

Optimized PID-Like Neural Network Controller for Single-Objective Systems

Gunawan Dewantoro, Johannes Nico Sukamto, Fransiscus Dalu Setiaji

Department of Electronic and Computer Engineering, Satya Wacana Christian University, Jl. Diponegoro 52-60, Salatiga 50711, Indonesia

ARTICLE INFO

Article history:

Received November 22, 2022

Revised December 15, 2022

Published December 23, 2022

Keywords:

Artificial neural network;

Intelligent controller;

PID;

Integral absolute error;

Optimization

ABSTRACT

The utilization of intelligent controllers becomes more prevalent as the hype of Industry 4.0 arises. Artificial neural network (ANN) exhibits the mapping ability and can estimate the output by means of either interpolation or extrapolation. These properties are sought to supersede the classical controllers. In this study, the ANN establishment was initiated by collecting dataset from the input and output of a well-known PID controller. The dataset was trained using a set of control factor combinations, including the number of neurons, the number of hidden layers, activation functions, and learning rates. Two kinds of ANN controllers were investigated, including one-input and three-input ANN. The testing was conducted under normal and uncertain conditions. These uncertainties include external disturbances, plant variations, and setpoint variations. The integral absolute error (IAE) was selected as the single objective to assess. The simulation results show that the response of three-input ANN controllers could yield smaller IAE at their best combinations under most kinds of conditions. Besides, the three-input ANN outperforms the one-input ANN both qualitatively and quantitatively. These facts might lead to a broader utilization of ANN as controllers.

This work is licensed under a [Creative Commons Attribution-Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



Corresponding Author:

Gunawan Dewantoro, Department of Electronic and Computer Engineering, Satya Wacana Christian University, Jl. Diponegoro 52-60, Salatiga 50711, Indonesia

Email: gunawan.dewantoro@uksw.edu

1. INTRODUCTION

The Industrial Revolution 4.0 and the artificial intelligence (AI) at its core are fundamentally changing the way we live, work and interact as citizens. The AI revolution is on target and will bring extensive changes affecting all aspects of our society and life [1][2][3]. Currently, AI (machine learning, neural network, deep learning, robotic), information security, big data, cloud computing, the internet, and forensic science are all hotspots and exciting topics of information and communication technology [4]. Artificial neural network (ANN) is an artificial representation of the human brain which always tries to simulate the learning process in the human brain. The term artificial is used because this neural network is implemented using a computer program that is able to complete several computation processes during the learning process. ANN has several uses in different areas, such as speech recognition, computer vision, face alignment, pattern recognition, detection, signal processing, prediction, and classification [5]. ANNs have been applied in many ways, including pattern recognition [6][7], power systems [8][9], robotics control [10][11][12], forecasting [13][14][15], smart manufacturing [16], social sciences [17], art [18], optimization [19], psychological sciences [20], education [21], etc. However, the ANN also faces several challenges, such as its susceptibility to overfitting, the presence of black box nature, and also long training process due to the large amount of data [22][23][24].

In control system area, there are several studies employing ANN as part of the controller. Various adaptation schemes are utilized to tune the PID parameters automatically. The Radial Basis Function (RBF) neural network dynamically regulates the parameters of PID controller and finally achieve the optimal PID

controller performance [25]. A Layer-Recurrent Network is trained on offline mode using a database constituted from the input/output signals of a PID-controlled closed loop system. The trained NN is implemented to add an auxiliary online control to improve the PID performance [26]. Three separate Multi-Layer Perceptron (MLP) are used to adjust each PID parameter, namely K_p , K_i , and K_d . The neurocontroller is placed in series to the plant and is able to maintain uniform response for various circumstances [27]. Not only can the ANN deal with a linear system but also numerous nonlinear systems. A feedforward neural network model employs a backpropagation algorithm to train its weights. After the network is trained, the PID parameters are adjusted to control nonlinear processes, and the stability condition must be met during the weight-updating process [28]. Also, a recurrent neural network is utilized for PID tuning tasks. The results demonstrate that the temporal information in the recurrent links can be abused by the network to improve the control performance of several nonlinear process benchmarks [29]. The implementation of ANN-based self-tuning PID control has been widely used in various applications, including load frequency control [30], underwater vehicles [31], light gasoline etherification [32], and so on. However, the works mentioned above require additional elements, either in series or parallel, to the PID controller during the operation. Thus, it requires longer computation time and also leads to the presence of transport delay which may degrade the stability [33][34].

Several attempts were made to combine the ANN and PID controller into a single subsystem. A study in [35] suggested a neural network consisting of up to three nodes in the hidden layer. This architecture makes the network work in the same way as that of P, PI, PD, or PID controller. Authors in [36] proposed a PID-like neural network adaptive controller (PIDNNC) for multivariable single-input-multi-output systems. The network is constructed using a mix locally recurrent neural network and able to meet the condition of closed-loop stability. In [37], ANN was utilized to resemble the PID formula and the differential evolution algorithm (DEA) to automatically adapt the weight of the ANN. In this study, ANN is trained with supervised learning using several training combinations [38]. The research contribution is that the operation no longer needs supplementary elements to resemble the PID controller. In addition, all three PID parameters are tuned simultaneously, not in a sequence, which saves computational time. The training combination includes the number of neurons (3, 5, and 7), the number of hidden layers (1, 2, and 3), activation functions (tansig and purelin), and learning rate (0.2 and 0.4). This training is conducted to obtain a reliable control scheme in both normal and uncertain conditions, which includes a reliable control scheme against external disturbances, limited variations in plant, and setpoint variations, with the Integral Absolute Error (IAE) [39] being the single objective.

2. METHODS

The experiment was carried out in three subsequent stages, namely model training, model testing, and evaluation, as explained in Fig. 1. In model training stage, a dataset derived from the input-output of the PID controller was employed to train the network. An optimization practice was conducted to better understand the best possible architecture. During the testing, IAE was calculated for both normal and perturbed conditions, followed by performance evaluations. All simulations are performed under Matlab/Simulink, running on Intel Core i7, RAM of 16 GB, and GPU accelerator RTX 3060.

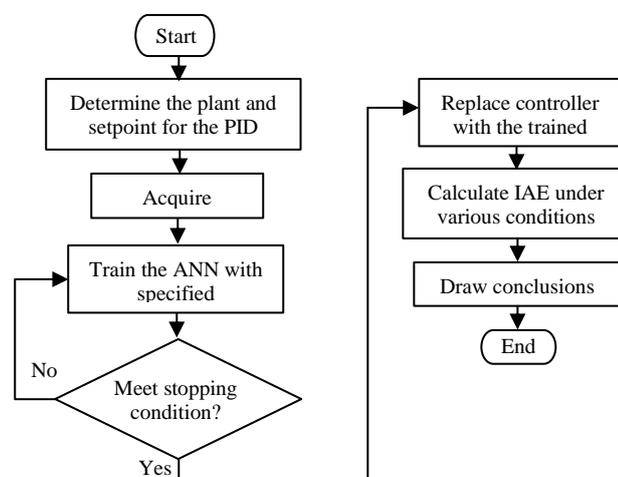


Fig. 1. Research stages

2.1. Training Setup

In this study, the plant is represented by a second-order DC motor whose transfer function between armature voltage and angular speed is given in [40]. The transfer function of the DC motor block is shown in (1).

$$T_f = \frac{\omega(s)}{V(s)} = \frac{K_m}{LJs^2 + (LB + RJ)s + (RB + K_m K_b)} \tag{1}$$

where J is moment of inertia (kgm²), B is viscous friction coefficient (Nms/rad), L is armature inductance (H), R is armature resistance (ohms), K_m is motor torque constant (Nm/A), K_b is electromotive force constant (Nms). The parameters of the plant are given in Table 1.

Table 1. Parameter Values

No	Parameter	Value
1	J	200
2	B	0.08
3	L	0.05
4	H	100
5	K_m	10
6	K_b	0.2

By using the parameter values in Table 1, the transfer function is obtained as follows:

$$T_f = \frac{10}{s^2 + 5s + 10} \tag{2}$$

The acquisition of dataset was made by recording the input and output of a PID controller. The PID parameters were determined by Matlab using the Ziegler-Nichols criterion, giving $K_p=2.34$; $K_i=5.25$; and $K_d=0.25$. In this study, the dataset was acquired from one-input and three-input PID blocks, as shown in Fig. 2.

