Comparison of Artificial Neural Network Controller and PID Controller in on Line of Real Time Industrial Temperature Process Control System

[1] Abdulgani Albagul, [2] Hafed Efheij, [3] Bileid Abdulsalam
[1,3]Engineering and Information Technology Research Center, Baniwalid, Libya
[2]College of Electronic Technology, Tripoli, Libya
Email: [1]albagoul@yahoo.com, [2]hafedrs@gmail.com, [3]abuahmedasdira051@gmail.com

Abstract— The conventional PID (proportional-integral- derivative) controller is widely applied to industrial automation and process control field because its structure is sample and its robust is well, but it do not work well for nonlinear system, time-delayed linear system and time varying system. This paper provides a new style of PID controller that is based on artificial neural network and evolutionary algorithm according to the conventional one's mathematical formula. Artificial Neural Network is an effective tool for highly nonlinear system. With the advent of high-speed computer system, there is more increased interest in the study of nonlinear system. Neuro control algorithm is mostly implemented for the application to robotic systems and also some development has occurred in process control systems. Process Control systems are often nonlinear and difficult to control accurately. Their dynamic models are more difficult to derive than those used in aerospace or robotic control, and they tend to change in an unpredictable way. This paper gives an example where a multilayered feed forward back propagation neural network is trained offline to perform as a controller for a temperature control system with no a priori knowledge regarding its dynamics. The inverse dynamics model is developed by applying a variety of input vectors to the neural network. The performance of neural network based on these input vectors is observed by configuring it directly to control the process. This paper compared the performance of PID controller with ANN based upon Set point change, Effect of load disturbances and Processes with variable dead time. The result shows that ANN outperforms the PID controller.

Keywords: Process control, Neural network (NN), PID controller, Temperature control, System modelling.

I. INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way 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 large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like peoples, learning by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustment to the synaptic connections that exist between the neurons. This is true ANN as well. It offers very large capabilities concerning complex system modeling, prediction, control and performance [1, 2, 3], However, the tuning of the PID control systems is not always easy, because of its simple control structure for wide class of process characteristics. While they are applicable to many control. The artificial neural network has the ability of learning and function approximation. In addition, the artificial neural network learning processes are independent of human intervention and expert experiences. For such situations, many studies use ANN to approximate PID formula to realize ANN controller. But the learning method of ANN usually adopts some traditional algorithm, including the delta rule, the steepest descent methods, Boltzman's algorithm, the back-propagation learning algorithm, the standard version of genetic algorithm [4,5,6], etc. These traditional learning methods of ANN exists some deficiency including such as the problem of the slow speed of convergence, local minima, and the large amount of computation of network, etc, which lead to ANN controller is difficult to use actually [7,8]. In this paper, a new ANN controller which is based on NN is proposed. Here, artificial neural network is used to approximate PID formula and train the weights of ANN. The simulation proves this controller can get better control effect, and it is easily realized and the less amount of computation.

PID control systems are widely used as a basic control technology for industrial control systems today, due to its well-known simple PID control structure. However, the tuning of the PID control systems is not always easy, because of its simple control structure for wide class of process characteristics. While they are applicable to many control problems, they can perform poorly in some applications. Highly nonlinear system control with constrained manipulated variable can be mentioned as an example. Following are the limitations of the PID controller [9].

- PID has the overshoot and undershoots in the output of temperature controlled system.
- PID gives late response and takes less number of iterations for derived period of time.
- PID not stables over the entire controlling process, it deviates the output slightly.
- It has to be retuned if the load disturbances occur. It has poor recovery rate.

II. SYSTEM BLOCK DIAGRAM

Figure 1 shows the system block diagram. It consists of following main blocks: Arduino, Temperature system LabVOLT - Model 3504, Signal conditioning circuit, TRIAC, Computer. PID controllers are widely used in industry these days due to their useful properties such as simple tuning or robustness. The point is to string together convenient qualities of conventional PID control and progressive techniques based on Artificial Intelligence. Proposed control method should deal with even highly nonlinear systems [8].

