Comparison Of An Artificial Neural Network Controller And Pid Controller In On Line Of Real Time Industrial Temperature Process Control System

[1] Abdulgani Albagul
[1]Engineering and Information Technology Research Center, Baniwalid, Libya
Email: [1]albagoul@yahoo.com

[2] Mosa Abdesalam,
[2]College of Electronic Technology Baniwalid-Libya

[3] Hafed Efheij,
[3]College of Electronic Technology, Tripoli-Libya
[3]hafedrs@gmail.com

[4] Bileid Abdulsalam.
[4]Engineering and Information Technology Research Center, Baniwalid, Libya
[4]abuahmedasdira051@gmail.com

***Abstract*: The conventional PID (proportional-integral-derivative) controller is widely applied in industrial automation and process control because of its simple structure and robustness. However, it does not work well for nonlinear system, time-delayed linear system and time varying system. This paper provides a new approach of PID controller that is based on an artificial neural network and an evolutionary algorithm. An Artificial Neural Network is an effective tool for a highly nonlinear system. Neuro control algorithms are mostly implemented for the application of robotic systems and 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 manner. 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 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. The performance the ANN is compared to the PID controller based on set point change, effect of load disturbances and processes with variable dead time. The result shows that ANN outperforms the PID controller.**

Index terms: process control Neural network (NN), PID controller, Temperature control, System modelling.

I. INTRODUCTION

Artificial Neural Networks (ANNs) are 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 a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by examples. An ANN is conFig.d 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 also true in the case of ANN. ANNs offer very large capabilities concerning complex system modeling, prediction, control and performance [1,2,3]. ANNs have the ability of learning and function approximation. In addition, the learning processes are independent of human intervention. 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 algorithms, including the delta rule and the steepest descent methods, Boltzman's algorithm, the back-propagation learning algorithm, the standard version of genetic algorithm, etc [4,5,6]. These traditional learning methods of ANN exhibits some deficiency including such as the problem of the slow speed of convergence, local minima, and the large amount of computation of network, etc. This leads to the difficult use of ANN controller [7,8]. In this paper, a new ANN controller which is based on NN is proposed. Here, an artificial neural network is used to approximate PID formula and train the weights of ANN. The simulation proves that the controller can produce a better control effect, and it is easily realized and the less amount of computation.

PID control systems are widely used as a basic control strategy for industrial control systems, they are applicable to many control problems due to its well-known simple structure. However, the tuning of the PID control systems is not always easy and they may perform poorly in some applications. Highly nonlinear system control with a constrained manipulated variable can be mentioned as an example. Classic PID controller has some limitations that could affect the performance of the system [9]. In the case of temperature control process, these limitations are:

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

II. SYSTEM DIAGRAM

The LAB-VOLT Temperature Process shown in Fig1 mainly consists of a 20-200 degree C (70-400 degree F) oven with a 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

Received-August 2020, Accepted-October 2020, and Published-December 2020

(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. 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].

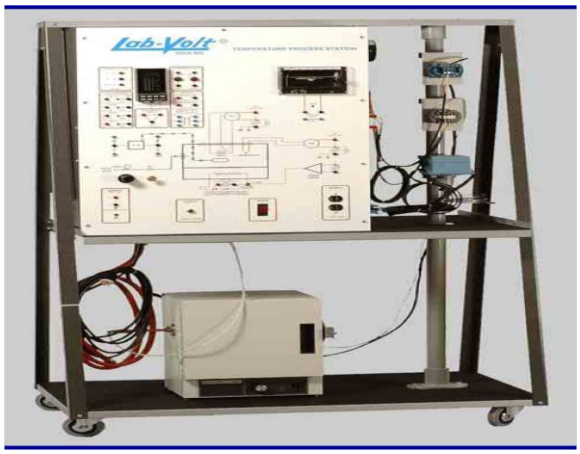

Fig. 1: The LabVolt temperature control process.

Fig. 2 shows the block diagram of the LabVOLT Temperature system Model 3504s, which consists of Arduino, Signal conditioning circuit, TRIAC, and a computer. The Lab-volt temperature Process can be represented by the loop schematic diagram as shown in Fig.3.

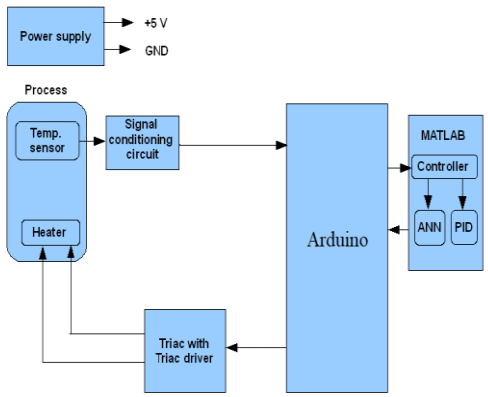

Fig. 2. Block diagram of system.

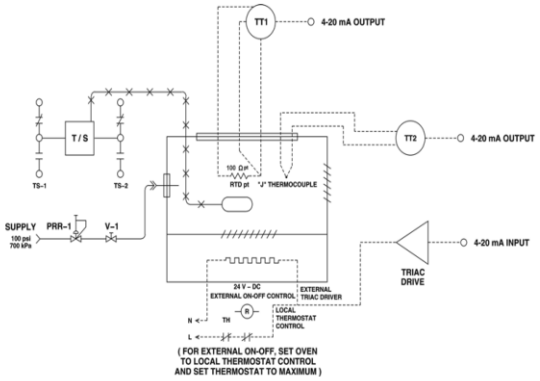

Fig. 3: Lab-volt temperature process schematic diagram.

The signal conditioning circuit is used to adjust the output of a temperature sensor to be compatible and satisfactory to drive the next stage. The analog output from the signal conditioning block is converted to a digital form so that it can be fed to the Arduino. The Arduino is the heart of the system which controls all necessary actions. Arduino measures the temperature from water tank and sends to PC and controls the heater actions. The PC is used to implement the control algorithms of PID and ANN which is written in MATLAB. Here, the comparison between the outputs of PID and ANN is conducted [13]. The output of PC is then fed to the Arduino which generates Pulse Width Modulation (PWM) to control the heater actions. The Opt isolators triac driver is an integrated circuit that is specifically designed 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 industry, a temperature control system. Fig. 4, shows the flowchart of the working principles based on a sequence of procedures to control the process.

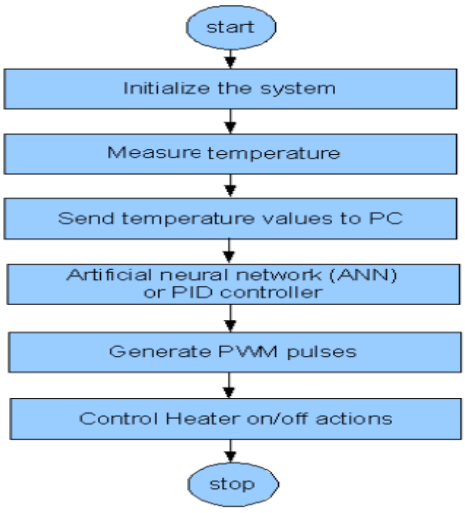

Fig. 4. The flowchart of the system.

The station is heated and the measured temperature is applied to the Artificial Neural Network controller and PID

Received-August 2020, Accepted-October 2020, and Published-December 2020

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, and the ratio ON-time to OFF-time is varied. This continues untill the desired temperature is achieved. Based on the observations, the neural network that gives best performance is then compared to a conventional PID controller to control the same process. Direct output comparisons are made with respect to set-point changes, effect of load disturbances, and variable dead time.

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, many of these are transformed into digital forms 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, K_p is the proportional gain, T_i is the integral time constant and T_d 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, only the proportional control action will be used to achieve an oscillatory output response as shown in Fig. 5.

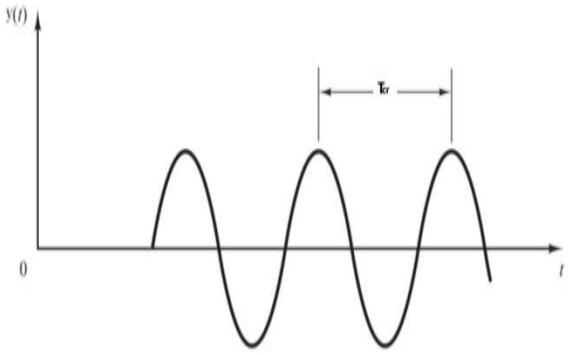

Fig. 5. The oscillatory response with a period of Tcr.

Ziegler and Nichols suggested that set the values of the parameters K_p, T_i and Td can be obtained 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)

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

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

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

V. NEURAL NETWORK CONTROL PROGRAMMING

The neural network toolbox in MatLab is used to design ANN controller. The design procedures are conducted in the following sequence:

Define neural network structure. This includes the number of inputs, the number of outputs, the number of hidden layers, the type of activation function used in each stage. In this case, 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 a very large integer number (e1), is a very small float number.

net.trainParam.goal = e1; % Error limit is e1

Then begin the training of the input (in1) and the 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 development 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 it 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 Fig. 6.

Received-August 2020, Accepted-October 2020, and Published-December 2020

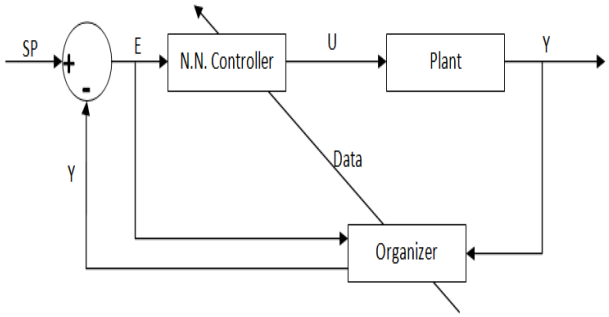

Fig. 6: The block diagram of neural network controller

The flowchart of neural network control operation is shown in Fig. 7.

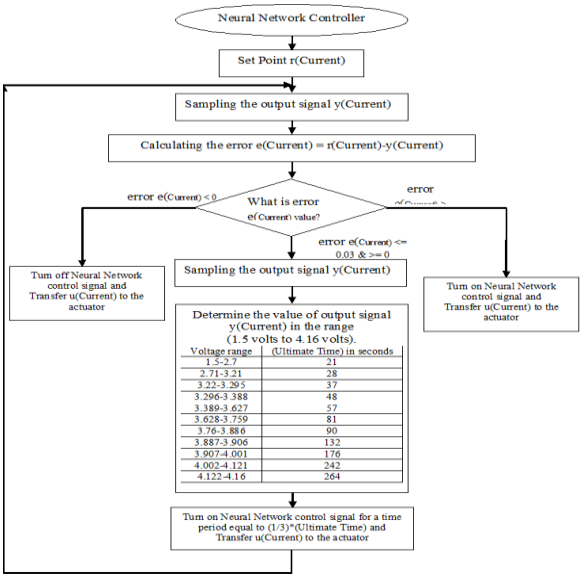

Fig. 7. The flowchart of neural network control operation.

VI. EXPERIMENTAL RESULTS

In this experimental result, the step response of Lab-volt Temperature Process will be obtained 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 it.
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 Fig.8, save the step response data in the computer to be used for analysis, stop the recorder and switch off the power.

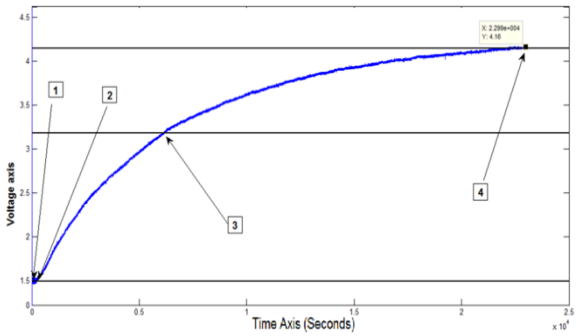

Fig. 8. Step response of the process using sdaq

10. From Fig. 8, perform the required calculations as follows [13]:

From Fig. 8 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 (T_o) → 1.5 Volts ≡ 19° C
Temp. Finish (T_f) → 4.13 Volts ≡ 118.5° OC
Temp. Change $T_c = T_f - T_o$ → 99.5° OC
Process Gain

- The Temperature Change (T_c) 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 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 (τ_d) = time difference between input step change (Sample 31) and point where temperature started to increase (Sample 209) from Fig. 8.

τ_d = 46 Sec - 6.82 Sec = 39.18 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 Fig. 8.

τ = 1349 Sec - 46 Sec = 1303 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 obtained:

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

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

Received-August 2020, Accepted-October 2020, and Published-December 2020

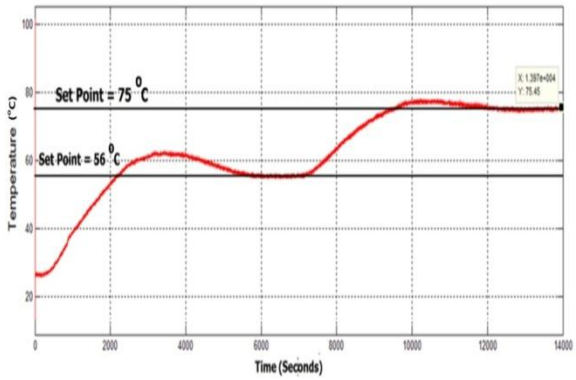

Fig. 9. PID controller response using FOXBORO controller

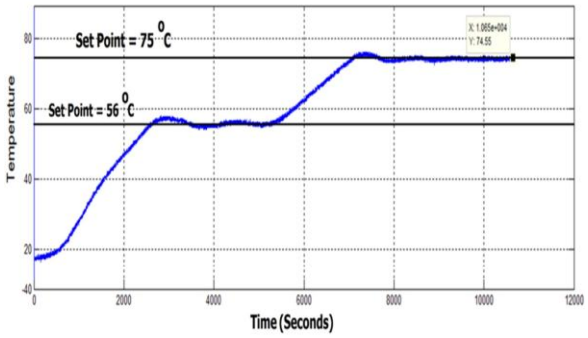

Fig. 10. PID controller response using MATLAB-Arduino controller

- 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. The number. of iterations to control the temperature at set point within dead time is smaller.
- Observations for neural network controller
  1. Response time to reach towards set point is less.
  2. Output does .not changes over all the process from set point.
  3. The number. of iterations to control the temperature at set point

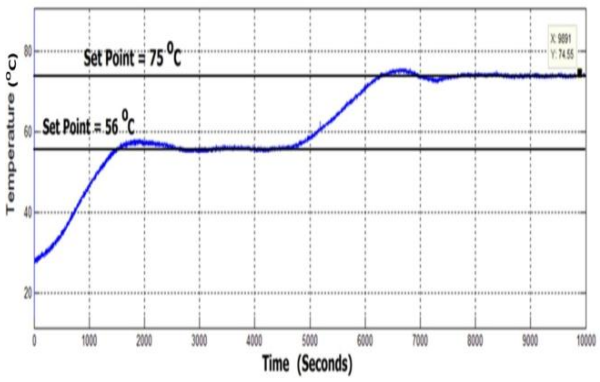

Fig. 11. neural network controller response using MATLAB-Arduino controller

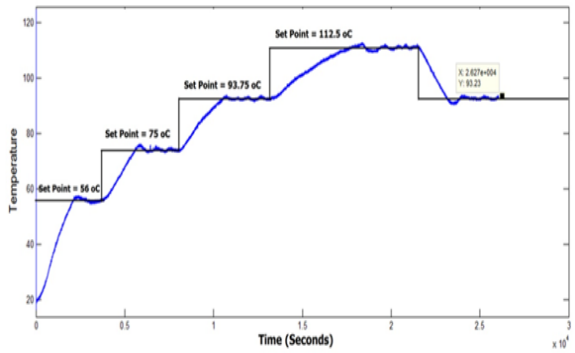

Fig. 12.Neural network controller response using matlab-sdaq controller. for multi step input

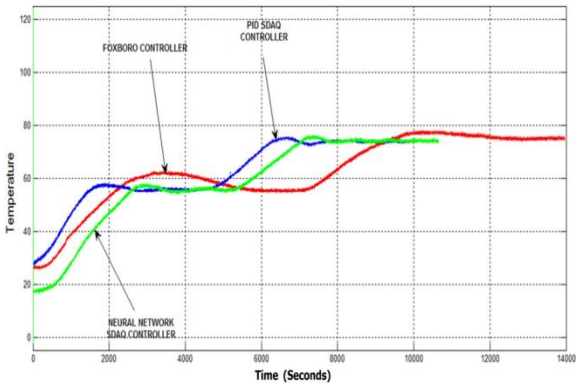

Fig. 13. Comparison between foxboro controller, pid sdaq controller and neural network sdaq controller

Table 2. Comparison of PID controller & AAN controller.

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

VII. 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 been 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 and processes with variable dead time.

REFERENCES

Received-August 2020, Accepted-October 2020, and Published-December 2020

[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.

Received-August 2020, Accepted-October 2020, and Published-December 2020