

On the Application of Neural Network Predictive Controller For Stirred Tank Reactor

¹D. I. Lanlege, ²L.A. Nafiu, ³U.M.Gana, ⁴A.A.Falaye

¹Department of Mathematics and Computer Science, Ibrahim Badamasi Babangida University, Lapai, Niger State, Nigeria

²Department of Mathematics and Statistics, Federal University of Technology, Minna, Niger State, Nigeria

¹davidlanlege@yahoo.co.uk, ²lanconserv@yahoo.com

ABSTRACT

In this paper, we proposed a Proportional Integral Derivative (PID) Neural Network Algorithm, which is used to model and solve continuous stirred tank mixer (CSTM) problem. This hybrid algorithm is robust and converges fast without being trapped into a local minimal as it is the case with the conventional neural network. We established the characteristics equation governing the dynamics of the continuous stirred tank mixer/ reactor. A controller was formulated tested and found to be consistent. The Proportional Integral Derivative (PID) network was used to simulate typical continuous stirred tank reactor (CSTR) problems, of which predictive accuracy was found to be 96%.

Keywords: *Neural Network, proportional Integral Derivative (PID), Predictive Controller, Stirred Tank Reactor and Hybrid Algorithm*

1. INTRODUCTION

Neural network simulations appear to be a recent development [1]. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances in technology have been boosted by the use of inexpensive computer emulations [2]. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support were minimal, important advances were made by relatively few researchers. According to [3], these pioneers were able to develop convincing technology which surpassed the limitations identified. An Artificial Neural Network (ANN), according to [4], is an information processing paradigm that is inspired by the biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (Neurons) working in unison to solve specific problems. A trained neural network can be thought of as an "EXPERT" in the category of information; it has been given to analyze [5]. This expert can be used to provide projections given new situation of interest. The importance of neural networks is:

- i. Adaptive Learning: - An ability to learn how to do tasks based on the data given for training or initial experience.
- ii. Self organization: - An Artificial Neural Network can create its own organization or representation of the information. It receives during learning time.
- iii. Real time operation: - Artificial Neural Network computation may be carried out in parallel and special hardware devices are being designed and manufacture which take advantage of this capacity [6].
- iv. Fault tolerance via redundant information coding partial destruction of a network leads to the corresponding degradation of performance.

However, some network capabilities may be retained even with major network damage.

According to [7], the application of neural network gives this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problem. In fact, they have already been successfully applied in many industries. Since neural networks are best for identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- Sales forecasting.
- Industrial processing control.
- Customer research.
- Data validation.
- Risk management.
- Target marketing.

But to give some more specific examples, artificial neural network are also used in the following specific paradigms [8]. Recognition of speaker in communication Diagnosis of hepatitis Recovery of telecommunication from faulty software Interpretations of naitinceaning Chinese word undersea mining detective Texture analysis. Three dimensional object recognition hand-written word recognition and facial recognition.

2. METHOD AND MATER ALS

Proportional-Integral-Derivative (PID) controller is a common feedback loop component in industrial control system. The PID loop tries to automate what an intelligent operator with a gauge and a control knob does. The PID loop adds positive correction, removing error from the process's controllable variable (input). The "Process Variable" is also called the Process Input or "Controller Output" as shown in figure 1. The input flow rates F_1 and F_2 in figure 2 are controlled by input valves while the output flow is assumed to be through a constant size orifice. This output flow rate is assumed to vary as the square root of the static pressure head at the orifice with constant cross-sectional area of the tank this means

[11].

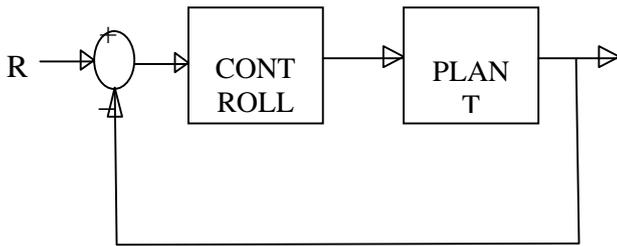


Fig 1: Output Feedback Control System

2.1 Proportional – Integral Derivative Control

$$U(S) = (K_p + \frac{K_i}{S} + K_d S) E(S) \tag{1}$$

$$\frac{K_d S^2 + K_p S + K_i}{S} E(S) \tag{2}$$

$$OLTF = \frac{K_d S^2 + K_p S + K_i}{S^3 + S^2 + S} \tag{3}$$

$$CLTF = \frac{K_d S^2 + K_p S + K_i}{S^3 + (10 + K_d)S^2 + (20 + K_p)S + K_i} \tag{4}$$

Now let's take a look at a Proportional Integral Derivative (PID) Controller the closed-loop transfer function of the given system with a Proportional Integral Derivative (PID) Controller is

$$\frac{X(S)}{F(S)} = \frac{K_D S^2 + K_p S + K_i}{S^3 + (10 + K_D)S^2 + (20 + K_p)S + K_i} \tag{5}$$

After several trial and error runs, the gain $K_p = 350$, $K_i = 300$, and $K_d = 50$ provided the desired response. To confirm, enter the following command window will get the following step response.

