Artificial Neural Network and PID Controller in Real Time Process Control System

Hasan N. Naji
Department of Control Engineering College of Electronic Technology Tripoli- Libya
hasan.naji2010@gmail.com

نظام التحكم في الشبكة العصبية الاصطناعية ووحدة التحكم PID في الوقت الفعلي

الملخص:

نظام الكرة والشعاع هو جهاز مختبري يتميز بعدم الخطية العالية في ديناميكيتها. أهداف هذا المشروع هي تطوير نموذج للكرة ونظام الشعاع الذي يأخذ في الاعتبار العوامل غير الخطية و تأثيرات الاقتران وتصميم المشتق التكاملي النسبي جهاز تحكم (PID) لتنظيم وضع الكرة. النظام يشتمل على متحكم Arduino الذي يستقبل موضع الكرة والقياسات من جهاز استشعار المسافة بالموجات فوق الصوتية ويقارن بهم إلى الموضع المطلوب. خوارزمية PID، وهي مبرمجة مسبقا في Arduino حيث يولد إشارة تحكم التي تقوم بتشغيل محرك سيرفو لضبط موضع الكرة وتحقيق الهدف المسافة المطلوبة. للحصول على معلمات PID الدقيقة (Kp, Ki)، يتم استخدام تحليل زيجلر نيكولز وبرنامج MATLAB التلقائي، ويتم إجراء مقارنة بين الطريقتين.

الكلمات المفتاحية: *الشبكة العصبية الاصطناعية (ANN)، جهاز التحكم PID، التحكم في درجة الحرارة، نظام التحكم في العمليات.*

Abstract:

The ball and beam system is a laboratory apparatus characterized by high nonlinearity in its dynamics. The primary objectives of this project are to develop a model of the ball and beam system that takes into account nonlinear factors and coupling effects, and to design a Proportional Integral Derivative (PID) controller to regulate the ball's position. The system comprises an Arduino microcontroller that receives ball position measurements from an ultrasonic distance sensor and compares them to the desired position. The PID algorithm, which is preprogrammed in the Arduino, generates a control signal that drives a servo motor to adjust the ball's position and achieve the desired distance. To obtain the exact PID parameters (Kp,Ki), Ziegler-Nichols analysis and MATLAB auto tuner are employed, and a comparison is made between the two methods.

Keywords: *Artificial Neural Network (ANN),PID controller, Temperature control, Process control system.*

Introduction:

Control systems engineering is a specialized branch of mathematics that focuses on managing the control of continuous, dynamic systems in engineered processes and machinery. The primary objective of this field is to create a control model that can optimize the control of these systems with minimal delays or overshoots, while also ensuring control stability.

To accomplish this objective, a controller with the appropriate corrective behavior is necessary. This controller oversees the controlled process variable (PV) and compares it to the reference or set point (SP). The error signal, or the difference between the actual and desired values of the process variable, is then used as feedback to generate a control action that will adjust the controlled process variable to the set point value.

The figure displayed in Figure 1 showcases the ball and beam system, which is composed of a base, beam, ball, lever arm, support block, and motorized gear. The beam is connected to the support block that remains fixed on one side, while a movable lever arm is attached on the other side, which is controlled by a servo motor. The ball is positioned on the beam and has the ability to move freely along the entire length of the beam. An electrical control signal is applied to the motor, which causes the beam to tilt about its axis at a particular angle. This movement of the beam results in the acceleration of the ball along the beam, which is proportional to the angle of the beam. An Ultrasonic sensor is utilized as a distance sensor and is located in a slot along the beam. The sensor's parameters are coded on the Arduino IDE and are capable of sensing the current linear actual position of the ball on the beam. The measured position is then fed back to the control system to create a closed-loop control mechanism for the ball and beam balancing system.

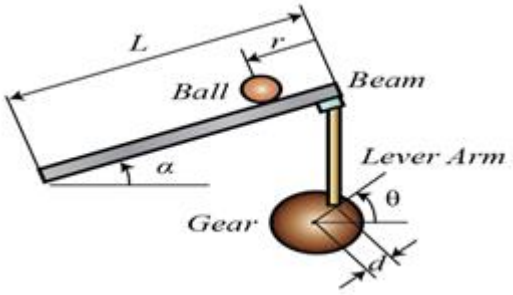


Figure 1: Ball and Beam System

The challenge at hand is to maintain the position of a ball on a freely rolling beam. The objective is to automatically regulate the ball's position by adjusting the beam's angle, which in turn affects the ball's acceleration rather than its position. However, controlling the ball's acceleration allows for precise control of its position. In control engineering, the system is considered open loop unstable as the system output (i.e. the ball's position) increases indefinitely for a constant input (beam angle). To ensure the ball remains in the desired position on the beam, feedback control is required. As such, a controller that is tailored to the dynamics of the ball and beam system must be developed. Table 1 provides the parameters necessary for modeling the dynamics of the system.

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, learn 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. ANN offers very large capabilities concerning complex system modelling, prediction, control and performance (Rahman & Hoque, 1997; Stankovic & Saric, 2003; Singh & Rai, 2005). The tuning of the PID control systems is not always easy, however, they are still applicable to many control systems. The artificial neural network has the ability of learning and performing approximation. In addition, the artificial neural network learning

processes are independent of human intervention and expert experiences. For such situations, many researches have used ANN to approximate PID formula to realize ANN controller. But the learning method of ANN usually adopts some traditional algorithms, including the delta rule, the steepest descent methods, Boltzman's algorithm, the back-propagation learning algorithm, the standard version of genetic algorithm (Vaclavek & Blaha, 2005; Bäck & Schwefel, 1993; Bartlett & Downs, 1990). These traditional learning methods of ANN have 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 (Magoulas et al., 2004; Hosen & Hussain, 2012). In this paper, a new ANN controller which is based on NN is proposed. Here, an artificial neural network is used to approximate PID parameters and train the weights of ANN. The simulation results shows that this type of controller can achieve a better control effect, easily realized and has less amount of computation.

SYSTEM BLOCK DIAGRAM .

Figure 1 shows the system block diagram of the LabVOLT-Model 3504 process control system, which consists of following main blocks: Arduino, Temperature system, Signal conditioning circuit, TRIAC, Computer .

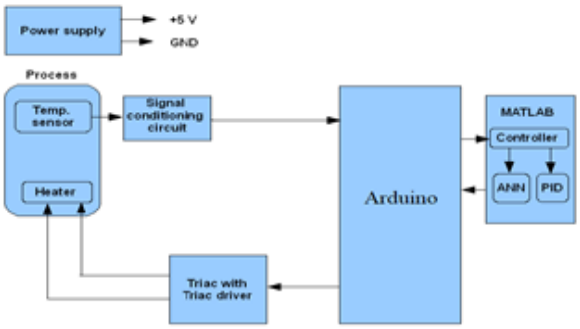


Figure 1: Block diagram of system

●Temperature system LabVOLT - Model 3504; The LabVOLT 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.

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 **(**Efheij & Albagul, 2021).

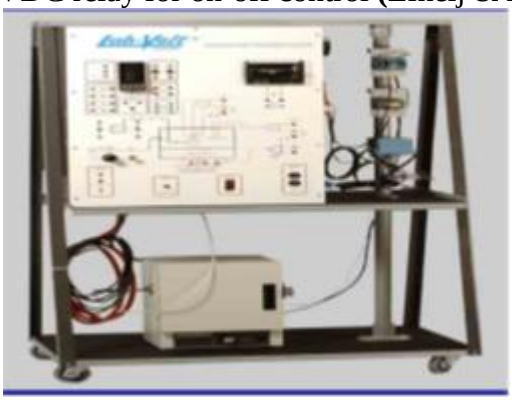


Figure 2: Lab-volt temperature process

The Lab-volt temperature Process can be represented by the loop diagram shown in Figure3.

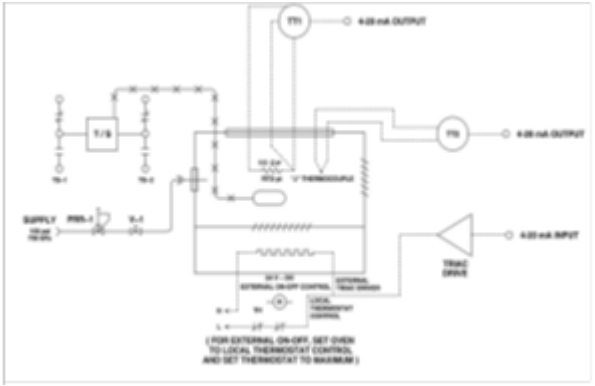


Figure 3: lab-volt temperature process schematic diagram.

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 **(**Efheij & Albagul, 2021). 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: SDAQ(Smart Data acquisition) 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 (Albagul et al., 2019). 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.

THE WORKING OF SYSTEM MODEL

The process considered here is one of the most widely used processes in the process industry, a temperature control 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.

ZIEGLER-NICHOLS RULES FOR TUNNING PID CONTROLLER

Interestingly, more than half of industrial controllers used today are PID control schemes or modified PID. Analog PID controllers are mainly hydraulic, pneumatic, electronic, and electrical combinations. Today, many of these controllers are transformed into digital form using microprocessors which indicates that PID controllers follow the equation (1).

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

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 equation of PID controller is: (2).

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

Ziegler and Nichols method determines the values of the proportional gain integral time and derivative time based on the transient response characteristics of a plant under study. Parameters of the of PID controllers or tuning of PID controllers can be estimated by the specified experiments on the plant.

●ZIEGLER–NICHOLS CLOSED LOOP METHOD

In this method, will use the proportional control action only as shown in Figure 4.

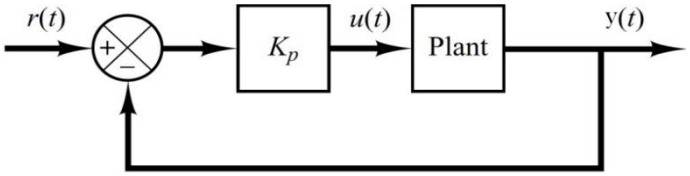


Figure 4: Closed-loop system with a proportional controller

increase K_p from 0 to a value named critical gain (K_{cr}) at which the output first exhibits sustained oscillations and this value is critical value. (If can not obtain sustained oscillations for any value of K_p, then this method does not apply).

And thus, the critical gain K_{cr} and the critical period T_{cr} are experimentally determined as shown in Figure 5.

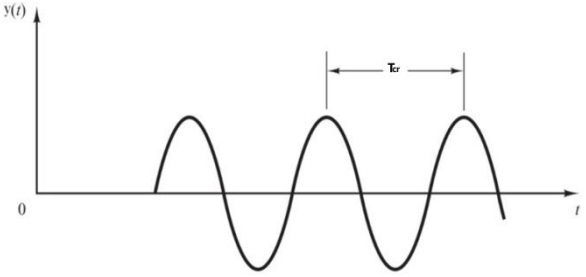


Figure 6: Oscillation response with period tcr (is measured in sec.).

Ziegler and Nichols suggested that set the values of the parameters K_p , T_i , and T_d according to the formula shown in Table 1.

Table 1: ZIEGLER–NICHOLS tuning rule based on critical gain K_{cr} and critical period T_{cr}.

Type of Controller	K_p	T_i	T_d
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 (3).

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

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

$$\frac{U(s)}{E(s)} = 0.075K_{cr}T_{cr}\frac{\left(s + \frac{4}{T_{cr}}\right)^2}{s} \quad ---(3)$$

However, there is another way to find out Kcr and Tcr, for instance; if the root-locus do cross the jω axis and also the transfer function is known. In this case, the root-locus method can be utilized to find the critical value Kcr and the frequency of the sustained oscillations ωcr , where 2π/ωcr = Tcr, and their values are calculated from the crossing points of the root-locus with jω axis (Normey-Rico & Camacho, 2007).

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. After completing this training can use the trained neural network 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. To simplify 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.

This neural network is designed and the overall block diagram of neural network controller is shown in Figure 7.

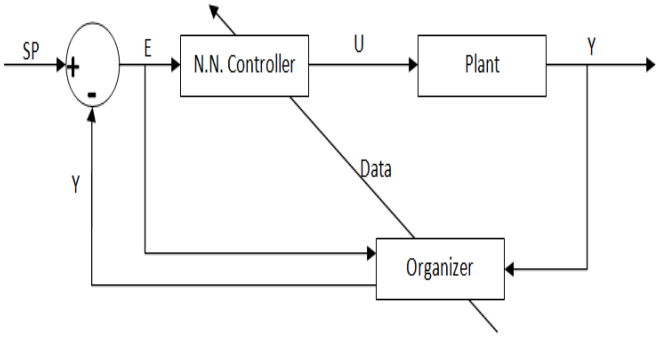


Figure 7: Overall neural network controller block diagram

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

RESULTS

In this experimental result will be obtained the step response of Lab-volt Temperature Process using Arduino, by applying the following procedure (Hagan et al., 1996):

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.

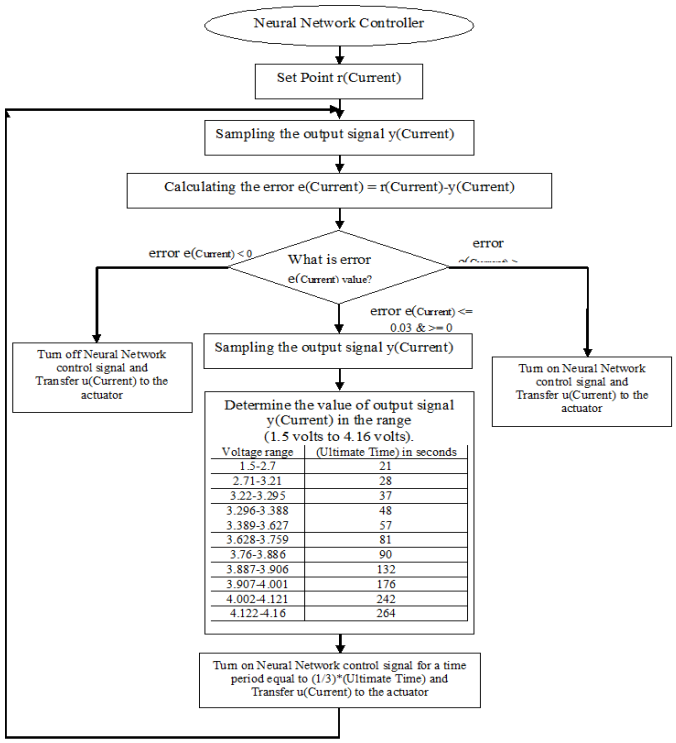


Figure 8 : The flowchart of neural network control operation.

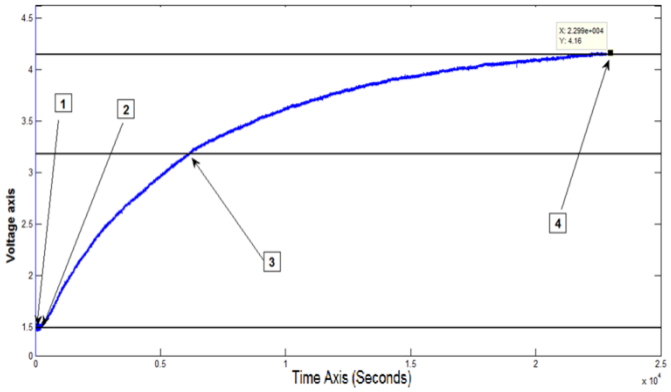


Figure 9: Step response of lab-volt temperature process using sdaq.

9.Perform the required calculations as follows (Albagul et al., 2019): From Figure 9 can obtain the following information.

- Point (1) at sample 31, → Then Point (1) at time: 6.82 Sec.
- Point (2) at sample 209, → Then Point (2) at time: 46 Sec.
- Point (3) at sample 6132, → Then Point (3) at time: 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:

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

•The input change as a percent of span:

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

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.

τd = 46 Sec - 6.82 Sec = 39.18 Sec = 0.653 min

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.

τ = 1349 Sec - 46 Sec = 1303 Sec = 21.7min

$$because\ \frac{\tau}{\tau d} = \frac{1303}{39.18} = 33.25\ and\ this > 10\ then\ plant\ easy\ to\ control.$$

Then the final transfer function of the system is:

$$G(s) = \frac{1.768}{1303s + 1} \times e^{-39.18s} \text{ if use } \tau \text{ \& } \tau d\ in\ seconds$$

$$G(s) = \frac{1.768}{21.7s + 1} \times e^{-0.653s} \text{ if use } \tau \text{ \& } \tau d\ in\ minutes$$

And the response of PID controller of the kit using FOXBORO controller as shown in Figure 9, where the Figure shows the relation between samples (Time) and temperature. The response depicted in Figure 9 shows an overshoot in the beginning of the process, the reason for this is due to incomplete identification process. And after the identification process is started the parameters of the controlled system are estimated where these parameters are unknown and starting with random values at the beginning of the identification process. And after the parameters of the controlled system are estimated the response of the controlled system becomes without any sustained oscillation and the controller eliminate an off-set and response gives small rise time and small settling time and the response stabilized.

The response depicted in Figure 11 shows an overshoot in the beginning of the process the reason for this is when the response of the controlled system becomes near set point then the only variable that work in this region is the integral action, and after a small period of time the response is improved.

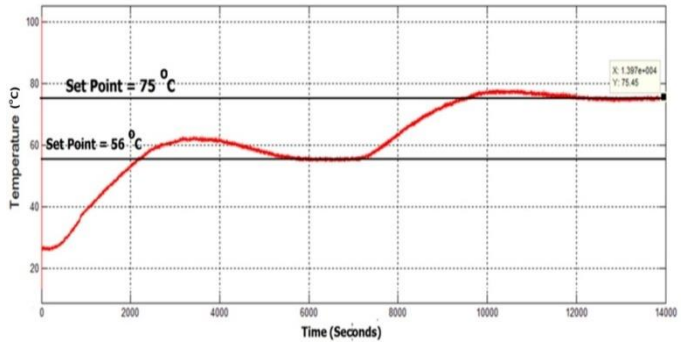


Figure 10: PID controller response using FOXBORO controller.

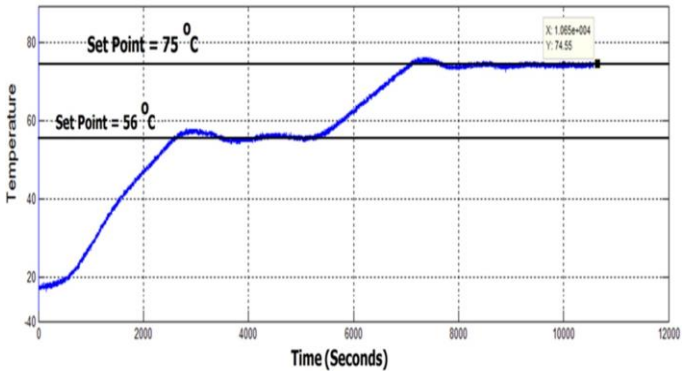


Figure 11: PID controller response using MATLAB-Arduino controller.

And after training the data to work as Neural network controller, used to obtain the Neural network controller response using MATLAB- Arduino controller and the response is shown in Figure 12.

The response depicted in Figure 12 shows an overshoot in the beginning of the process the reason for this is when the response of the controlled system becomes near set point the neural network controller is working, the neural network will identify the value of received signal from plant and based on this received signal the controller will produce the control signal output, and after a small period of time the response is improved. The response of the controlled system becomes with small fluctuations and the controller eliminate an off-set to small value and response gives small rise time & settling time and the response stabilized at value Nearly equal to the set points.

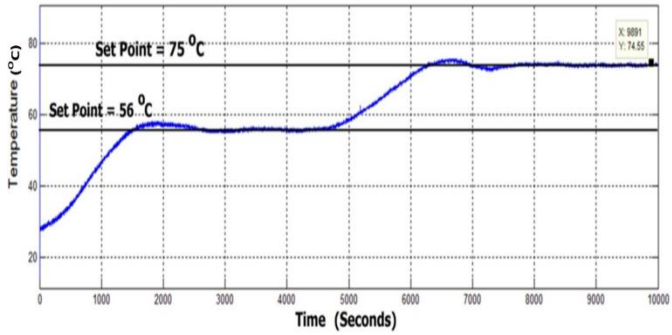


Figure 12: Neural network controller response using MATLAB-Arduino.

And the following plot is the neural network controller response for (time vs. temperature), is applied to more than one step input (2.5 volts, 3 volts, 3.5 volts, 4 volts and 3.5 volts) using MATLAB-Arduino neural network controller and the result is shown in Figure 13.

Figure 14 below shows a comparison between FOXBORO controller, PID Arduino controller and Neural network Arduino controller.