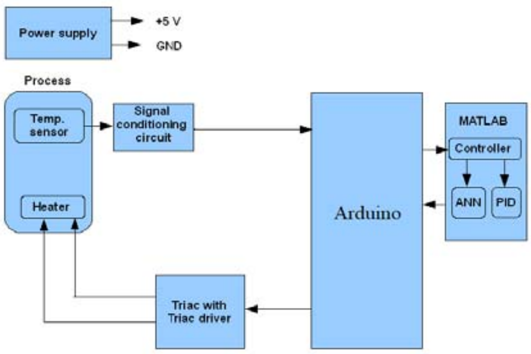

Figure 1: Block diagram of system.

- Temperature system LabVOLT - Model 3504: The LAB-VOLT Temperature Process shown in Figure 2, mainly consists of a 20-200 degree C (70-400 degree F) oven with built-in capillary-bulb temperature switch (on/off controller), thermostat, air cooling injector, adjustable damper, and overheating protection. The process instrumentation includes a capillary-bulb thermometer mounted on the side of the oven, as well as an RTD temperature transmitter and a J-type thermocouple temperature transmitter with electrical connections terminated by banana jacks on the main control panel. Control of the oven temperature can be achieved either manually by adjusting the thermostat and observing the oven temperature on the thermometer (on-off control), or remotely (PID control) by varying the amount of electrical power applied by a TRIAC driver to the heating element of the oven, using a 4-20 mA signal. The air cooling injector establishes a flow of air into the oven, thereby creating a cooling load on the process. The air pressure applied to this injector can be varied, using a pressure regulator and a needle valve, in order to change the process load. The oven damper can be used to change the process load and create disturbances.

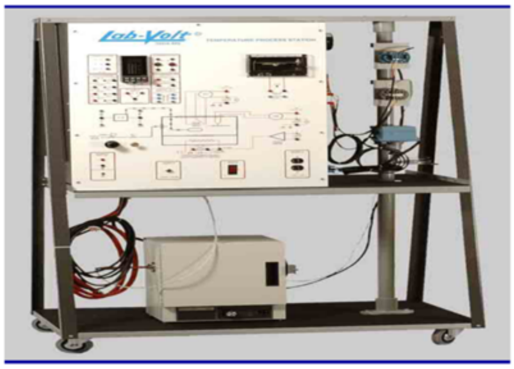

Figure 2: Block diagram of system.

The Lab-volt temperature Process can be represented by the loop diagram shown in Figure3. The Temperature Process workstation consists of a 20-200-degree Celsius (70-400-degree Fahrenheit) oven operated manually as an on-off process using a 24 Vdc relay or proportionally controlled by a TRIAC driver with 4-20 mA input. The oven is modified with an air cooling injector and adjustable damper for load and process disturbances, SYSTEM VOLTAGES 120, 220, 240 V - 50/60 Hz. The unit features a pipe-mounted on-off capillary bulb temperature controller with two sets of contacts terminated at banana jacks on the main control panel and a toggle switch that changes control from the triac driver to an NC 24 VDC relay for on-off control [10].

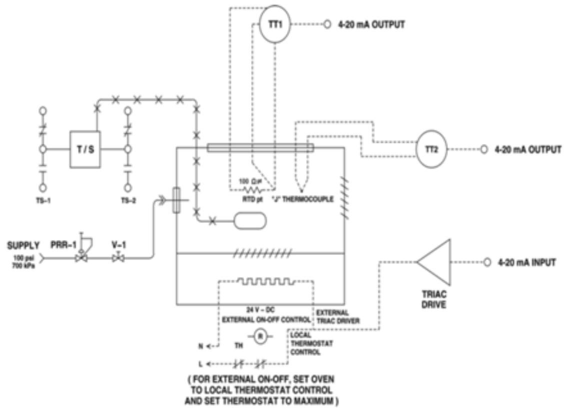

Figure 3: Lab-volt temperature process schematic diagram.

