Real-time control of tank water level with neural-PID

Abstract:

In this paper, a gradient-free neural network PID control is designed for water level control in a single tank. The control system includes two parts: a neural network system and a PID controller. Due to the nonlinearity of the water tank and its process, conventional PID cannot adapt to the process changes; therefore, a neural network is approach is applied for online PID gain scheduling.

The proposed control system is implemented to level control of an actual laboratories plant using the real-time toolbox in Simulink and data acquisition card. The plant is a single-input single-output system with constant water in-flow. Both simulation and real-time results have shown adaptability and robustness of the neural network PID control system of water tank level.

Keywords: Neural Network; Gradient-free; PID; Real-time control; Adaptive System; Tank Level.

I. INTRODUCTION

Liquid level control is applied in many fields, including the petroleum industry. Keeping the tank level in a particular range is essential to guarantee the desired quality or other desired criteria. However, level control is a typical nonlinear and time-delay system. The characteristics of valves, tanks, and other system parts make it challenging to obtain an accurate mathematical model. Therefore, conventional approaches to control these systems cannot reach a good output.

In situations where overshoot is not acceptable and rise time is important, conventional controllers such as PID and PI will not provide the desired output [1]. Also, fixed-gain PID controllers do not perform well in nonlinear and time-varying systems. Many approaches have been proposed to overcome these limitations. A second-order sliding mode controller has been implemented on a quadruple tank system [2]. Using a twisting algorithm, they overcome the chattering problem of sliding mode. In [3], a fuzzy gain-scheduled PID controller is designed for liquid level control, which has a better performance compared with the PID controller in overshoot and settling time. Other works on using fuzzy controllers show that it has a better performance than conventional PID; in most cases, the system still has some overshoot [4], [5], [6].

In this paper, a neural-PID control system is demonstrated, developed, and implemented on an actual laboratory plant using Simulink desktop real-time and data acquisition card. Due to the gradient-based neural network's slow convergence, trapping at local minima, and not being robust in the presence of large bounded input disturbances, we use a gradient-free neural network to avoid these problems [7].

II. PLANT CHARACTERISTICS

The laboratory plant used can be referred to as "OSK4550, Liquid Level Automatic Control Model Plant, Ogawa Seiki CO., LTD." Fig. 1. The plant consists of three main parts; level transmitter, tank, and control valve.

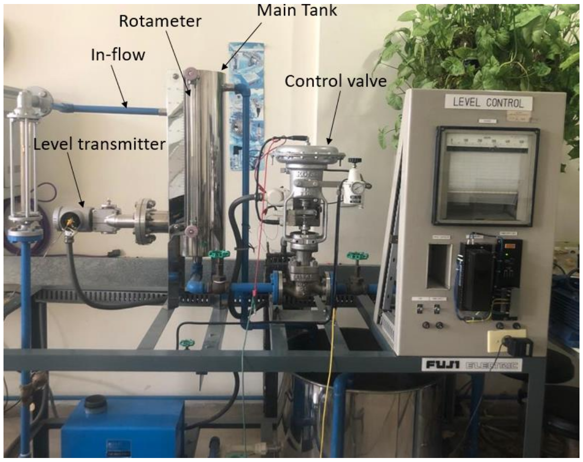

Figure 1 Actual Water-Level Control Plant

A. Level transmitter

The level transmitter in this plant is a Fuji Electric level transmitter that needs a 24 v DC supply and has a range of 0-500 mm that outputs a signal in the range of 4-20 mA. This transmitter uses the presser at the end of the tank to calculate the liquid level. The equation for this process is below.

$$h = \frac{P - P_0}{\rho G} \tag{1}$$

Which ρ is the liquid torque, G is the gravity constant, and P_0 is the air pressure.

B. Liquid Tank

The main tank has 0.5 meters in height, with two pipes at the top, one for in-flow and one for overflow, and one pipe at the bottom for outflow, connected to a control valve.

C. Control Valve

The control valve is normally open, and it is the only control actuator in this system that controls the liquid level in the tank. The control signal to this valve should be in a range of 4-20 mA. Equation (2) can express the outflow of this valve.

$$\mathrm{Q} = C_v\sqrt{\Delta P/G} \tag{2}$$

C_v is the flow constant, and ΔP is the pressure difference between two points of the valve.