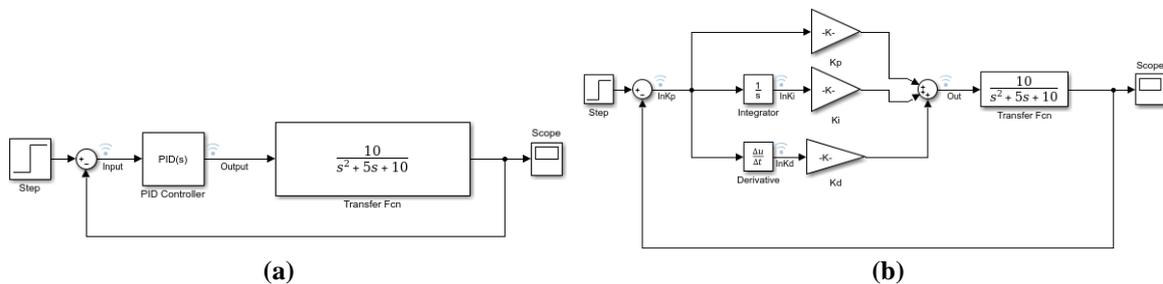


Fig. 2. Dataset acquisition of (a) one-input and (b) three-input PID controller

ANN requires a dataset for training taken from the PID input and output. By using logged signals data inspector as a viewing tool on Simulink, we are able to obtain the PID input and output values. The input and output dataset from the PID controller is then transposed. It is because the acquired dataset is in the form of a column matrix, while `nnTool` in MATLAB uses a row matrix format. The dataset size for one input is 1505, while the dataset size for three inputs is 3x61. Furthermore, the dataset is transferred to Ms. Excel to make it easier to conduct the ANN training. The ANN has several combinations, such as the number of neurons, the number of hidden layers, the activation function, and the learning rate. The ANN training is carried out in MATLAB/Simulink environment.

2.2. Testing Setup

The IAE is calculated under normal and uncertain conditions. As for normal condition, the testing diagrams are shown in Fig. 3 with a setpoint magnitude of 5. Meanwhile, the testing against a variety of uncertain conditions was undertaken as follows:

1. Reliable control schemes against external disturbance. This method is done by providing a signal interference in the form of steps with a magnitude of 5 - 10% of the setpoint, as shown in Fig. 4.

2. Limited variation on the plant. This method is done by changing the internal plant parameters i.e., the natural frequency increases by 22.5%, and the damping ratio increases by 63%, as shown in Fig. 5.
3. Setpoint variation. This method is done by changing the setpoint to a piece-wise step signal, as shown in Fig. 6. The simulation diagrams are shown in Fig. 7.

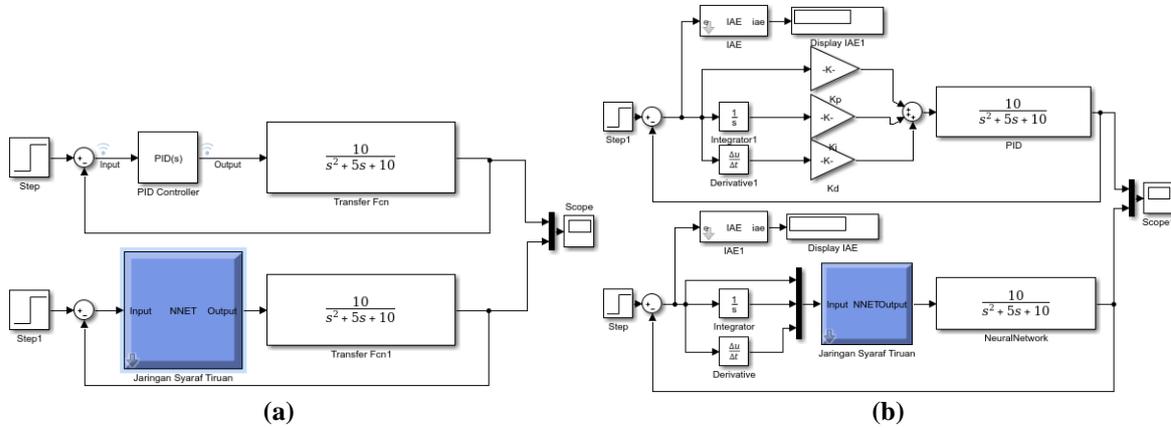


Fig. 3. Testing configuration of (a) one-input and (b) three-input ANN

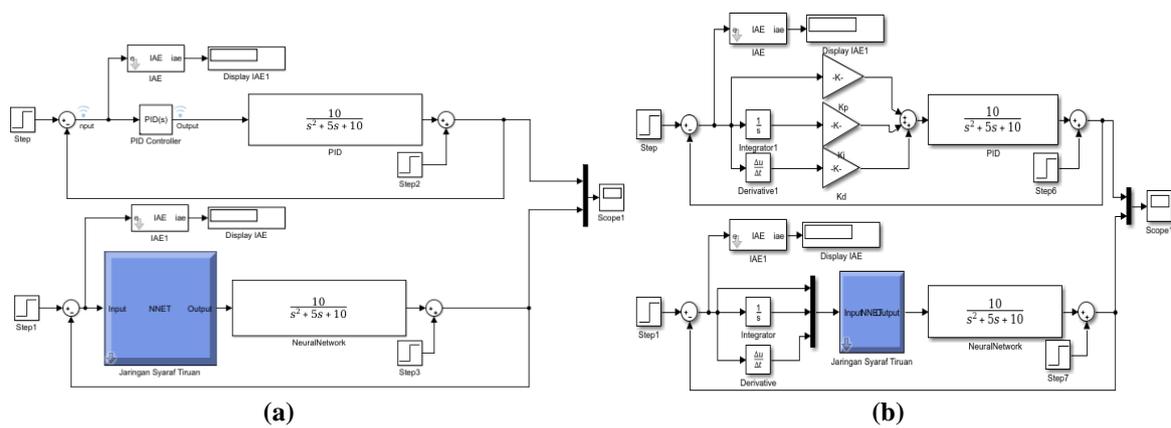


Fig. 4. Testing configuration of (a) one-input and (b) three-input ANN with external disturbance

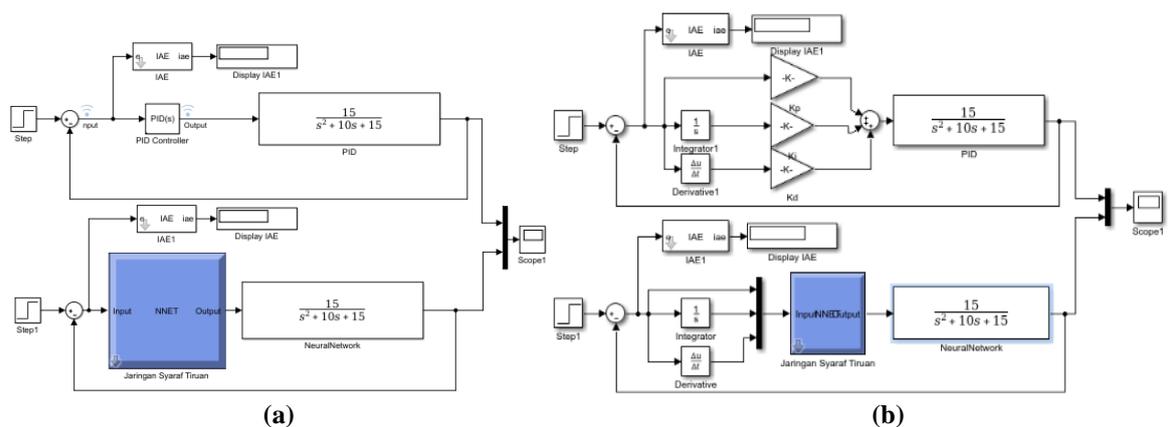


Fig. 5. Testing configuration of (a) one-input and (b) three-input ANN with plant variation