`Kp = 350, Ki = 300, Kd = 50, num [Kd, Kp, Ki], Den [1, 10+Kd, 20+Kp, Ki], t = 0:0.01:2, Step (num, den, t)`

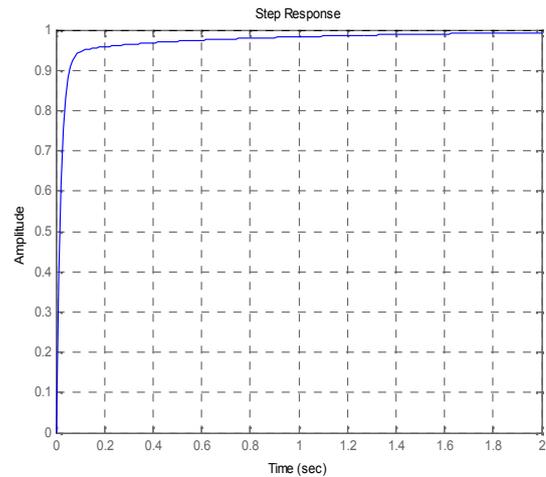


Fig 2: Closed – Loop Step: $K_p = 350$ $K_i = 5500$

Now, we have obtained the system with no overshoot, fast rise time, and no steady-state error. A proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control (K_i) will have effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response. Effects of each of the controllers K_p , K_d and K_i on a closed-loop system

2.2 The Dynamic of the Stirred Tank Mixer

The problem under consideration is a non-linear stochastic problem which to be precise is stirring tank mixer (CSTM). This has practical application in many chemical, pharmaceutical and petroleum industries as well as in environmental engineering and waste management.

The problem from [10], is illustrated in fig. 3 below. There are two input flows (with flow rates F_1 and F_2 , concentrations C_1 and C_2 respectively) going in at the top of the mixer. The two inputs are mixed in the tank to produce output with flow rate F_3 and concentration C_3 out at the bottom.

CONTROL

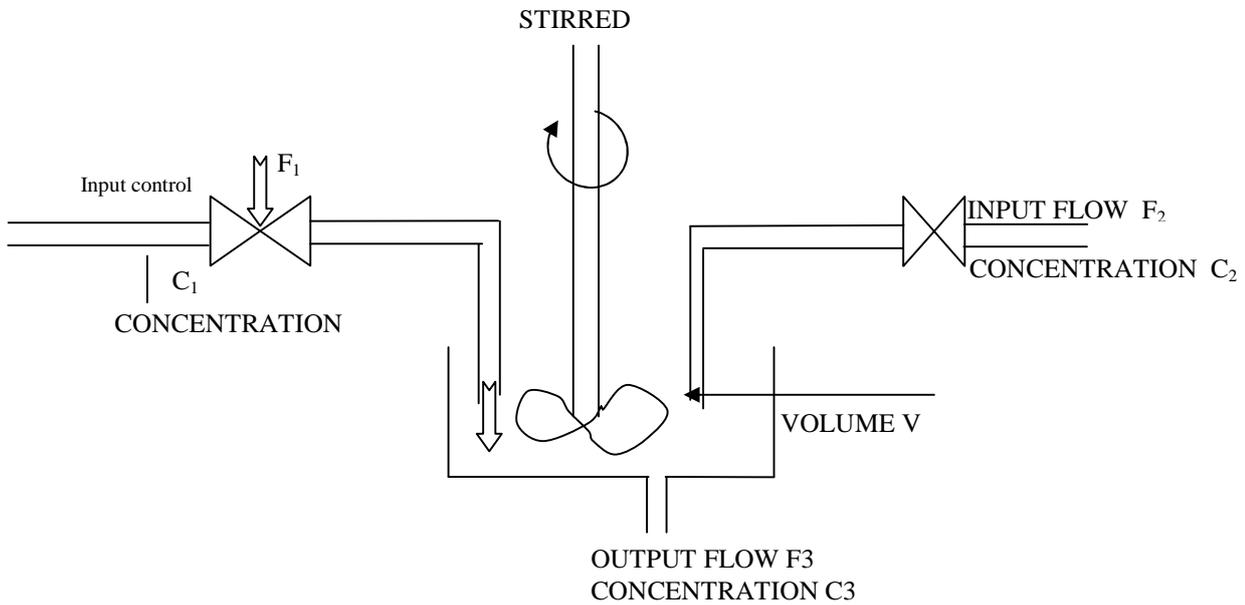


Fig 3: Stirred Tank Rector

We see that volume V in the tank changes as the difference between flow in and flow out with a few obvious manipulations.

$$\frac{dC_3}{dt} = \frac{(C_1 - C_3)F_1 + (C_2 - C_3)F_2}{V}$$

$$\frac{dV}{dt} = F_1 + F_2 - F_3 \quad (7)$$

which is the $x = F(x)$ from desired here.