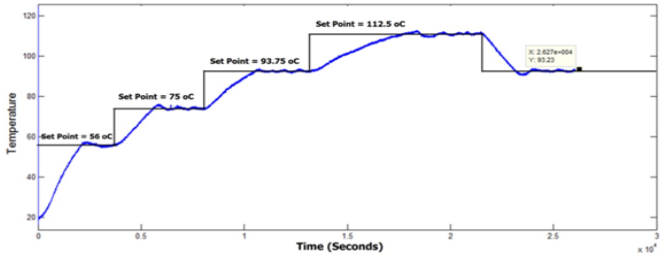


Figure 13: Neural network controller response using matlab-sdaq controller

Figure 14 below shows a comparison between FOXBORO controller, PID Arduino controller and Neural network Arduino controller.

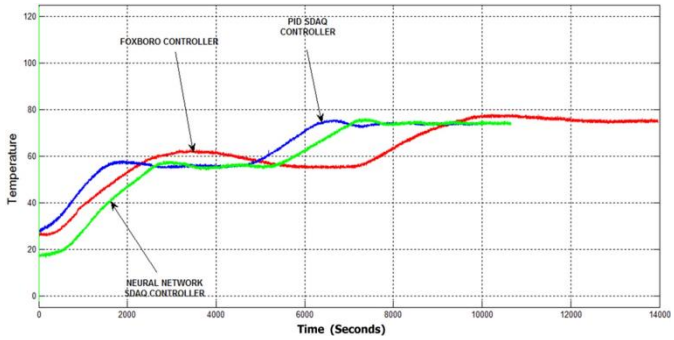


Figure 14: Comparison between foxboro controller, pid sdaq controller and neural network sdaq controller.

●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 it various network topologies and learning rules. feed-forward back propagation network is used in this system. A neural network controller has been designed and 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 PID controller to evaluate the system performance. The results shows that the performance of the neural network controller achieved much better tracking and totally overcomes the disadvantages of PID Controller in dealing with the change in the set-point, effect of load disturbances and , dead time variation in the processes.

References:

Albagul, A., Efheij, H., & Alsharif, B. (2019). Comparison of Artificial Neural Network Controller and PID Controller in on Line of Real Time Industrial Temperature Process Control. *International Journal of Control Systems and Robotics*, *4*.

Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, *1*(1), 1-23.

Bartlett, P., & Downs, T. (1990). *Training a neural network with a genetic algorithm*. University of Queensland.

Efheij, H., & Albagul, A. (2021, May). Comparison of PID and artificial neural network controller in on line of real time industrial temperature process control system. In *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA* (pp. 110-115). IEEE.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). O. Jesu s. *Neural Network Design*, *20*.

Hosen, M. A., & Hussain, M. A. (2012). Optimization and control of polystyrene batch reactor using hybrid based model. In *Computer Aided Chemical Engineering* (Vol. 31, pp. 760-764). Elsevier.

Magoulas, G. D., Plagianakos, V. P., & Vrahatis, M. N. (2004). Neural network-based colonoscopic diagnosis using on-line learning and differential evolution. *Applied Soft Computing*, *4*(4), 369-379.

Normey-Rico, J. E., & Camacho, E. F. (2007). The Smith predictor. *Control of Dead-Time Processes*, 131-163.

Peng, D., Zhang, H., Huang, C., Xia, F., & Li, H. (2011, June). Immune PID cascade control based on neural network for main steam temperature system. In *2011 9th World Congress on Intelligent Control and Automation* (pp. 480-484). IEEE.

Rahman, M. A., & Hoque, M. A. (1997). Online self-tuning ANN-based speed control of a PM DC motor. *IEEE/ASME Transactions On Mechatronics*, *2*(3), 169-178.

Singh, P., & Rai, P. (2005, August). An ANN based X-PC target controller for speed control of permanent magnet brushless DC motor. In *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.* (pp. 1027-1032). IEEE.

Stankovic, A. M., & Saric, A. T. (2003, June). An integrative approach to transient power system analysis with standard and ANN-based dynamic models. In *2003 IEEE Bologna Power Tech Conference Proceedings*, (Vol. 3, pp. 6-pp). IEEE.

Vaclavek, P., & Blaha, P. (2005, August). Hybrid speed estimation scheme for ac induction machine sensorless control. In *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.* (pp. 1455-1460). IEEE.