The liquid is stored in the storage tank, and it flows into the tank at a constant in-flow rate in 50-500 lit/h. Using the "Advantech 1716" data acquisition card, the tank liquid level is transmitted to a computer for real-time processing. After processing the signal in the Simulink environment, the control signal is applied to the control valve through the data acquisition card.

III. Mathematical Model

Fluid flows to the tank through a pipe at the top of the tank with a constant flow Q_{in}And it outflows from the bottom pipe with the Q_{out} flow. The difference between inflow and outflow is the rate of storing fluid in the tank.

$$Q_{in} = Q_{out} + Q \tag{3}$$

Q is the rate of storing fluid which can be written as:

$$Q = A\, dh/dt \tag{4}$$

Where h is the height of liquid in the tank.

The outflow Q_{out} can be written as:

$$Q_{out} = C_d a\sqrt{2gh} \tag{5}$$

Where a is the area of the outflow pipe, and C_d is the rate of outflow. Therefore, (3) can be written as:

$$Q_{in} = Q_{out} + Q = C_d a\sqrt{2gh} + A\frac{dh}{dt} \tag{6}$$

From Eq. (5) we have

$$\frac{dh}{dt} = \frac{1}{A}\left(Q_{in} - C_d a\sqrt{2gh}\right) \tag{7}$$

In [5], the linear transfer function from the data of the plant is demonstrated as below:

$$G = \frac{0.0002974}{s^2 + 3.772s + 0.03263} \tag{8}$$

This equation is necessary for conventionally obtaining PID gains.

IV. Neural PID Controller system

A. Neural network structure

PID control is widely used in industries and factories due to its simple structure and strong stability. Hence, the PID controller cannot work properly in nonlinear systems. The neural network system is used here to modify PID gains. An adaptive PID controller is designed to obtain appropriate parameters based on the current situations of the plant. Due to the gradient-based BP training algorithm's slow convergence, the possibility of trapping at local minima, and not being able to find the global minimum point when the neural network is subjected to large bounded input disturbances, many works have been done on gradient-free algorithms [7]. Therefore, in this paper, the Lyapunov stability-based neural network is used. Since this algorithm uses a time-dependent learning rate, when the time becomes large enough, the effect of error will be eliminated from updating rules. We used a time-independent learning rate to overcome this problem and proved the network's stability. The structure of this neural network used is shown in Figure 2 [7].

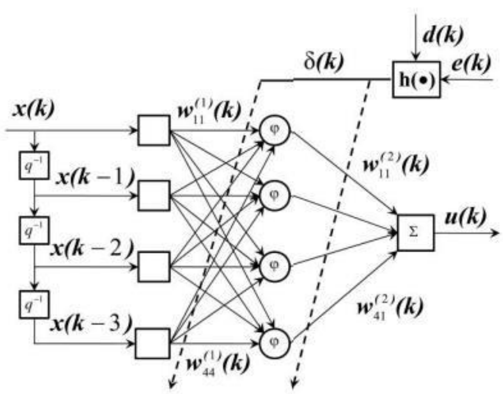

Figure 2 Lyapunov based neural network structure

The neural network is three-layer feedforward neural network. According to formula (1), u(k) is the output of the system, and x(k) is the neural network's input, x(k)=[x(k), x(k-1),…,x(k-m+1)]; m is the number of input neurons, e(k) and d(k) are the error and network's desired response respectively. The network has m input neurons, n hidden neurons, and one output. The output of the i-th neuron of the hidden layer $S_j(k)$ and the output of neural network $u(k)$ is expressed as

$$u(k) = \sum_{j=1}^{4} w_{1j}^{(2)}(k) S_j(k) \quad (9)$$

$$S_j(k) = \varphi\left(\sum_{i=1}^{4} w_{ij}^{(1)}(k) x(k-i+1)\right) \quad (10)$$

Where $w_{1j}^{(2)}$ and $w_{ij}^{(1)}(k)$ are the weights of the neural network output layer and hidden layer, respectively, and $\varphi(.)$ is the hidden layer's activation function.