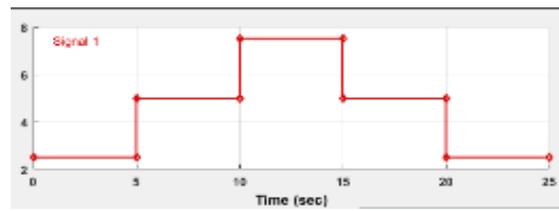


Fig. 6. The piece-wise step signal input

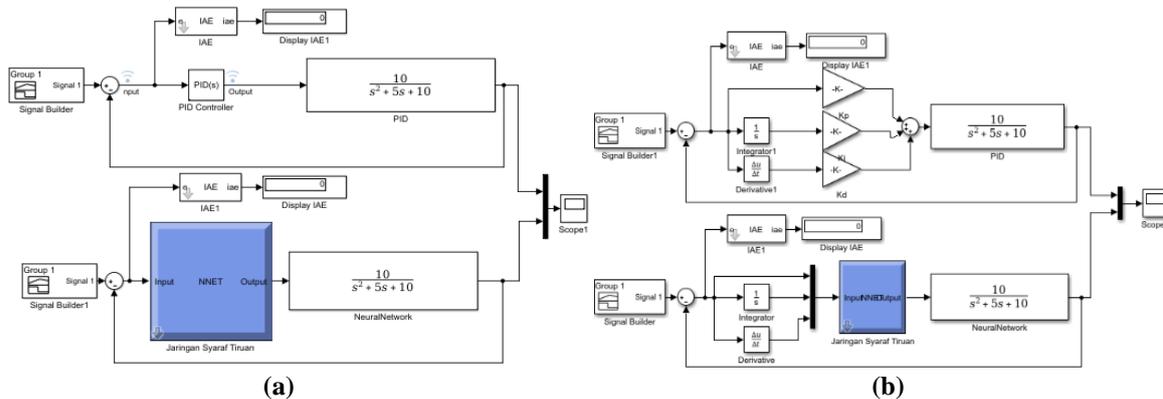


Fig. 7. Testing configuration of (a) one-input and (b) three-input ANN with setpoint variation

3. RESULTS AND DISCUSSION

3.1. Testing Results of One-Input ANN

This test uses 36 combinations, including the number of neurons (3, 5, and 7), the number of hidden layers (1, 2, and 3), the activation function (purelin-tansig and tansig-purelin), and the learning rate (0.2 and 0.4). The first experiment was carried out under normal condition. Table 2 shows the IAE values of various one-input ANN parameters and PID controller. To be more readable, all parameters are represented by letters i.e., parameter *A* denotes the number of neurons for each layer, *B* denotes the number of hidden layers, *C* denotes the activation function, and *D* denotes the learning rate. The row in red indicates the smallest IAE value obtained from the one-input ANN controller i.e., 1.258. Compared to each 36 combinations, the PID controller is still superior in terms of IAE. The best one-input ANN architecture yielding the smallest IAE for this case is as follows: 3 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.4. The one-input ANN training stops at the 14th epoch, and the smallest Mean Squared Error value of 0.16837 is found at the 8th epoch.

Table 2. IAE of One-Input ANN Under Normal Condition

One-Input Artificial Neural Network					PID	One-Input Artificial Neural Network					PID			
A	B	C	D	IAE		A	B	C	D	IAE				
3	1	tansig-purelin	0.2	16.89	1.24	5	7	2	purelin-tansig	0.2	7.53	1.24		
3	1	tansig-purelin	0.4	1.258	1.24	5	7	2	purelin-tansig	0.4	7.412	1.24		
3	1	purelin-tansig	0.2	7.499	1.24	7	7	2	tansig-purelin	0.2	2.357	1.24		
3	1	purelin-tansig	0.4	7.708	1.24	7	7	2	tansig-purelin	0.4	1.955	1.24		
5	1	tansig-purelin	0.2	2.07	1.24	7	7	2	purelin-tansig	0.2	2.15	1.24		
5	1	tansig-purelin	0.4	2.175	1.24	7	7	2	purelin-tansig	0.4	7.353	1.24		
5	1	purelin-tansig	0.2	6.941	1.24	3	5	7	3	tansig-purelin	0.2	2.078	1.24	
5	1	purelin-tansig	0.4	7.691	1.24	3	5	7	3	tansig-purelin	0.4	2.25	1.24	
7	1	tansig-purelin	0.2	2.1	1.24	3	5	7	3	purelin-tansig	0.2	1.979	1.24	
7	1	tansig-purelin	0.4	2.035	1.24	3	5	7	3	purelin-tansig	0.4	7.612	1.24	
7	1	purelin-tansig	0.2	7.334	1.24	5	5	7	3	tansig-purelin	0.2	1.882	1.24	
7	1	purelin-tansig	0.4	7.745	1.24	5	5	7	3	tansig-purelin	0.4	1.474	1.24	
3	7	2	tansig-purelin	0.2	2.09	1.24	5	5	7	3	purelin-tansig	0.2	7.576	1.24
3	7	2	tansig-purelin	0.4	1.657	1.24	5	5	7	3	purelin-tansig	0.4	6.995	1.24
3	7	2	purelin-tansig	0.2	7.682	1.24	7	5	7	3	tansig-purelin	0.2	2.217	1.24
3	7	2	purelin-tansig	0.4	7.252	1.24	7	5	7	3	tansig-purelin	0.4	1.292	1.24
5	7	2	tansig-purelin	0.2	2.949	1.24	7	5	7	3	purelin-tansig	0.2	6.138	1.24
5	7	2	tansig-purelin	0.4	4.223	1.24	7	5	7	3	purelin-tansig	0.4	7.109	1.24

Fig. 8 shows the step response comparison between PID and one-input ANN in normal condition. The red line represents one-input ANN, while the black line represents the PID controller. Based on the graph

measurement, the one-input ANN has more apparent oscillations and steady-state error compared to that of the PID controller. However, the rise time of the one-input ANN is 112.645 ms quicker than that of the PID controller.

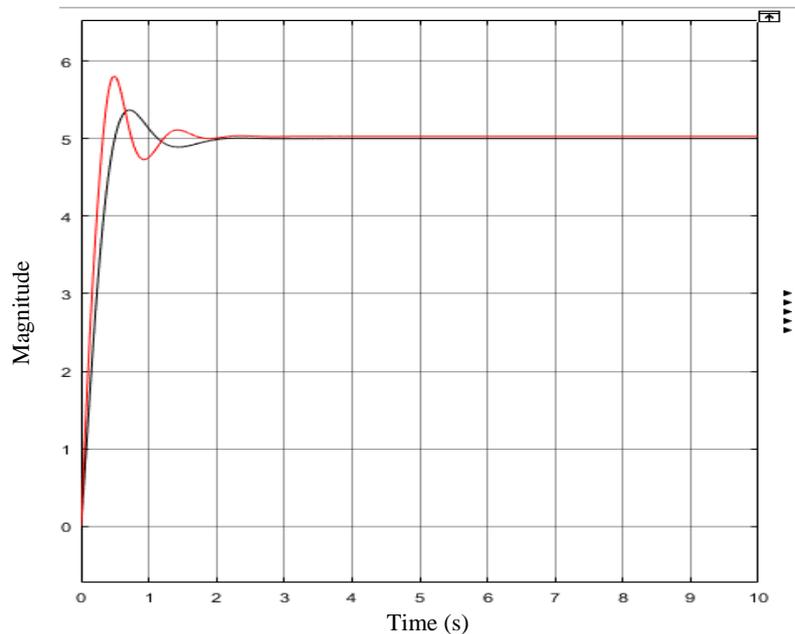


Fig. 8. Step response of one-input ANN in normal condition

As for the experimentation under an uncertain environment, Table 3 shows the IAE values of various one-input ANN parameters and PID controller with an external disturbance exerted at $t = 5$ s. The row in red indicates the smallest IAE value obtained from the ANN controller i.e., 1.756. Compared to each 36 combinations, the PID controller is still superior in terms of IAE. The best one-input ANN architecture yielding the smallest IAE for this case is as follows: 3 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.4. The one-input ANN training also stops at the 14th epoch, and the smallest Mean Squared Error value of 0.16837 is found at the 8th epoch.