The second stage of model predictive control is to make a neural network to represent the forward dynamics of the plant. The prediction error between the plant output and the neural network output is used as the neural network training signal. The process is represented by figure 1. (6)

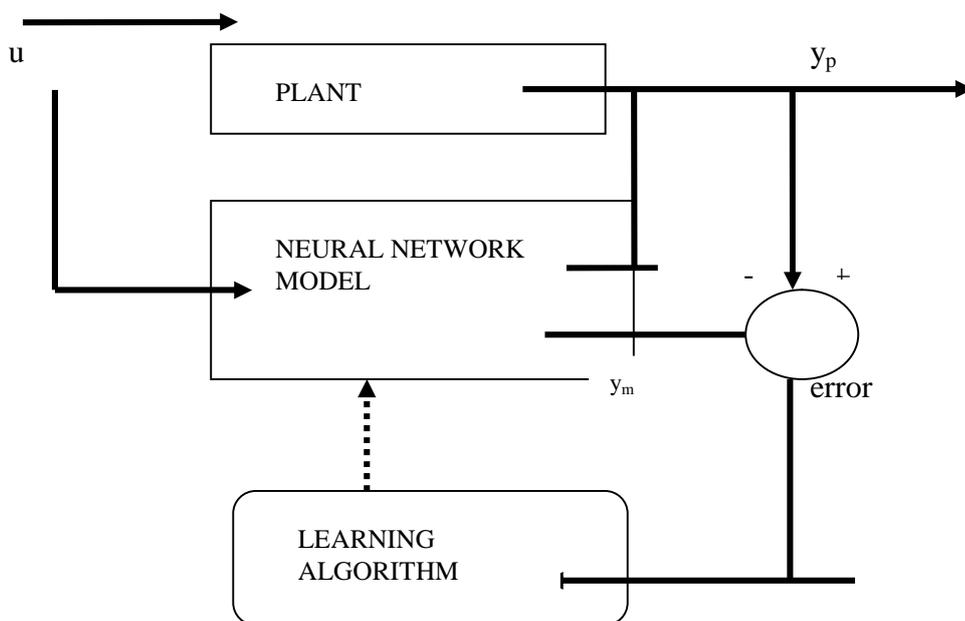


Fig 4: Neural Network Plant Model

The neural network plant model uses previous inputs and previous plant outputs to predict future values of the plant output. The structure of the neural network plant model is given in figure 5