- Signal Conditioning Circuit The output of temperature sensor has to be modified so that it becomes usable and satisfactory to drive next stage. The analog output from the signal conditioning block needs to be converted to digital form so that it can be given to the Arduino.

- Arduino: It is heart the system which controls all necessary actions of. Arduino measures the temperature from water tank and send to PC. Also it controls the heater actions.
- PC: In PC, algorithm of PID and ANN is written in MATLAB. Here comparison between the outputs of PID and ANN is done [13]. The output of PC is then given to Arduino which generates Pulse Width Modulation (PWM) to control heater actions.
- Opt isolators triac driver output: The opt isolator is an integrated circuit that is specifically design to connect low voltage DC controls to high voltage AC triac. The opto in the triac output board is a 6 pin chip, but only 4 pins are used: 2 for the DC in and 2 for the AC out. Internally opto has three major sections: a LED input, a zero crossing detector and a triac driver output.

III. THE WORKING PRINCIPLE OF THE SYSTEM MODEL

The process considered here is one of the most widely used processes in the process industry, a temperature control system. Figure 4, shows the flowchart of the working principles based on a sequence of procedures to control the process.

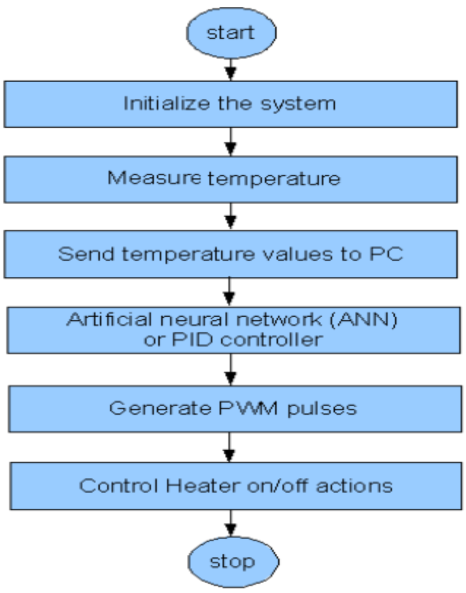

Figure 4: The flowchart of the system

The station is heated and measured temperature is given to Artificial Neural Network controller and PID controller. Desired temperature is provided to both controllers for comparison with measured temperature. Depending on magnitude of error, PWM signal is generated using Arduino. As PWM changes, the ratio ON-time to OFF-time is varied. This is continued till the desired temperature is achieved. Based on the observations, the neural network that gives best performance is then compared to a conventional feedback controller, namely, a proportional-plus-integral-plus derivative controller (PID), in controlling the same process. Direct output comparisons are made with respect to set-point changes, effect of load disturbances, and variable dead time. The comparison shows that ANN controller out-performs PID in the extreme range of non-linearity.

IV. ZIEGLER-NICHOLS RULES FOR TUNNING PID CCONTROLLER

Interestingly, more than half of industrial controllers used today use PID control schemes or modified PID. Analog PID controllers are mainly hydraulic, pneumatic, electronic, electrical or combinations thereof. Today, many of these are transformed into digital form using microprocessors. It may indicate that a PID controller responds to the equation (1).

$$u(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)\,\partial t + K_p T_d \frac{\partial e(t)}{\partial t} \qquad (1)$$

Where e(t) is the error signal, u(t) is the control input of the process. Kp is the proportional gain, Ti is the integral time constant and Td is the derivative time constant. In the frequency domain, the PID controller can be written as in equation (2).

$$U(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)E(s) \qquad (2)$$

Ziegler-Nichols method that determining the values of the proportional gain integral time and derivative time based on the transient response characteristics of a plant under study. These parameters of the PID controllers or tuning of PID controllers can be determined by making the related experiments on the plant. In this method, the proportional control action only will be used to achieve an oscillatory output response as shown in Figure 5.