Table 3. IAE of One-Input ANN Under With External Disturbance

One-Input Artificial Neural Network						One-Input Artificial Neural Network								
A	B	C	D	IAE	PID	A	B	C	D	IAE	PID			
3	1	tansig-purelin	0.2	18.85	1.364	5	7	2	purelin-tansig	0.2	5.03	1.364		
3	1	tansig-purelin	0.4	1.756	1.364	5	7	2	purelin-tansig	0.4	4.912	1.364		
3	1	purelin-tansig	0.2	4.949	1.364	7	7	2	tansig-purelin	0.2	3.023	1.364		
3	1	purelin-tansig	0.4	5.208	1.364	7	7	2	tansig-purelin	0.4	2.111	1.364		
5	1	tansig-purelin	0.2	2.833	1.364	7	7	2	purelin-tansig	0.2	2.77	1.364		
5	1	tansig-purelin	0.4	3.013	1.364	7	7	2	purelin-tansig	0.4	4.855	1.364		
5	1	purelin-tansig	0.2	4.441	1.364	3	5	7	3	tansig-purelin	0.2	3.077	1.364	
5	1	purelin-tansig	0.4	5.191	1.364	3	5	7	3	tansig-purelin	0.4	3.034	1.364	
7	1	tansig-purelin	0.2	2.936	1.364	3	5	7	3	purelin-tansig	0.2	2.922	1.364	
7	1	tansig-purelin	0.4	2.816	1.364	3	5	7	3	purelin-tansig	0.4	5.112	1.364	
7	1	purelin-tansig	0.2	4.834	1.364	5	5	7	3	tansig-purelin	0.2	2.972	1.364	
7	1	purelin-tansig	0.4	5.245	1.364	5	5	7	3	tansig-purelin	0.4	1.413	1.364	
3	7	2	tansig-purelin	0.2	2.92	1.364	5	5	7	3	purelin-tansig	0.2	5.076	1.364
3	7	2	tansig-purelin	0.4	2.102	1.364	5	5	7	3	purelin-tansig	0.4	4.495	1.364
3	7	2	purelin-tansig	0.2	5.182	1.364	7	5	7	3	tansig-purelin	0.2	2.962	1.364
3	7	2	purelin-tansig	0.4	4.752	1.364	7	5	7	3	tansig-purelin	0.4	1.785	1.364
5	7	2	tansig-purelin	0.2	3.404	1.364	7	5	7	3	purelin-tansig	0.2	5.123	1.364
5	7	2	tansig-purelin	0.4	3.915	1.364	7	5	7	3	purelin-tansig	0.4	4.609	1.364

Fig. 9 shows the step response comparison between PID and one-input ANN. The red line represents one-input ANN, while the black line represents the PID controller. Based on the graph measurement, the one-input ANN has more apparent oscillations and steady-state error compared to that of the PID controller before and after the exertion of disturbance. However, the rise time of the one-input ANN is 98.543 ms quicker than that of the PID controller.

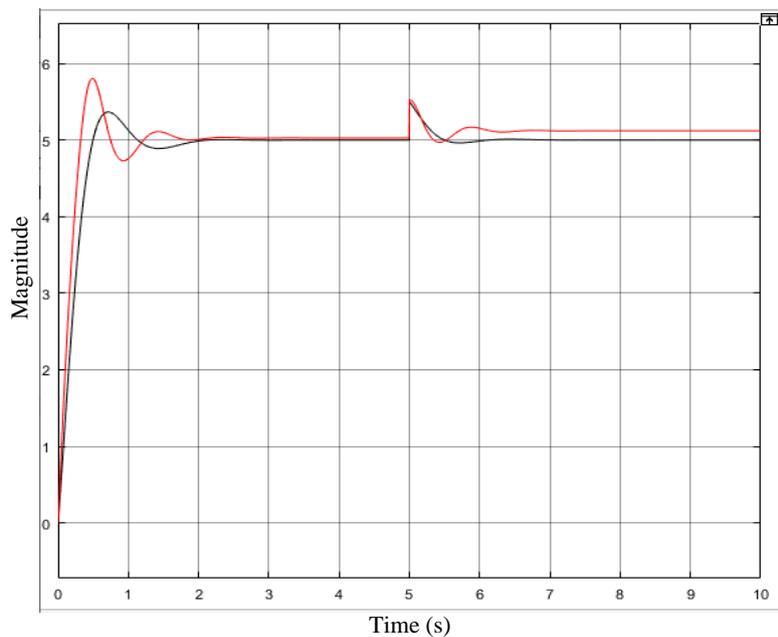


Fig. 9. Step response of one-input ANN with an external disturbance at $t = 5$ seconds

Then, the controllers were tested on a varied plant to demonstrate the sensitivity to internal parameter changes. Table 4 shows the IAE values of one-input ANN in various combinations and PID controller obtained from a varied plant. The row in bold red indicates the smallest IAE value obtained from the one-input ANN controller i.e., 1.141, while the row in bold black indicates that the IAE of the one-input ANN is smaller than that of the PID controller. It can be seen that there are two parameter combinations giving lower IAE than that of the PID controller. The best one-input ANN architecture yielding the smallest IAE for this case is as follows: 3 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.4. The one-input ANN training also stops at the 14th epoch and the smallest Mean Squared Error value of 0.16837 is found at the 8th epoch.

Table 4. IAE of One-Input ANN Obtained From A Varied Plant

One-Input Artificial Neural Network					PID	One-Input Artificial Neural Network					PID			
A	B	C	D	IAE		A	B	C	D	IAE				
3	1	tansig-purelin	0.2	16.11	1.351	5	7	2	purelin-tansig	0.2	8.48	1.351		
3	1	tansig-purelin	0.4	1.141	1.351	5	7	2	purelin-tansig	0.4	8.385	1.351		
3	1	purelin-tansig	0.2	8.414	1.351	7	7	2	tansig-purelin	0.2	2.554	1.351		
3	1	purelin-tansig	0.4	8.627	1.351	7	7	2	tansig-purelin	0.4	7.647	1.351		
5	1	tansig-purelin	0.2	2.252	1.351	7	7	2	purelin-tansig	0.2	3.775	1.351		
5	1	tansig-purelin	0.4	2.275	1.351	7	7	2	purelin-tansig	0.4	8.338	1.351		
5	1	purelin-tansig	0.2	7.999	1.351	3	5	7	3	tansig-purelin	0.2	2.033	1.351	
5	1	purelin-tansig	0.4	8.614	1.351	3	5	7	3	tansig-purelin	0.4	2.387	1.351	
7	1	tansig-purelin	0.2	2.198	1.351	3	5	7	3	purelin-tansig	0.2	2.123	1.351	
7	1	tansig-purelin	0.4	2.104	1.351	3	5	7	3	purelin-tansig	0.4	8.458	1.351	
7	1	purelin-tansig	0.2	8.32	1.351	5	5	7	3	tansig-purelin	0.2	1.947	1.351	
7	1	purelin-tansig	0.4	8.658	1.351	5	5	7	3	tansig-purelin	0.4	1.169	1.351	
3	7	2	tansig-purelin	0.2	2.229	1.351	5	5	7	3	purelin-tansig	0.2	8.518	1.351
3	7	2	tansig-purelin	0.4	1.749	1.351	5	5	7	3	purelin-tansig	0.4	8.039	1.351
3	7	2	purelin-tansig	0.2	8.606	1.351	7	5	7	3	tansig-purelin	0.2	8.948	1.351
3	7	2	purelin-tansig	0.4	8.254	1.351	7	5	7	3	tansig-purelin	0.4	6.07	1.351
5	7	2	tansig-purelin	0.2	3.142	1.351	7	5	7	3	purelin-tansig	0.2	4.19	1.351
5	7	2	tansig-purelin	0.4	4.178	1.351	7	5	7	3	purelin-tansig	0.4	7.984	1.351

Fig. 10 shows the step response comparison between PID and one-input ANN if the plant is changed. The red line represents one-input ANN, while the black line represents the PID controller. Based on the graph measurement, the rise time of the one-input ANN is 110.509 ms quicker, and the overshoot is 2.16% lower than that of the PID controller.

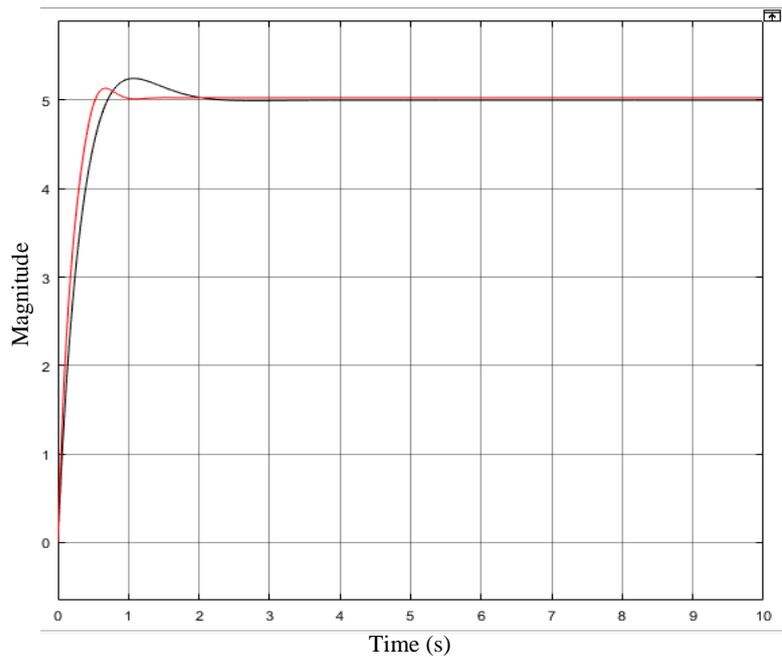


Fig. 10. Step response of one-input ANN for a varied plant

Finally, a piece-wise step input was applied to examine the ability of the controllers’ output to follow the trajectory. Table 5 shows the IAE values of various one-input ANN parameters and PID controller. The row in red indicates the smallest IAE value obtained from the one-input ANN controller i.e., 11.05. Compared to each 36 combinations, the PID controller is still superior in terms of IAE. The best one-input ANN architecture yielding the smallest IAE for this case is as follows: 3 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.4. The one-input ANN training also stops at the 14th epoch, and the smallest Mean Squared Error value of 0.16837 is found at the 8th epoch.

Table 5. IAE of One-Input ANN Obtained With A Varied Setpoint

One-Input Artificial Neural Network					PID	One-Input Artificial Neural Network					PID			
A	B	C	D	IAE		A	B	C	D	IAE				
3	1	tansig-purelin	0.2	51.37	3.1	5	7	2	purelin-tansig	0.2	40.17	3.1		
3	1	tansig-purelin	0.4	11.05	3.1	5	7	2	purelin-tansig	0.4	40.17	3.1		
3	1	purelin-tansig	0.2	40.17	3.1	7	7	2	tansig-purelin	0.2	16.13	3.1		
3	1	purelin-tansig	0.4	40.17	3.1	7	7	2	tansig-purelin	0.4	37.83	3.1		
5	1	tansig-purelin	0.2	19.22	3.1	7	7	2	purelin-tansig	0.2	35.03	3.1		
5	1	tansig-purelin	0.4	19.54	3.1	7	7	2	purelin-tansig	0.4	40.17	3.1		
5	1	purelin-tansig	0.2	40.17	3.1	3	5	7	tansig-purelin	0.2	26.23	3.1		
5	1	purelin-tansig	0.4	40.17	3.1	3	5	7	tansig-purelin	0.4	38.51	3.1		
7	1	tansig-purelin	0.2	18.55	3.1	3	5	7	purelin-tansig	0.2	28.47	3.1		
7	1	tansig-purelin	0.4	31.59	3.1	3	5	7	purelin-tansig	0.4	40.17	3.1		
7	1	purelin-tansig	0.2	40.17	3.1	5	5	7	tansig-purelin	0.2	17.55	3.1		
7	1	purelin-tansig	0.4	40.17	3.1	5	5	7	tansig-purelin	0.4	18.74	3.1		
3	7	2	tansig-purelin	0.2	18.24	3.1	5	5	7	3	purelin-tansig	0.2	40.17	3.1
3	7	2	tansig-purelin	0.4	13.58	3.1	5	5	7	3	purelin-tansig	0.4	14.26	3.1
3	7	2	purelin-tansig	0.2	40.17	3.1	7	5	7	3	tansig-purelin	0.2	20.52	3.1
3	7	2	purelin-tansig	0.4	12.67	3.1	7	5	7	3	tansig-purelin	0.4	30.77	3.1
5	7	2	tansig-purelin	0.2	17.21	3.1	7	5	7	3	purelin-tansig	0.2	37.89	3.1
5	7	2	tansig-purelin	0.4	19.57	3.1	7	5	7	3	purelin-tansig	0.4	40.19	3.1

Fig. 11 shows the response comparison between PID and one-input ANN with a varied setpoint. The red line represents one-input ANN, while the black line represents the PID controller. Based on the graph measurement, the ANN has more apparent oscillations and steady-state error compared to that of the PID controller for each step input. However, the rise time of the one-input ANN is 22.394 ms quicker than that of the PID controller.

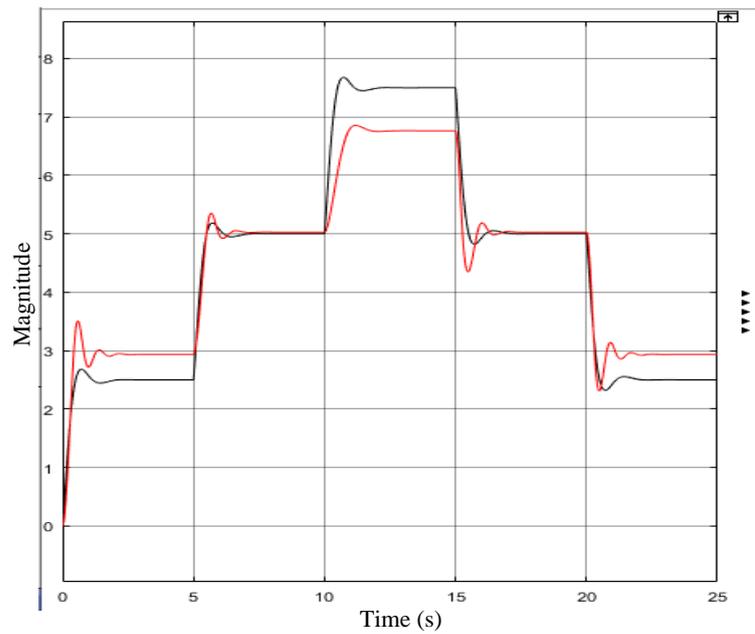


Fig. 11. Step response of one-input ANN with a varied setpoint

3.2. Testing Results of Three-Input ANN

Likewise, this test uses the same 36 combinations, including the number of neurons (3, 5, and 7), the number of hidden layers (1, 2, and 3), the activation function (purelin-tansig and tansig-purelin), and the learning rate (0.2 and 0.4). The first experiment was carried out under normal condition. Table 6 shows the IAE values of various three-input ANN parameters and PID controller. The row in red indicates the smallest IAE value obtained from the three-input ANN controller i.e., 1.712, while the row in bold black indicates that the IAE of the three-input ANN is smaller than that of the PID controller. It can be seen that there are two parameter combinations giving lower IAE than that of the PID controller. It can be seen that there are 12 parameter combinations giving lower IAE than that of the PID controller. The best three-input ANN architecture yielding the smallest IAE for this case is as follows: (3 7) neurons, 2 hidden layers, the activation function of tansig-purelin, and a learning rate of 0.2. The three-input ANN training stops at the 22nd epoch and the smallest MSE value of 5.4305×10^{-7} is found at the 16th epoch.

Table 6. IAE of Three-Input ANN Under Normal Condition

Three-Input Artificial Neural Network					PID		Three-Input Artificial Neural Network					PID	
A	B	C	D	IAE			A	B	C	D	IAE		
3	1	tansig-purelin	0.2	1.813	1.771	5	7	2	purelin-tansig	0.2	1.76	1.771	
3	1	tansig-purelin	0.4	1.769	1.771	5	7	2	purelin-tansig	0.4	1.825	1.771	
3	1	purelin-tansig	0.2	150.7	1.771	7	7	2	tansig-purelin	0.2	1.774	1.771	
3	1	purelin-tansig	0.4	157.1	1.771	7	7	2	tansig-purelin	0.4	1.732	1.771	
5	1	tansig-purelin	0.2	1.727	1.771	7	7	2	purelin-tansig	0.2	1.759	1.771	
5	1	tansig-purelin	0.4	1.759	1.771	7	7	2	purelin-tansig	0.4	6.989	1.771	
5	1	purelin-tansig	0.2	161.6	1.771	3	5	7	3	tansig-purelin	0.2	1.811	1.771
5	1	purelin-tansig	0.4	24.09	1.771	3	5	7	3	tansig-purelin	0.4	147	1.771
7	1	tansig-purelin	0.2	1.856	1.771	3	5	7	3	purelin-tansig	0.2	92.25	1.771
7	1	tansig-purelin	0.4	1.719	1.771	3	5	7	3	purelin-tansig	0.4	150.6	1.771
7	1	purelin-tansig	0.2	1.813	1.771	5	5	7	3	tansig-purelin	0.2	2.185	1.771
7	1	purelin-tansig	0.4	71.46	1.771	5	5	7	3	tansig-purelin	0.4	1.715	1.771
3	7	tansig-purelin	0.2	1.712	1.771	5	5	7	3	purelin-tansig	0.2	1.758	1.771
3	7	tansig-purelin	0.4	1.852	1.771	5	5	7	3	purelin-tansig	0.4	1.798	1.771
3	7	purelin-tansig	0.2	1.821	1.771	7	5	7	3	tansig-purelin	0.2	1.959	1.771
3	7	purelin-tansig	0.4	1.952	1.771	7	5	7	3	tansig-purelin	0.4	1.927	1.771
5	7	tansig-purelin	0.2	1.74	1.771	7	5	7	3	purelin-tansig	0.2	1.78	1.771
5	7	tansig-purelin	0.4	1.95	1.771	7	5	7	3	purelin-tansig	0.4	285.1	1.771

Fig. 12 shows the step response comparison between PID and three-input ANN. The red line represents three-input ANN, while the black line represents the PID controller. Based on the graph measurement, the rise time of the three-input ANN is 19.01 ms quicker and the overshoot is 1.361% lower than that of the PID controller.

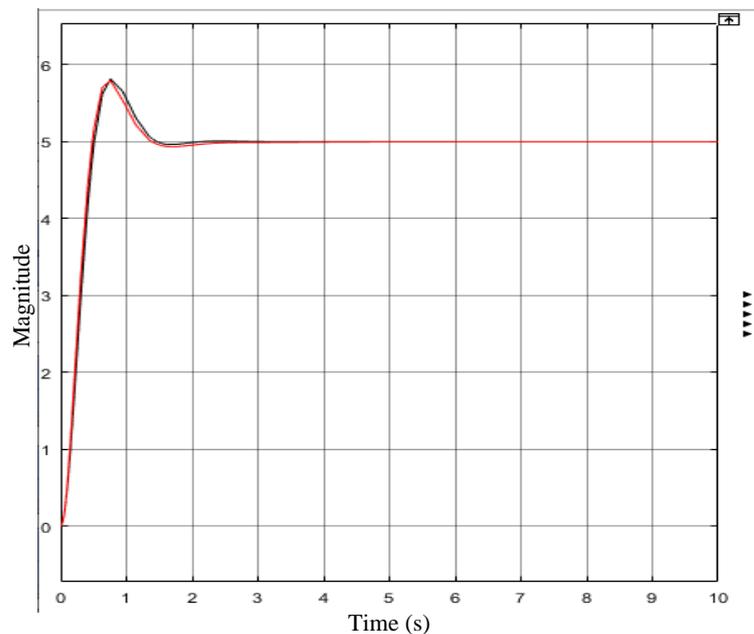


Fig. 12. Step response of three-input ANN under normal condition

As for the experimentation under an uncertain environment, Table 7 shows the IAE values of various three-input ANN parameters and PID controller with an external disturbance exerted at $t = 5$ s. The row in red indicates the smallest IAE value obtained from the three-input ANN controller i.e, 1.874, while the row in bold black indicates that the IAE of the three-input ANN is smaller than that of the PID controller. It can be seen that there are two parameter combinations giving lower IAE than that of the PID controller. It can be inferred that the best three-input ANN architecture yielding the smallest IAE for this case is as follows: 5 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.2. The ANN training also stops at the 8th epoch, and the smallest Mean Squared Error value of 0.060748 is found at the 2nd epoch.

Table 7. IAE of Three-Input ANN with External Disturbance

Three-Input Artificial Neural Network						PID	Three-Input Artificial Neural Network						PID					
A	B	C	D	IAE		A	B	C	D	IAE		A	B	C	D	IAE		
3	1	tansig-purelin	0.2	11.09	1.916	5	7	2	purelin-tansig	0.2	1.937	1.916						
3	1	tansig-purelin	0.4	1.948	1.916	5	7	2	purelin-tansig	0.4	2.546	1.916						
3	1	purelin-tansig	0.2	50.92	1.916	7	7	2	tansig-purelin	0.2	1.953	1.916						
3	1	purelin-tansig	0.4	57.26	1.916	7	7	2	tansig-purelin	0.4	1.932	1.916						
5	1	tansig-purelin	0.2	1.874	1.916	7	7	2	purelin-tansig	0.2	1.936	1.916						
5	1	tansig-purelin	0.4	1.939	1.916	7	7	2	purelin-tansig	0.4	9.432	1.916						
5	1	purelin-tansig	0.2	61.65	1.916	3	5	7	3	tansig-purelin	0.2	4.336	1.916					
5	1	purelin-tansig	0.4	7.148	1.916	3	5	7	3	tansig-purelin	0.4	37.97	1.916					
7	1	tansig-purelin	0.2	2.062	1.916	3	5	7	3	purelin-tansig	0.2	29.4	1.916					
7	1	tansig-purelin	0.4	1.909	1.916	3	5	7	3	purelin-tansig	0.4	51.26	1.916					
7	1	purelin-tansig	0.2	2.105	1.916	5	5	7	3	tansig-purelin	0.2	3.85	1.916					
7	1	purelin-tansig	0.4	6.167	1.916	5	5	7	3	tansig-purelin	0.4	1.886	1.916					
3	7	2	tansig-purelin	0.2	2.085	1.916	5	5	7	3	purelin-tansig	0.2	1.944	1.916				
3	7	2	tansig-purelin	0.4	28.41	1.916	5	5	7	3	purelin-tansig	0.4	5.915	1.916				
3	7	2	purelin-tansig	0.2	2.027	1.916	7	5	7	3	tansig-purelin	0.2	2.122	1.916				
3	7	2	purelin-tansig	0.4	2.127	1.916	7	5	7	3	tansig-purelin	0.4	6.264	1.916				
5	7	2	tansig-purelin	0.2	1.918	1.916	7	5	7	3	purelin-tansig	0.2	1.996	1.916				
5	7	2	tansig-purelin	0.4	22.31	1.916	7	5	7	3	purelin-tansig	0.4	215.3	1.916				

Fig. 13 shows the step response comparison between PID and three-input ANN. The red line represents three-input ANN, while the black line represents the PID controller. Based on the graph measurement, the rise time of the three-input ANN is 1.487 ms quicker and the overshoot is 1.33% lower than that of the PID controller.

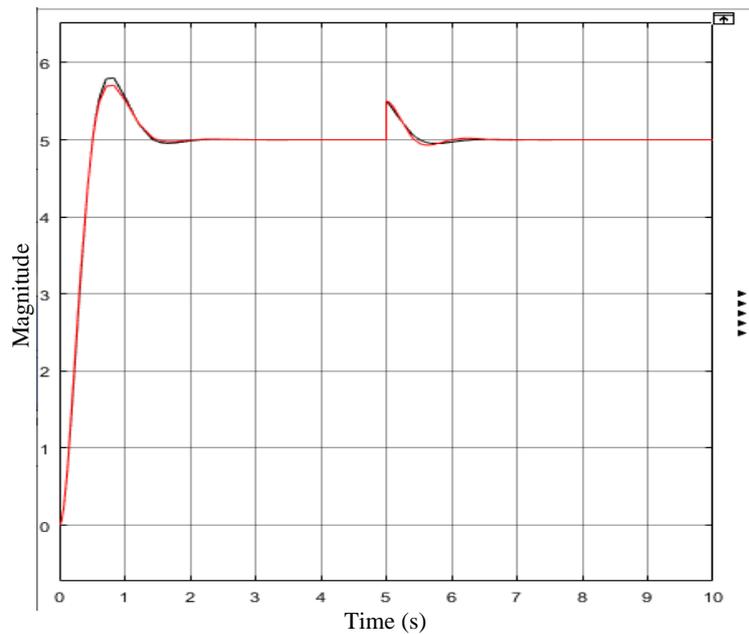


Fig. 13. Step response of three-input ANN with an external disturbance at $t = 5$ seconds

Then, the controllers were tested on a varied plant to demonstrate the sensitivity to internal parameter changes. Table 8 shows the IAE values of three-input ANN in various combinations and PID controller obtained from a varied plant. The row in bold red indicates the smallest IAE value obtained from the three-input ANN controller i.e., 1.366, while the row in bold black indicates that the IAE of the three-input ANN is smaller than that of the PID controller. It can be seen that there are 16 parameter combinations giving lower IAE than that of the PID controller. The best three-input ANN architecture yielding the smallest IAE for this case is as follows: (3 7) neurons, 2 hidden layers, the activation function of tansig-purelin, and a learning rate of 0.2. The three-input ANN training also stops at the 22nd epoch and the smallest Mean Squared Error value of 5.4305×10^{-7} is found at the 16th epoch.

Table 8. IAE of Three-Input ANN Obtained from A Varied Plant

Three-Input Artificial Neural Network					PID		Three-Input Artificial Neural Network					PID	
A	B	C	D	IAE			A	B	C	D	IAE		
3	1	tansig-purelin	0.2	1.828	1.885	5 7	2	purelin-tansig	0.2	1.878	1.885		
3	1	tansig-purelin	0.4	1.883	1.885	5 7	2	purelin-tansig	0.4	1.896	1.885		
3	1	purelin-tansig	0.2	9.765	1.885	7 7	2	tansig-purelin	0.2	1.871	1.885		
3	1	purelin-tansig	0.4	9.913	1.885	7 7	2	tansig-purelin	0.4	1.975	1.885		
5	1	tansig-purelin	0.2	1.736	1.885	7 7	2	purelin-tansig	0.2	1.88	1.885		
5	1	tansig-purelin	0.4	1.892	1.885	7 7	2	purelin-tansig	0.4	4.169	1.885		
5	1	purelin-tansig	0.2	9.793	1.885	3 5 7	3	tansig-purelin	0.2	11.47	1.885		
5	1	purelin-tansig	0.4	10.06	1.885	3 5 7	3	tansig-purelin	0.4	7.579	1.885		
7	1	tansig-purelin	0.2	5.988	1.885	3 5 7	3	purelin-tansig	0.2	2.497	1.885		
7	1	tansig-purelin	0.4	1.996	1.885	3 5 7	3	purelin-tansig	0.4	1.926	1.885		
7	1	purelin-tansig	0.2	1.912	1.885	5 5 7	3	tansig-purelin	0.2	2.845	1.885		
7	1	purelin-tansig	0.4	4.516	1.885	5 5 7	3	tansig-purelin	0.4	1.679	1.885		
3 7	2	tansig-purelin	0.2	1.366	1.885	5 5 7	3	purelin-tansig	0.2	1.86	1.885		
3 7	2	tansig-purelin	0.4	1.806	1.885	5 5 7	3	purelin-tansig	0.4	1.8	1.885		
3 7	2	purelin-tansig	0.2	1.838	1.885	7 5 7	3	tansig-purelin	0.2	2.845	1.885		
3 7	2	purelin-tansig	0.4	1.707	1.885	7 5 7	3	tansig-purelin	0.4	1.893	1.885		
5 7	2	tansig-purelin	0.2	1.797	1.885	7 5 7	3	purelin-tansig	0.2	1.867	1.885		
5 7	2	tansig-purelin	0.4	1.633	1.885	7 5 7	3	purelin-tansig	0.4	6.955	1.885		

Fig. 14 shows the step response comparison between PID and three-input ANN if the plant is changed. The red line represents three-input ANN, while the black line represents the PID controller. Based on the graph measurement, the rise time of the three-input ANN is 66.548 ms quicker, and the overshoot is 5.819% lower than that of the PID controller.

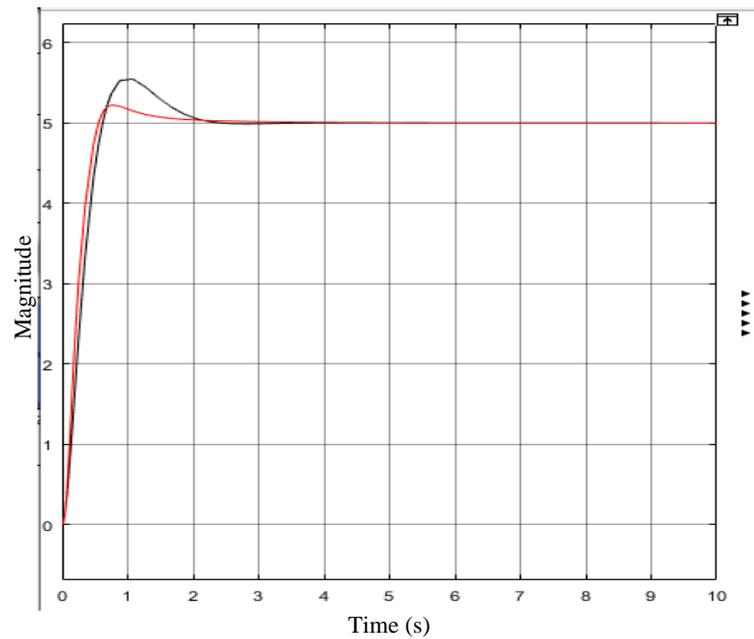


Fig. 14. Step response of three-input ANN for A Varied Plant

Finally, a piece-wise step input was applied to examine the ability of the controllers’ output to follow the trajectory. Table 9 shows the IAE values of various three-input ANN parameters and PID controller. The row in red indicates the smallest IAE value obtained from the three-input ANN controller i.e., 4.472. Compared to each 36 combinations, the PID controller is still superior in terms of IAE. It can be inferred that the best three-input ANN architecture yielding the smallest IAE for this case is as follows: 3 neurons, 1 hidden layer, the activation function of tansig-purelin, and a learning rate of 0.4. The three-input ANN training also stops at the 139th epoch, and the smallest Mean Squared Error value of 4.1864×10^{-7} is found at the 133rd epoch.

Table 9. IAE of Three-Input ANN with A Varied Setpoint

Three-Input Artificial Neural Network					PID	Three-Input Artificial Neural Network					PID
A	B	C	D	IAE		A	B	C	D	IAE	
3	1	tansig-purelin	0.2	16.99	4.332	5 7	2	purelin-tansig	0.2	40.17	4.332
3	1	tansig-purelin	0.4	4.476	4.332	5 7	2	purelin-tansig	0.4	40.17	4.332
3	1	purelin-tansig	0.2	177.6	4.332	7 7	2	tansig-purelin	0.2	16.13	4.332
3	1	purelin-tansig	0.4	178.1	4.332	7 7	2	tansig-purelin	0.4	37.83	4.332
5	1	tansig-purelin	0.2	5.961	4.332	7 7	2	purelin-tansig	0.2	35.03	4.332
5	1	tansig-purelin	0.4	4.769	4.332	7 7	2	purelin-tansig	0.4	40.17	4.332
5	1	purelin-tansig	0.2	178.1	4.332	3 5 7	3	tansig-purelin	0.2	26.23	4.332
5	1	purelin-tansig	0.4	178.1	4.332	3 5 7	3	tansig-purelin	0.4	38.51	4.332
7	1	tansig-purelin	0.2	91.35	4.332	3 5 7	3	purelin-tansig	0.2	28.47	4.332
7	1	tansig-purelin	0.4	6.367	4.332	3 5 7	3	purelin-tansig	0.4	40.17	4.332
7	1	purelin-tansig	0.2	24.64	4.332	5 5 7	3	tansig-purelin	0.2	17.55	4.332
7	1	purelin-tansig	0.4	152.4	4.332	5 5 7	3	tansig-purelin	0.4	18.74	4.332
3 7	2	tansig-purelin	0.2	9.593	4.332	5 5 7	3	purelin-tansig	0.2	40.17	4.332
3 7	2	tansig-purelin	0.4	166.9	4.332	5 5 7	3	purelin-tansig	0.4	14.26	4.332
3 7	2	purelin-tansig	0.2	77.53	4.332	7 5 7	3	tansig-purelin	0.2	20.52	4.332
3 7	2	purelin-tansig	0.4	12.67	4.332	7 5 7	3	tansig-purelin	0.4	30.77	4.332
5 7	2	tansig-purelin	0.2	17.21	4.332	7 5 7	3	purelin-tansig	0.2	37.89	4.332
5 7	2	tansig-purelin	0.4	19.57	4.332	7 5 7	3	purelin-tansig	0.4	40.19	4.332

Fig. 15 shows the response comparison between PID and three-input ANN with a varied setpoint. The red line represents three-input ANN, while the black line represents the PID controller. Based on the graph measurement, the three-input ANN has more apparent oscillations and steady-state error compared to that of the PID controller for each step input. However, the rise time of the three-input ANN is 3.478 ms quicker than that of the PID controller.

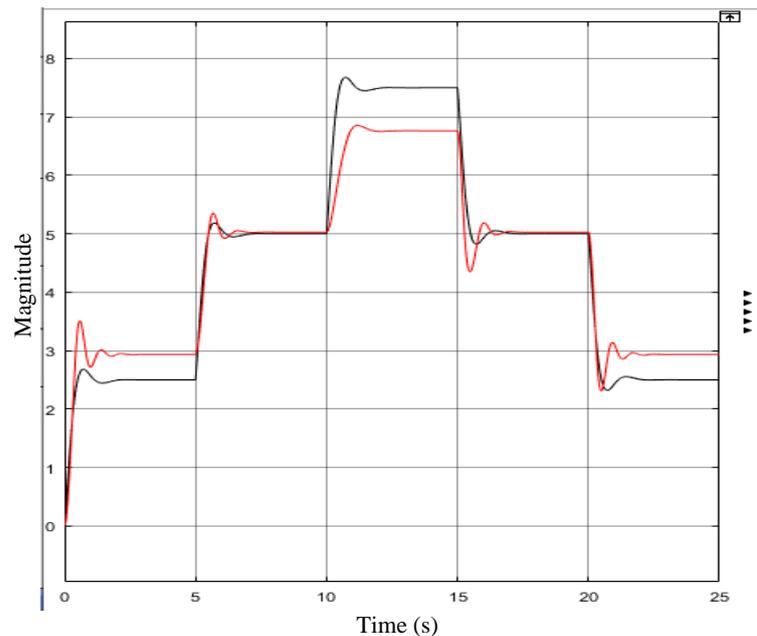


Fig. 15. Output comparison of three-input ANN with a varied setpoint

3.3. Discussions

The three-input ANN provides a smaller IAE than that of the one-input ANN for either normal condition or perturbed condition. The three-input ANN is more capable of mapping the nonlinear relationship between the input and output of the PID controller which contain integral and derivative terms. As such, the three-input outperforms the one-input ANN under all testing scenarios. With regard to computation time, the number of hidden layers and neurons plays a significant effect for one-input and three-input ANN. The deeper the network, the longer the elapsed time. In average, the one-input ANN requires 4.704 seconds during simulation, as opposed to 9.835 seconds for three-input ANN.

For the one-input ANN, the best training combination is 1 hidden layer, 3 neurons per layer, the activation function of tansig-purelin, and a learning rate of 0.4. Under the normal condition, the IAE value of the PID controller is still smaller than that of the one-input ANN with a difference of 0.014. In addition, the reliability against uncertain conditions such as external disturbances and setpoint variation of the PID controller is also better than that of the one-input ANN with an IAE difference of 0.392 and 7.95, respectively. However, when tested with limited variations of the plant, the IAE of one-input ANN is better than that of the PID controller, with a difference of 0.210. With regard to the time response, one-input ANN is able to improve the rise time for all test scenarios.

As for the three-input ANN, the best training combination depends on the test scenario under consideration. However, there is one best combination that gives lower IAE for most scenario tests, namely 1 hidden layer, 5 neurons per layer, the activation function of tansig-purelin, and a learning rate of 0.2. Under the normal condition, the IAE value of the three-input ANN is better than that of the PID controller, with a difference of 0.044. In addition, the reliability against uncertain conditions such as external disturbances and limited variations of the three-input controller is also superior to that of the PID controller than the IAE value of the ANN with a difference of 0.042 and 0.149, respectively. However, when tested with setpoint variation, the IAE value of the three-input ANN is higher than that of the PID controller with a difference of 1.629. With regard to the time response, three-input ANN is able to improve the rise time and overshoot percentage for most cases. The error value depends on the training iteration instead of the learning rate. As for the activation function, the tansig-purelin gives the best performance for all testing scenarios.

In comparison with the relevant previous work, this research has successfully condensed the neural network architecture for controlling a linear process. The recent work in [41] employs three inputs to yield three recommended PID parameters. In contrast, our proposed scheme generates only one output representing the PID-like control scheme. This simplification is somewhat significant in reducing the propagation delay since the ANN output is directly coupled to the plant without passing through any elements in between. In addition, during its operation, our ANN controller is adequate to regulate the whole process. This leads to a more straightforward mechanism, as opposed to ANN-based self-tuned PID controller, which requires both PID controller and ANN simultaneously [42].

4. CONCLUSION

The ANN based controller has been investigated to seek the possibility of replacing the well-established PID controller. The three-input ANN outperforms the one-input ANN in terms of IAE for either normal condition or uncertain condition. Compared to the PID controllers, the one-input ANN is not able to surpass the PID controller. Meanwhile, the three-input ANN is able to give smaller IAE, improved rise time, and lower overshoot compared to that of the PID controller under normal condition and uncertain condition, except for varied setpoint. The parameter combination of ANN plays a critical role in yielding a superior outcome. In the future, unsupervised learning can be introduced to omit the dataset acquisition prior to the training.

REFERENCES

- [1] S. Makridakis, "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms," *Futures*, vol. 90, pp. 46–60, 2017, <https://doi.org/10.1016/j.futures.2017.03.006>.
- [2] V. Dignum, "AI is multidisciplinary," *AI Matters*, vol. 5, no. 4, pp. 18–21, 2019, <https://doi.org/10.1016/j.ijinfomgt.2019.08.002>.
- [3] L. Floridi and J. Cowls, "A unified framework of five principles for AI in society," *Harv Data Sci Rev*, pp. 535–545, 2019, <https://doi.org/10.1162/99608f92.8cd550d1>.
- [4] A. Bécue, I. Praça and J. Gama, "Artificial intelligence, cyber-threats and Industry 4.0: challenges and opportunities," *Artif Intell Rev*, vol. 54, no. 5, pp. 3849–3886, 2021, <https://doi.org/10.1007/s10462-020-09942-2>.
- [5] O. I. Abiodun *et al.*, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, pp. 1–41, 2018, <https://doi.org/10.1016/j.heliyon.2018.e00938>.
- [6] O. I. Abiodun *et al.*, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158820–158846, 2019, <https://doi.org/10.1109/ACCESS.2019.2945545>.
- [7] A. Addeh, A. Khormali and N. A. Golilarz, "Control chart pattern recognition using RBF neural network with new training algorithm and practical features," *ISA Trans*, vol. 79, pp. 202–216, 2018, <https://doi.org/10.1016/j.isatra.2018.04.020>.
- [8] S. R. Khuntia, J. L. Rueda and M. A. M. M. van der Meijden, "Forecasting the load of electrical power systems in mid- and long-term horizons: a review," *IET Generation, Transmission, & Distribution*, vol. 10, no. 16, pp. 3971–2977, 2016, <https://doi.org/10.1049/iet-gtd.2016.0340>.
- [9] A. Ahmed and M. Khalid, "A review on the selected applications of forecasting models in renewable power systems," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 9–21, 2019, <https://doi.org/10.1016/j.rser.2018.09.046>.
- [10] R. Gao, "Inverse kinematics solution of robotics based on neural network algorithms," *J Ambient Intell Humaniz Comput*, vol. 11, pp. 6199–6209, 2020, <https://doi.org/10.1007/s12652-020-01815-4>.
- [11] T. G. Thuruthel, Y. Ansari, E. Falotico and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robot*, vol. 5, no. 2, pp. 149–163, 2018, <https://doi.org/10.1089/soro.2017.0007>.
- [12] Z. Bing, C. Meschede, F. Röhrbein, K. Huang and A. C. Knoll, "A survey of robotics control based on learning-inspired spiking neural networks," *Front Neurobot*, vol. 12, 2018, <https://doi.org/10.3389/fnbot.2018.00035>.
- [13] A. Tealab, H. Hefny and A. Badr, "Forecasting of nonlinear time series using ANN," *Future Computing and Informatics Journal*, vol. 2, no. 1, pp. 39–47, 2017, <https://doi.org/10.1016/j.fcij.2017.05.001>.
- [14] P. M. Datilo, Z. Ismail and J. Dare, "A review of epidemic forecasting using Artificial Neural Networks," *Int J Epidemiol Res*, vol. 6, no. 3, pp. 132–143, 2019, <https://doi.org/10.15171/ijer.2019.24>.
- [15] D. M. Ahmed, M. M. Hassan and R. J. Mstafa, "A review on Deep Sequential Models for forecasting time series data," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–19, 2022, <https://doi.org/10.1155/2022/6596397>.
- [16] J. Wang, Y. Ma, L. Zhang, R. X. Gao and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J Manuf. Syst.*, vol. 48, no. Part C, pp. 144–156, 2018, <https://doi.org/10.1016/j.jmsy.2018.01.003>.
- [17] D. Strohmaier, "Ontology, neural networks, and the social sciences," *Synthese*, vol. 199, pp. 4775–4794, 2021, <https://doi.org/10.1007/s11229-020-03002-6>.
- [18] Y. Hong and J. Kim, "Art painting identification using convolutional neural network," *International Journal of Applied Engineering Research*, vol. 12, no. 4, pp. 532–539, 2017, https://www.ripublication.com/ijaer17/ijaerv12n4_17.pdf.
- [19] M. J. Kim, T. S. Kim, R. J. Flores and J. Brouwer, "Neural-network-based optimization for economic dispatch of combined heat and power systems," *Applied Energy*, vol. 265, pp. 1–18, 2020, <https://doi.org/10.1016/j.apenergy.2020.114785>.
- [20] S. J. