and inverse neural network models of the isothermal CSTR process with van de vusse reaction scheme have been developed using the simulated input – output data of the CSTR Process. The optimum structure of NN is obtained as with four input nodes, ten hidden nodes and one output node. The performance of inverse NN controller is found to be superior to conventional PID controller for set point changes. The Performance of inverse NN controller is improved for regulatory problem with IMC structured forward and inverse NN models in the closed loop. The NN controller is found to be robust i.e., its closed loop performance is remain unaltered in the presence of perturbations in process parameters.

REFERENCES

[1] Dr.Baughman and Y.Liu, ".Neural Networks in Bio processing and Chemical Engineering, " Academic press, 1995.

[2] S.Rajasekaran and G.A.Vijaya lakshmi pai, " *Neural Networks, FuzzyLogic, and Genetic Algorithms Synthesis and applications*," Prentice Hall of India Pvt.Ltd.,2004.

[3] Robert E. King, " *Computational Intelligence in Control Engineering* " (pg No. 153-166), Marcel Dekker Inc., NY, 1999.

[4] P.L.Lee, R.B.Newell & I.T.Cameron, " *Process control & Management*," Blackie Academic & professional

[5] F.G.Shinskey, " *Process control systems*, " McGraw Hill higher Education, II Rev Ed, 1979.

[6] Chin-Teng Lin & C.S.George Lee, " *Neural Fuzzy Systems*, " Prentice Hall, 1996.

[7] Mohamed Azlan Hussain, Paisan kittisupokorn & Wachira Daosud, "Implementation of Neural Network based invere model control strategies on an exothermic reactor", *Science Asia* 27, 2001, pp. 41- 50.

[8] Mohamed Azlan Hussain "Review of the applications of neural networks in chemical process control–Simulation & online implementation,"*Artificial Intelligence in Engineering*,1999 pp.55-68.

[9] J .L.Dirion, M.Cabassud, M.V.Le Lann & G.Casamatta , "Design of neural controller by inverse modeling," *Computers Chemical Engineering* Volume 19, Suppl , 1995.

[10] 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.

[11] N.Bhatt & T.J.Mac Avoy, " Use of neural nets for dynamic modeling & control of chemical process systems, " *Computers Chemical Engineering Volume,* 14, No.4/5, 1990, pp.573-583

[12] M.Chidambaram & G.P.Reddy, " Non linear IMC of systems with input multiplicities", *Indian Journal of Chemical Technology*, 1995, pp.115-118