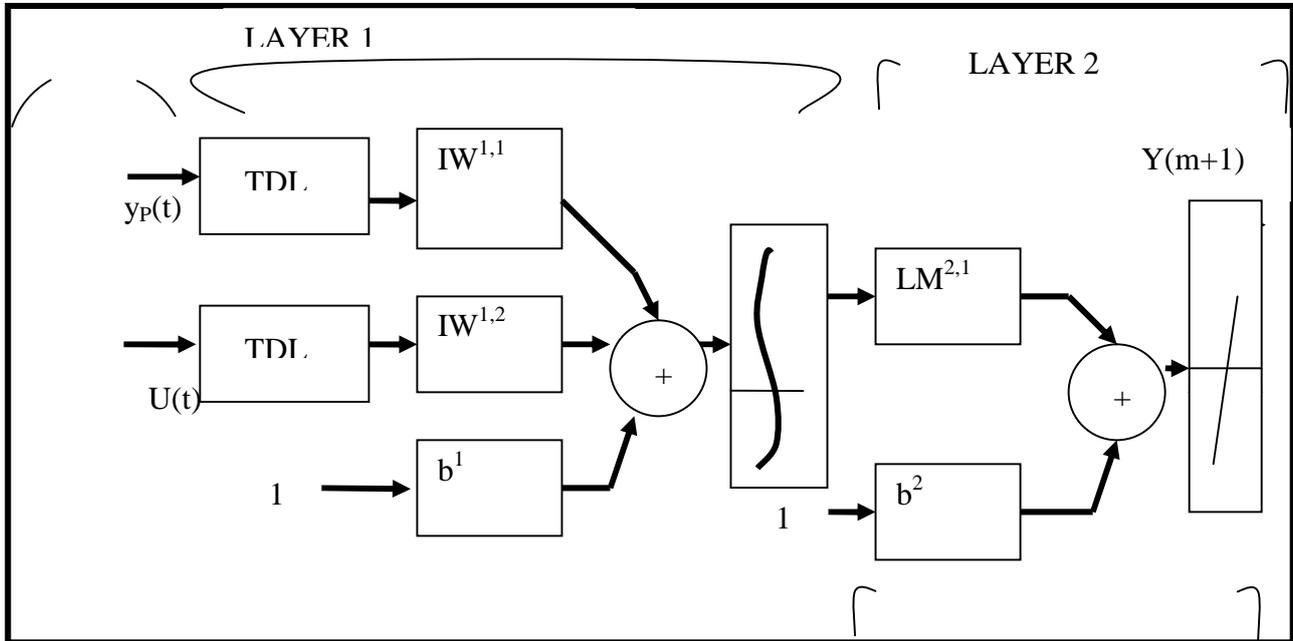


Fig 5: Neural Network

This network can be trained offline in batch mode, using data collected from the operation of the plant. Any of the training algorithms discussed in Back propagation, can be used for network training. This process is discussed in more detail later in this paper. The model predictive control method is based on the preceding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization program to determine the control signal that minimizes the following performance criterion over the specified horizon

$$\begin{aligned}
 \text{Minimize } J = & \sum_{N_1}^{N_2} (y_r(t+j) - y_m(t+j))^2 + \\
 & \dots \sum_{j=1}^{N_u} (u'(t+j-1) - u'(t+j-2))
 \end{aligned}
 \tag{8}$$

Where N_2 , N_1 and N_u define the horizons over which the tracking error and the control increments are evaluated. The u' variable is the tentative control signal, y_r is the desired response and y_m is the network model response. The ... value determines the contribution that the sum of the squares of the control increments has on the performance index. The following block diagram illustrates the model predictive control process. The controller consists of the neural network plant model and the optimization block. The optimization block

determines the values of u' that minimize J , and then the optimal u is input to the plant. The controller block has been implemented in Simulink, is shown in figure 6

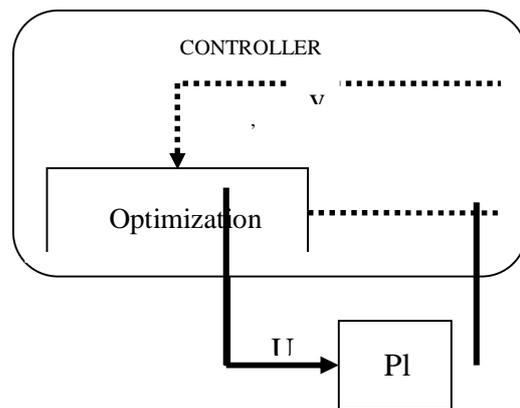


Fig 6: Neural Network Predictive System Controller Block

Illustration: This section demonstrates how the NN Predictive Controller block is used. The first step is to copy the NN Predictive Controller block from the Neural Network Toolbox block set to your model window. This demo uses a catalytic Continuous Stirred Tank Reactor (CSTR). A diagram of this process is shown in the following figure 7.

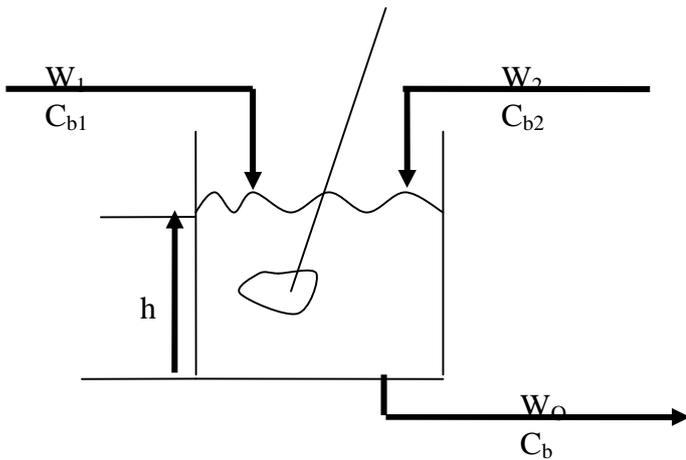


Fig 7: A Stirring Tank reactor

where $h(t)$ is the liquid level, $C_b(t)$ is the product concentration at the output of the process, $w_1(t)$ is the flow rate of the concentrated feed C_{b1} , and $w_2(t)$ is the flow rate of the diluted feed C_{b2} . The input concentrations are set to $C_{b1} = 24.9$ and $C_{b2} = 0.1$. The constants associated with the rate of consumption are $k_1 = 1$ and $k_2 = 1$.