The weights of the neural network are updated via backpropagation with $\delta(k) = h(d(k), e(k))$. The weights update laws are equations (11) and (12).

$$w_{1j}^{(2)}(k) = \frac{\delta(k)}{n S_j(k-1)} \quad (11)$$

$$w_{ij}^{(1)}(k) = \frac{1}{m x(k-i+1)} \varphi^{-1}\left(\frac{\delta(k)}{n w_{1j}^{(2)}(k)}\right) \quad (12)$$

Where

$$\begin{aligned} \delta(k) &= h\big(d(k), e(k)\big) \\ &= \beta e(k-1) + d(k) \end{aligned} \quad (13)$$

β is the learning rate that determines the behavior of error convergence.

B. Neural Network stability

The selected Lyapunov function is $V(k) = e^2(k)$. Therefore, by calculating $\Delta V(k)$ we have

$$\begin{aligned} \Delta V(k) &= \frac{V(k) - V(k-1)}{k-(k-1)} \\ &= V(k) - V(k-1) \\ &= e^2(k) - e^2(k-1) \end{aligned} \quad (14)$$

Therefore, for $e^2(k)$ we have

$$e^2(k) = \left(r(k) - y(k)\right)^2 \quad (15)$$
$$= r^2(k) - 2r(k)y(k) - y^2(k)$$

As we know, y(k) is the neural network's output, so

$$y(k) = \sum_{j=1}^{n} w_{1j}^{(2)}(k) S_j(k) \quad (16)$$
$$= \sum_{j=1}^{n} \frac{\delta(k)}{n S_j(k-1)} S_j(k)$$
$$= \delta(k) \sum_{j=1}^{n} \frac{S_j(k)}{n S_j(k-1)} = \delta(k) * \mu(k)$$

Where

$$\mu(k) = \sum_{j=1}^{n} \frac{S_j(k)}{n S_j(k-1)} \quad (17)$$
$$= \frac{1}{n}\left(\frac{S_1(k)}{S_1(k-1)} + \frac{S_2(k)}{S_3(k-1)} + \cdots + \frac{S_n(k)}{S_n(k-1)}\right)$$

Where $S_j(k)$ is the j-th hidden layer's output

$$S_j(k) = tanh\left(\sum_{i=1}^{m} w_{ij}^{(1)}(k) x(k-i+1)\right) \quad (18)$$

The standard deviation of $S_j(k)$ is 1, so Eq. (17) is

$$\mu(k) = \frac{1}{n}(1+1+1+1) = 1 \quad (19)$$

So, by combining Eq. (15), (16), and (19), we can calculate Eq. (14) now

$$\Delta V(k) = e^2(k) - e^2(k-1) \quad (20)$$
$$= \beta e^2(k) - e^2(k-1)$$
$$= -(1-\beta^2) e^2(k-1)$$

If $-1 < \beta < 1$ then we have $\Delta V(k) < 0$, which proves Lyapunov stability.

C. **Control system structure**

The control system consists of two main parts: PID controller and neural system tuner. The neural system tunes PID parameters. This system is consists of three neural blocks, NNP, NNI, and NND. Each of these neural blocks has two neural networks, NNC, which acts as the direct adaptive inverse controller and minimizes the closed-loop error, and NNE computes the desired control signal for NNC. These blocks are based on the Lyapunov function neural network tracking (LNT) control architecture [8]. Fig. 3 and Fig 4 show LNT blocks and control system structure.

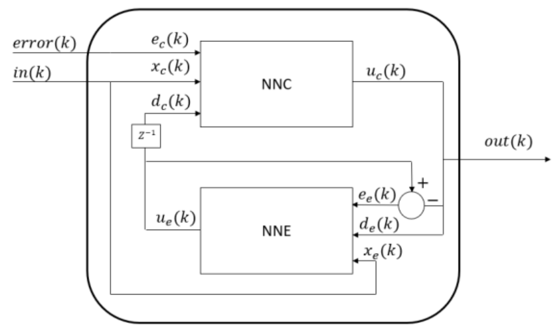

Figure 3 LNT block

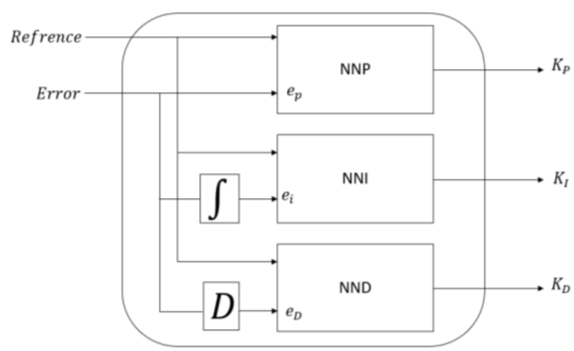

Figure 4 Neural network system to tune PID parameters

Where $\int$ represents integrator and D represents derivative.

V. **Simulation and Analysis**

We perform simulation and real-time control on this system to assess control system performance.

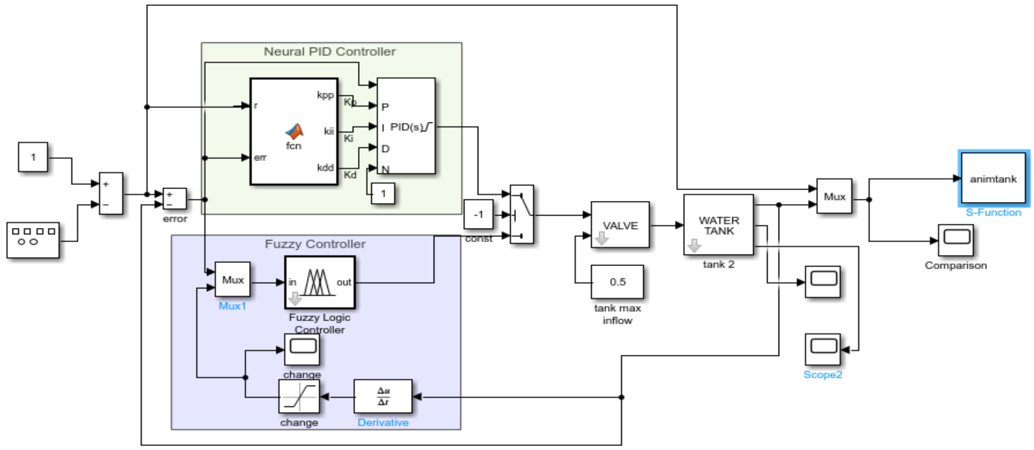

Figure 5 Simulink structure of control system with fuzzy and NN-PID controller

A. Simulation

We use Mathwork's water tank control in the simulation and compare our control system with a fuzzy controller. The Simulink structure is shown below

Fig. 6 shows the results of the step response.

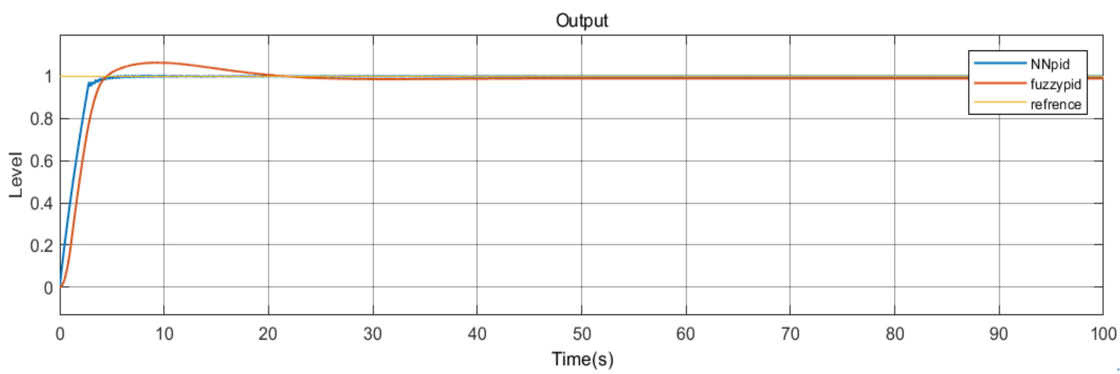

Figure 6 Step response of fuzzy and NN-PID controller

From Fig. 6, the fuzzy controller has a higher overshoot and rise time. Table 1 shows the comparison between these two controllers.

Table 1

	NN-PID	Fuzzy
t_r	2.325 s	2.643 s
OS	0.505 %	8.152 %
MSE	0.0051	0.0080

To have better insight into the performance of these controllers, Fig. 7 and 8 show the system's output and error when the setpoint is changing.

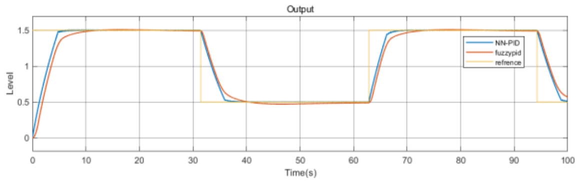

Figure 7 System output when setpoint change

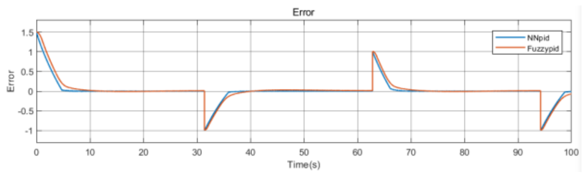

Figure 8 System's error when setpoint change

We can see that the NN-PID outperformance the fuzzy controller.

Fig. 9 shows the tuned PID parameters by the neural network.

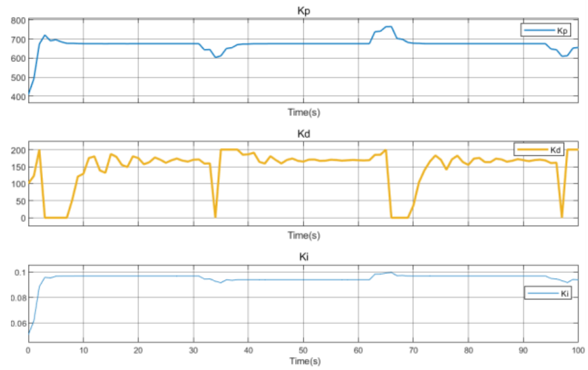

Figure 9 The PID parameters tuned by Neural network

B. Real-time Control

First, we should consider some practical considerations such as anti-windup for real-time process and control. The windup phenomenon causes the control valve to become fully open or completely closed due to the unbounded accumulation of integral action of a controller [9]. There are many techniques proposed to deal with the windup problem. In our work, we used the Clamping method in a conventional PID controller. The control valve has a limitation for its input signal, so the control signal must be limited in its range of 4-20 mA. For comparing the NN-PID, we use conventional PID with gains calculated from its data in previous works on this exact plant [5], which are Kd = 200, Ki = 3.5, and Kd=0.

To lower the effect of noises, the Kd from the neural system derive by 1000, which means Kd = Kd/1000.

The results of real-time control are shown below.

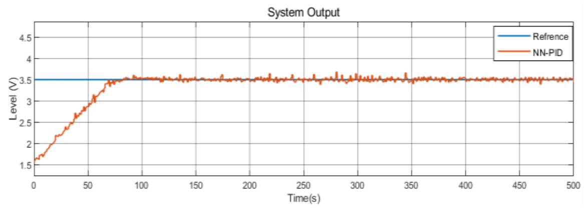

Figure 10 Tank level with NN-PID controller

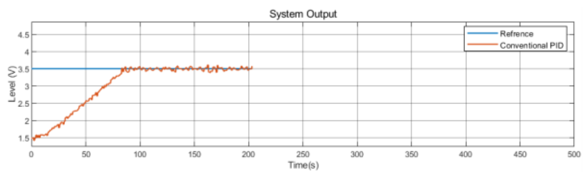

Figure 11 Tank level with conventional PID controller

In real-time, the control signal must be in a range so that the actuator does not go on and off too much, and it should be stable. The control signal of these two controllers is shown below.

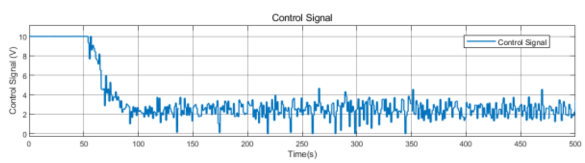

Figure 12 NN-PID control signal

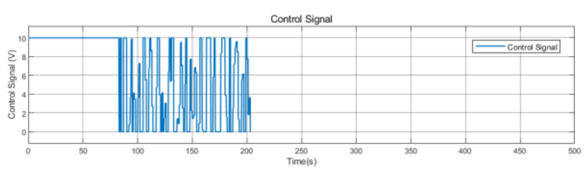

Figure 13 Conventional PID control signal

We can see that the PID controller control signal is not stable, and it has adverse effects on the valve and the entire system.