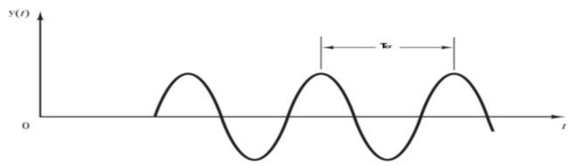

Figure 5: The oscillatory response with a period of tcr

Ziegler and Nichols suggested that set the values of the parameters Kp , Ti , and Td according to the formula shown in Table 1.

Table.1 Ziegler–Nicholas tuning rule based on critical gain Kcr and critical period Tcr

Type of Controller	Kp	Ti	Td
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}T_{cr}$	0

PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.125T_{cr}$

Notice that the PID controller tuned by this method of Ziegler–Nichols rules gives equations (5)

$$U(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) E(s) \quad (3)$$

$$U(s) = 0.6K_{cr}\left(1 + \frac{1}{0.5T_{cr}s} + 0.125T_{cr}s\right) E(s) \quad (4)$$

$$\frac{U(s)}{E(s)} = 0.075K_{cr}T_{cr}\frac{\left(s+\frac{4}{T_{cr}}\right)}{s} \quad (5)$$

And there is another way to find Kcr and Tcr if the mathematical model of the system is known (as e.g : transfer function), by using the root-locus method to find the critical value Kcr and the frequency of the sustained oscillations ωcr , where 2π/ωcr = Tcr. These values can be found from the crossing points of the root-locus with jω axis, but if the root-locus do not cross the jω axis, this method cannot be used) [11].

V. NEURAL NETWORK CONTROL PROGRAMMING

The tools used from neural network toolbox are in the following sequence:
Define neural network structure, this includes, number of inputs, number of outputs, the number of hidden layers, the type of activation function used in each stage. As an example Here a three-layer feed-forward network is created with a one-element input ranging from –1.5 to 4.16, hidden logsig neurons (n), hidden logsig neurons (m) and one purelin output neuron.
net = newff([1.5 4.16],[n m 1],{'logsig' 'logsig' 'purelin'});
Then train the neural network for up to epochs (ep1) to satisfy an error goal of (e1) and then stop the training, this is done as follows: net.trainParam.epochs = ep1;
Where
(ep1) is very large integer number.
(e1) is very small float number.
net.trainParam.goal = e1; % Error limit is e1
Then begin the training of input (in1) and related output (o1) data as follows: net = train(net, in1, o1); % in1 is input and o1 is output
After completing this training, the trained neural network can be used as the direct controller to the Lab-volt Temperature Process.
The developing of neural network using MATLAB toolbox is more efficient than the method that is done without using MATLAB toolbox.
The development of neural network will depend on the early stages of developing on PID controller, and will use the step response of Lab-volt temperature Process as a reference model to develop methods that can be used to merge PID controller and neural network. The overall block diagram of neural network controller is shown in Figure 7.

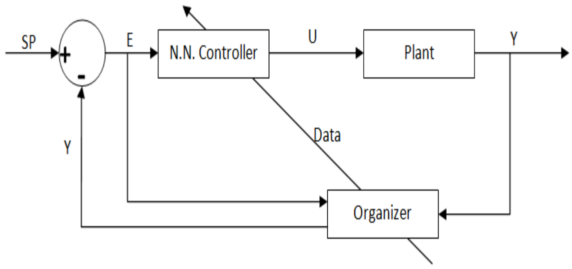

Figure 7: The block diagram of neural network controller

The flowchart of neural network control operation as shown in Figure 8.