The objective of the controller is to maintain the product concentration by adjusting the flow $w_2(t)$. To simplify the demonstration, we set $w_1(t) = 0.1$. The level of the tank $h(t)$ is not controlled for this experiment. To run this demo, follow these steps

The dynamic model of the system is:

$$\frac{dh(t)}{dt} = w_1(t) + w_2(t) - 0.2\sqrt{h(t)} \quad (7)$$

$$\frac{dC_b(t)}{dt} = (C_{b1}(t) - C_{b2}(t))\frac{w_1(t)}{h(t)} + (C_{b2} - C_b(t))\frac{w_2(t)}{h(t)} - \frac{k_1 C_b(t)}{(1 + k_2 C_b(t))^2} \quad (8)$$

2.3 Illustration: (Running the Predictive Controller)

To run this demo, follow these steps.

- i. Start MATLAB
- ii. Run the demo model by typing `predcstr` in the MATLAB® command window. This command starts Simulink and creates the following model window. The NN Predictive Controller block has already been placed in the model as show in figure 5

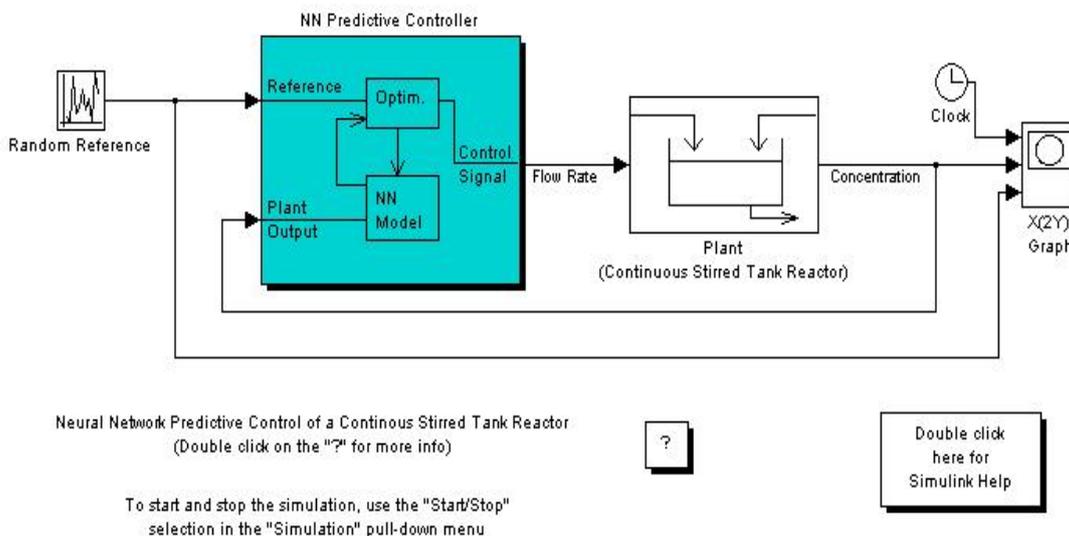


Fig 8: Predictive Controller block

- iii. Double-click the NN Predictive Controller block. This brings up the window for designing the model predictive controller as shown in figure 9. This window enables us to change the controller horizons N_2 and N_u (N_1 is fixed at 1). The weighting parameter \dots , described earlier, is also defined in this window. The parameter Γ is

used to control the optimization. It determines how much reduction in performance is required for a successful optimization step. One can select which linear minimization routine is used by the optimization algorithm, and decide how many iterations of the optimization algorithm are performed at each sample time.

<http://www.ejournalofscience.org>

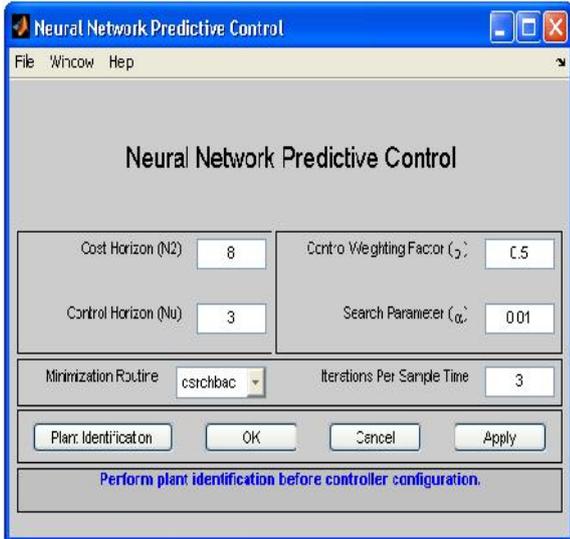


Fig 9: Neural Network Predictive Control

iv. Select Plant Identification. This opens the window figure 10. The neural network plant model must be developed before the controller is used. The plant model predicts future plant outputs. The optimization algorithm uses these predictions to determine the control inputs that optimize future performance. The plant model neural network has one hidden layer, as shown earlier. The size of that layer, the number of delayed inputs and delayed outputs, and the training function are selected in this window. One can select any of the training functions described in Back propagation, to train the neural network plant model.

v. Select the Generate Training Data button. The program generates training data by applying a series of random step inputs to the Simulink plant model. The potential training data is then displayed as given in figure 8.

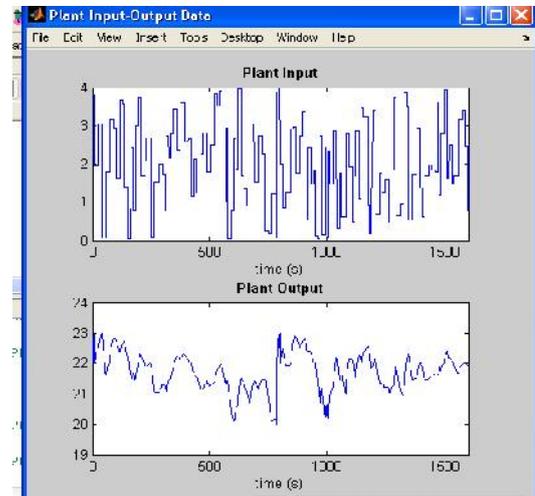


Fig 11: Plant Input and Output Data

vi. Select Accept Data, and then select Train Network from the Plant Identification window. Plant model training begins. The training proceeds according to the selected training algorithm (trainlm in this case). This is a straightforward application of batch training, as described in Back propagation. After the training is complete, the response of the resulting plant model is displayed. There are also separate plots for validation and testing data, if they exist. One can then continue training with the same data set by selecting Train Network again, erase Generated Data and generate a new data set, or accept the current plant model and begin simulating the closed loop system. For this demonstration, begin the simulation, as shown in figure 11 and 12.

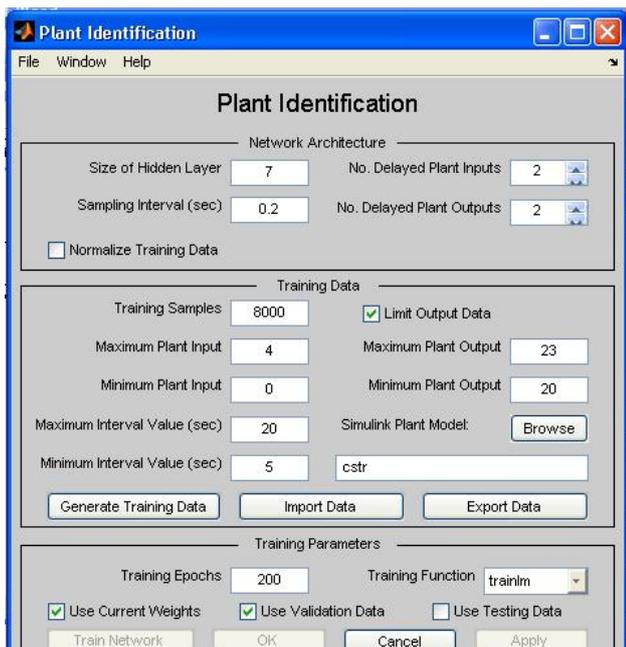


Fig 10: Plant identification

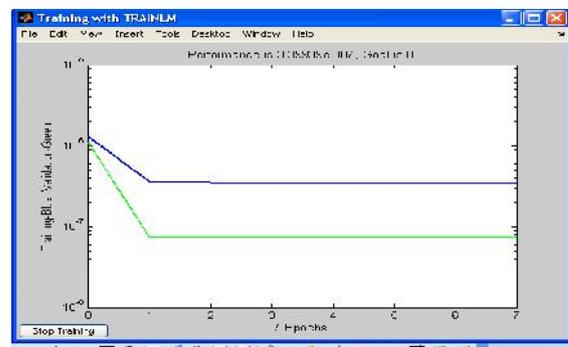


Fig 12: Training with Trainlm

<http://www.ejournalofscience.org>

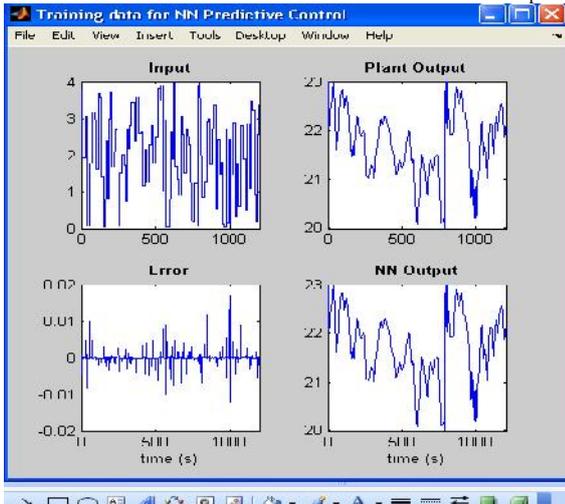


Fig 13: Training Data for NN Predictive Control

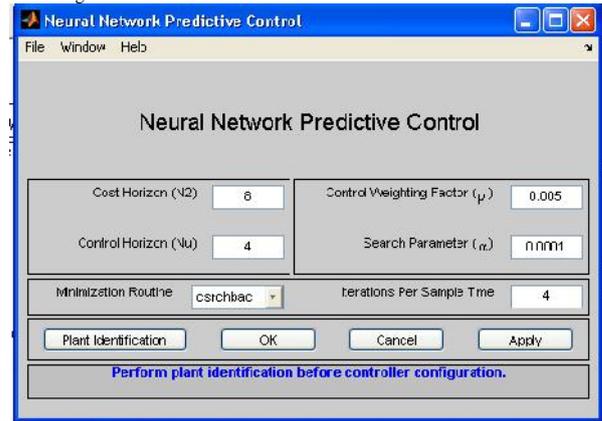


Fig 15: Neural Network Predictive Control

- vii. Select OK in the Plant Identification window. This loads the trained neural network plant model into the NN Predictive Controller block.
- viii. Select OK in the Neural Network Predictive Control window. This loads the controller parameters into the NN Predictive Controller block.
- ix. Return to the Simulink model and start the simulation by choosing the Start command from the Simulation menu. As the simulation runs, the plant output and the reference signal are displayed, as in figure 13.

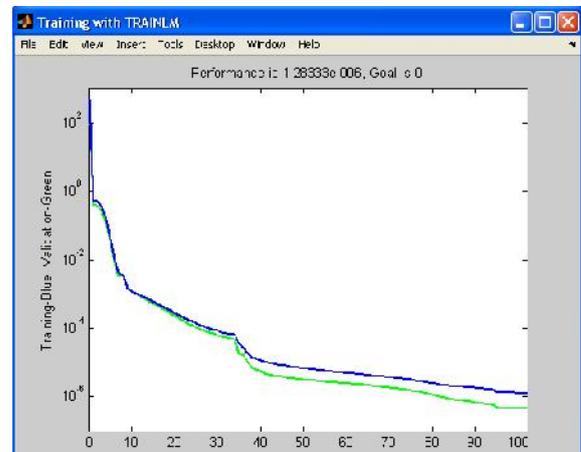


Fig 16: Training the Identification Controller

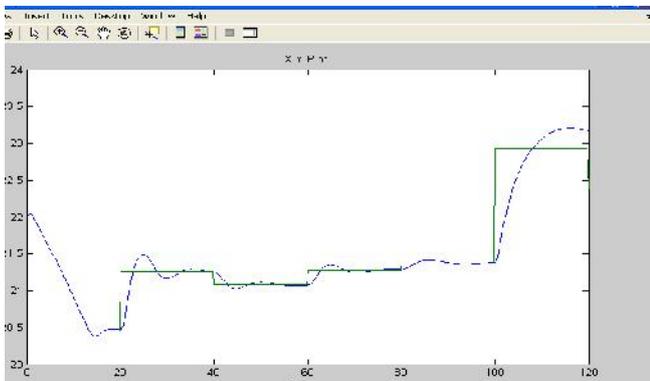


Fig 14: The Plant output and Reference signal displayed

2.4 (NN Predictive Controller)

In a sequential procedure as in example 4.1, we carry on the process of simulating for the Plant output and reference signal in this order. This second example is obtained by changing the inputs data as shown in the figures below:

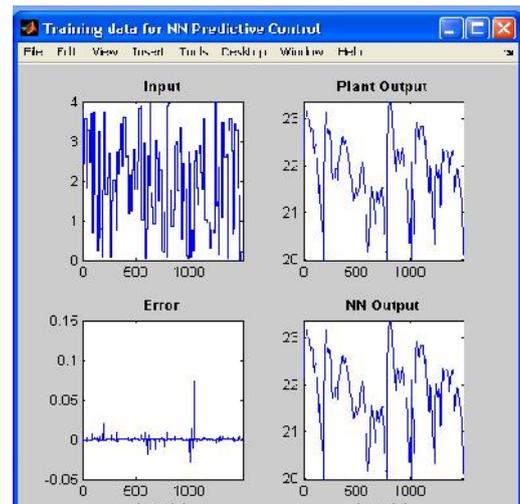


Fig 17: Training Data Inputs and Output plant, Error and NN Output signals

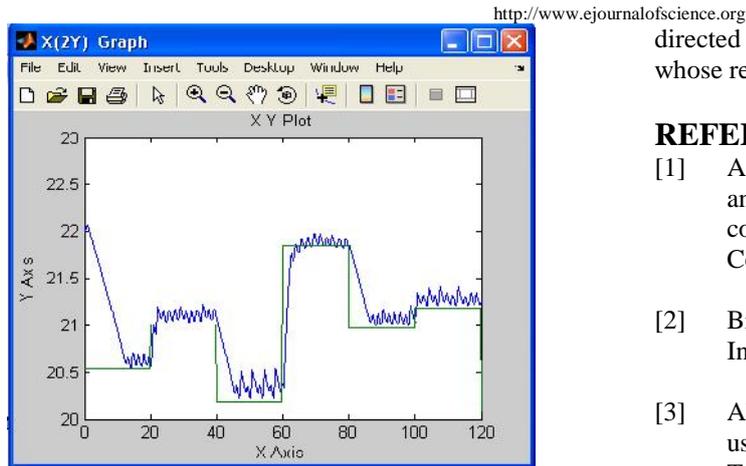


Fig18: The Plant output and Reference signal displayed

3. D SCUSS ON OF RESULTS

We have successfully modeled the mechanism governing the functionality of the brain. To achieve this goal, biological neurons were described in sufficient detail to permit meaningful modeling. The brain has been viewed as a network of billions of simple computing devices, each operating on its own set of inputs and transmitting its output to specific sets of other computing devices in the network.

The highly complex interconnections of neurons produced the computation power of the brain. Artificial neural network mimicking the computation richness of the brain implanting hybrid Proportional Integral Derivative Algorithm (maintaining) the volume of the mixture in the tank, checking unnecessarily high inlet flow rate and ensuring the stability of the system for the continuous stirred tank reactor (CSTR) problem, which is nonlinear and dynamic in nature. The results of this work strengthen the fact that neural networks can be used for fault detection and diagnosis purposes. This paper also shows that this Proportional Integral Derivative (PID) controller provides an alternative attractive method for solving the Continuous Stirred Tank Reactor (CSTR) problems.

4. RECOMMENDAT ONS

Since neural network model offer very promising approach to building truly intelligent. Systems, which can provide good optimal solutions for the control problems it is therefore recommended that effort should be made to explore the full potential of this new computing technique to industrial problems and further research should be

directed towards continuous stirred tanks reactor problem whose reaction is of higher order

REFERENCES

- [1] Aarts, E. and Korst, J. (1989). Simulated Annealing and Boltmann Machines: A Stochastic Approach to combinatorial Optimization and Neural Computing. John Wiley and Son: New York.
- [2] Branko, S. (1991). Neural and Intelligent System Integration. John Wiley & Son Inc: New York.
- [3] Alvager, T., Smith, T.J. and Vijia, F. (1994). The use of Artificial Neural Networks in Biomedical Technologies: An Introduction; Biomedical Instrumentation and Technology, Vol. 28, No.4, Pp315-322.
- [4] Bose, N.K. and Liang, P. (1996). Neural Network Fundamentals with Graph: Algorithms and Applications. Mc graw- Hill Inc: New York.
- [5] Muhammad, A. (2001). "Computerized Patient Monitoring System for ICU patient Care", Undergraduate Final Year Project, Sir Syed University of Engineering and Technology, Karachi, Pakistan.
- [6] Szu, H. and Hwang, W. L. (2003). "Self-Supervised Back propagation in Stem Cells", Newsletter of the IEEE Neural Networks Society vol. 1, No. 3, Pp5-10.
- [7] Goita, K. (1994). "Literature Review of Artificial Neural Networks and Knowledge Based Systems for Image Analysis and Interpretation of Data in Remote Sensing", Canadian Journal of Electrical and Computer Engineering, Vol.19, No.2, Pp53-61.
- [8] Cooper, B.S. (1994). "Selected Application of Neural Networks in Telecommunication Systems" Australian Telecommunication Research; Vol. 28, No. 2, Pp9-20
- [9] Chua, L.O. and Yang, L. (1988). "Cellular Neural Networks: Theory". IEEE Trans. Circuit and Systems, Vol. 35, Pp1257-1272.
- [10] Hasdorff, L.(1996) Gradient Optimization and Non-Linear Control. John Wiley & Sons, New York