The PID parameters from the neural system are shown below

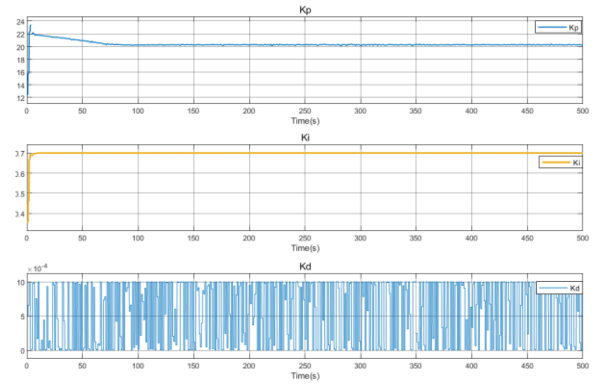

Figure 14 PID parameters tuned by neural network

To assess the robustness of NN-PID, we changed the in-flow water at 180s from 400 lit/h to 300 lit/h and at 320s from 300 lit/h to 500 lit/h. The figures below show that the controller responds to these changes very well, so the NN-PID is robust.

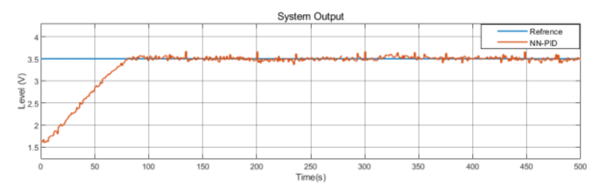

Figure 15 Tank level when the in-flow water rate change

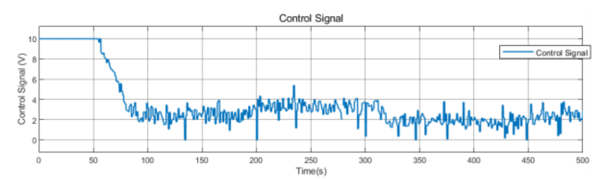

Figure 16 Control signal when the in-flow water rate change

VI. Conclusion

Liquid level control is vital in industries such as petroleum, water treatment, and paper-making. Because of its benefits, the PID controller is widely used in these processes. But, the PID controller is linear and cannot correctly control nonlinear systems. In this paper, a gradient-free neural PID controller is represented to control a laboratory water tank level. The main controller is a PID controller that its gains are tunning with a neural network system. This robust control system has better performance than commonly used PID and fuzzy controllers in both simulation and real-time control by comparing their rise time, overshoot, and mean square error.

References

[1] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, 2005.

[2] V. Chaudhari, B. Tamhane, and S. Kurode, "Robust liquid level control of quadruple tank system - Second order sliding mode approach," *IFAC-PapersOnLine*, vol. 53, no. 1, pp. 7–12, 2020, doi: 10.1016/j.ifacol.2020.06.002.

[3] S. Ahmad, S. Ali, and R. Tabasha, "The design and implementation of a fuzzy gain-scheduled PID controller for the Festo MPS PA compact workstation liquid level control," *Eng. Sci. Technol. an Int. J.*, vol. 23, no. 2, pp. 307–315, 2020, doi: 10.1016/j.jestch.2019.05.014.

[4] M. Alotaibi, M. Balabid, W. Albeladi, and F. Alharbi, "Implementation of Liquid Level Control System," *2019 IEEE Int. Conf. Autom. Control Intell. Syst. I2CACIS 2019 - Proc.*, no. June, pp. 311–314, 2019, doi: 10.1109/I2CACIS.2019.8825058.

[5] A. M. O. Fini, M. B. Gogani, and M. Pourgholi, "Fuzzy gain scheduling of PID controller implemented on real time level control," in *2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2015, pp. 1–5.

[6] T. L. Mien, "Liquid Level Control of Coupled-Tank System Using Fuzzy-Pid Controller," *Int. J. Eng. Res. Technol.*, vol. 6, no. 11, pp. 459–464, 2019, [Online]. Available: www.ijert.org.

[7] Z. Man, H. R. Wu, S. Liu, and X. Yu, "A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks," *IEEE Trans. Neural Networks*, vol. 17, no. 6, pp. 1580–1591, 2006.

[8] M. S. Aftab and M. Shafiq, "Neural networks for tracking of unknown SISO discrete-time nonlinear dynamic systems," *ISA Trans.*, vol. 59, pp. 363–374, 2015.

[9] U. M. Nath, C. Dey, and R. K. Mudi, "Designing of anti-windup feature for internal model controller with real-time performance evaluation on temperature control loop," *2019 2nd Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2019*, pp. 787–790, 2019, doi: 10.1109/ICICICT46008.2019.8993360.