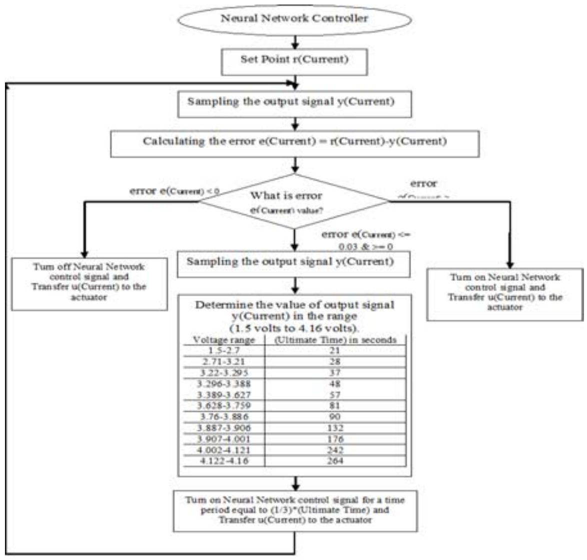

Figure 8 : The flowchart of neural network control operation

VI. EXPERIMENTAL RESULTS

In this experimental result will be obtained the step response of Lab-volt Temperature Process using Arduino, by applying the following procedure [12]:

1. Calibrate the thermocouple transmitter 0-200.
2. Set up and connect the equipment with SDAQ.
3. Set the control signal from MATLAB GUI to 4 mA output.
4. Switch on the AC mains. Open the oven damper and apply a small cooling load to the oven (open V1 approximately 1/4 turn).
5. Wait for the temperature to stabilize and record the temperature.
6. Start the recorder on slow speed. Verify that the signal has stabilized.
7. Increase the calibrator output from 4 to 12 mA, and record sample number.
8. When the temperature has stabilized as shown in Figure 9, save step response data in computer to use

for analysis, stop the recorder and switch off the power.

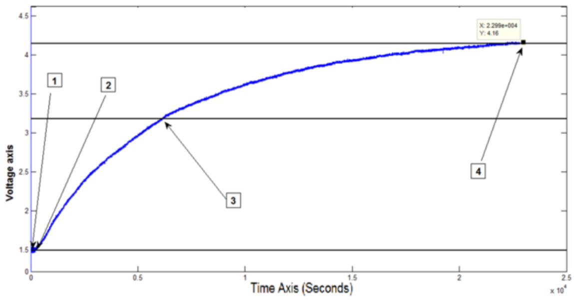

Figure 9: Step response of the process using sdaq

10. From Figure 9, perform the required calculations as follows [13]:

From Figure 9 can obtain the following information.

- Point (1) at sample 31 and time of 6.82 Sec.
- Point (2) at sample 209 and time of 46 Sec.
- Point (3) at sample 6132 and time of 1349 Sec.
- Point (4) approximately stabilised at 4.16 Volts.

Where the sample time = 0.220 Sec.

Then:

Temp. Start (TO) → 1.5 Volts ≡ 19 OC

Temp. Finish (Tf) → 4.13 Volts ≡ 118.5 OC

Temp. Change Tc = Tf - TO → 99.5 OC

Process Gain

- The Temperature Change (Tc) to a percent of transmitter span:

$$\frac{99.5}{150} * 100\% = 66.3\%$$

- The input change as a percent of span:

$$\frac{12mA - 6mA}{20mA - 4mA} * 100\% = 37.5\%$$

Then

$$\text{Process Gain} = \frac{\text{Output change (in \% of span)}}{\text{Input change (in \% of span)}} = \frac{66.3\%}{37.5\%} = 1.768$$

Process Dead Time

- Process Dead Time (τd) = time difference between input step change (Sample 31) and point where temperature started to increase (Sample 209) from Figure 9.

$$\tau d = 46 \text{ Sec} - 6.82 \text{ Sec} = 39.18 \text{ Sec}$$

Process time constant

Process time constant (τ) = time taken to reach 63.2% of the final steady state value this displayed at point (3) in Figure 9.

$$\tau = 1349 \text{ Sec} - 46 \text{ Sec} = 1303 \text{ sec}$$

$$\frac{\tau}{\tau d} = \frac{1303}{39.18} = 33.2$$

This value is greater than 10 which indicates that the process is easy to be controlled.

Then the final transfer function of the system is:

$$G(s) = \frac{1.768}{1303s + 1} \times e^{-39.18s}$$

The response of PID controller of the kit using FOXBORO controller is shown in Figure 9, where the Figure shows the relation between time samples and temperature.

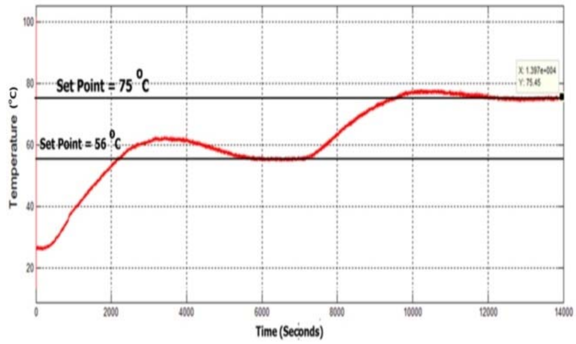

Figure 10: PID controller response using FOXBORO controller

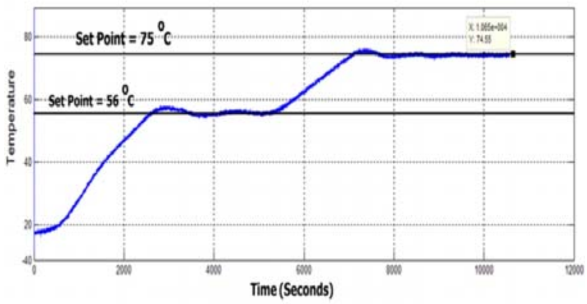

Figure 11: PID controller response using MATLAB-Arduino controller

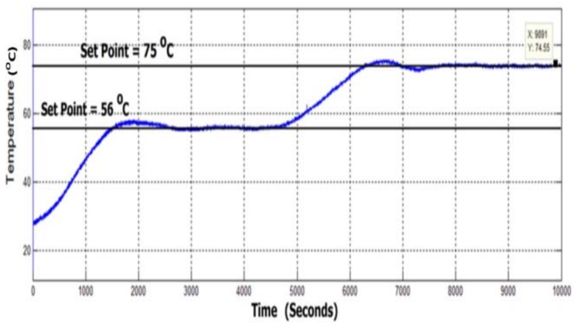

Figure 12: neural network controller response using MATLAB-Arduino controller

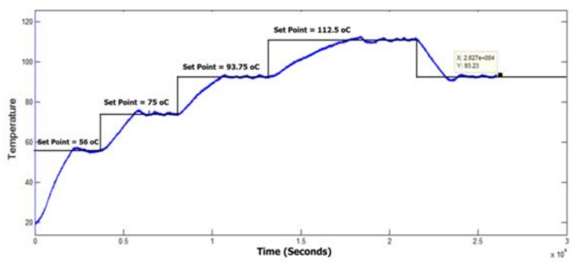

Figure 13: Neural network controller response using matlab-sdaq controller. for multi step input (2.5 volts, 3 volts, 3.5 volts, 4 volts and 3.5 volts).

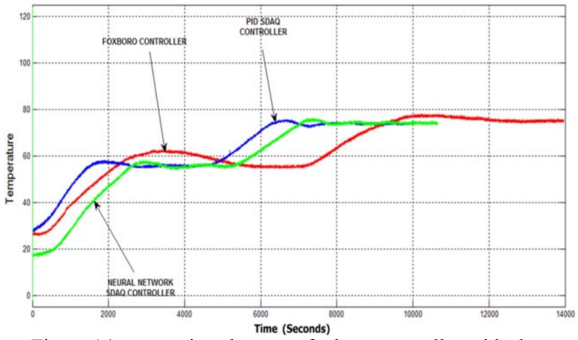

Figure 14: comparison between foxboro controller, pid sdaq controller and neural network sdaq controller

- Observations for PID controller
  1. Response time to reach towards set point is more.
  2. Output changes slightly all over the process from set point.
  3. No. of iterations to control the temperature at set point within dead time are less.
- Observations for neural network controller
  1. Response time to reach towards set point is less.
  2. Output not changes all over the process from set point.
  3. No. of iterations to control the temperature at set point

Table 2. Comparison of PID controller & AAN controller.

Feature	PID	ANN
Set point change	-No generalization property. -has to be returned and depenf upon loop respones	- very good generalization property. -need to be trained once
Effect of load	-Poor rate of recovery. Has to be returned	-very fasr recovery - adapts quickly to change in inputs.
Dead time	No improvements with time as output oscillates adversely	When retrained, performance improved with time

CONCLUSION

Neural network is studied along with its various network topologies and learning rules. feed-forward back propagation network is used in this system. A neural network controller has implemented on a real time temperature control system without the use of knowledge regarding its dynamics. The self learning ability based on its input vector has been observed. The performance of neural network controller is compared with the performance of PID controller to evaluate the system performance. It can be seen that the neural network controller totally overcomes the PID controller regarding the set point changes, effect of load disturbances, processes with variable dead time.

REFERENCES

[1] M. Azizur Rahman, *Fellow, IEEE,* and M. Ashraful Hoque, “On-Line Self-Tuning ANN-Based Speed Control of a PM DC Motor”, IEEE/ASME Transactions On Mechatronics, Vol. 2, No. 3, September 1997.

[2] Aleksandar M. Stankovi C and Andrija T. Sari6, “An Integrative Approach to Transient Power System Analysis with Standard and ANN-Based Dynamic Models”, 2003 IEEE Bologna PowerTech Conference, June 23-26, Bologna, Italy.

[3] Pravat K.Singh and Pankaj Rai, “An ANN Based X-PC Target Controller for Speed Control of Permanent.

[4] Magnet Brushless DC Motor”, Proceedings of the 2005 IEEE Conference on Control Applications Toronto, Canada, August 28-31, 2005.

[5] T. Back,H.P. Schwefel. “An overviewof evolutionary algorithms for parameter optimization,” Evol. Comput., pp. 1–23,1993.

[6] N. Dodd, “Optimization of network structure using genetic techniques,” in Proceedings of the 1990 International Joint Conference on Neural Networks, 1990.

[7] P. Bartlett, T. Downs, “Training a Neural Network Using a Genetic Algorithm,” Tech. Report, Department of Electrical Engineering, University of Queensland, 1990.

[8] G.D. Magoulas, V.P. Plagiannakos, M.N. Vrahatis, “Neural network-based colonoscopic diagnosis using online learning and differential evolution,” Appl. Soft Comput., vol. 4, pp. 357–367, 2004.

[9] M. A. Hosen, M. A. Hussain and F. S. Mjalli, “Control of Polystyrene Batch Reactors Using Neural Network Based Model Predictive Control (NNMPC): An Experimental Investigation,” Control Engineering Practice, Vol. 19, No. 5, 2011, pp. 454-467.

[10] Daogang Peng, Hao Zhang, Conghua Huang, Fei Xia, Hui Li , “Immune PID cascade control based on neural network for main steam temperature system”, Intelligent Control and Automation (WCICA), 2011 9th World Congress, 21-25 June 2011.

[11] Temperature Process Station Model 3504 Student Manual 75944-20, By The Staff Of Lab-Volt (Quebec) Ltd.

[12] Control Of Dead-Time Processes, Springer By J.E. Normey-Rico And E.F. Camacho (2007).

[13] Neural Network Design, By Martin T. Hagan , Howard B. Demuth , Mark H. Beale, (1996).

[14] Temperature Process Station Model 3504 Student Manual 75944-20, By The Staff of Lab-Volt (Quebec) Ltd.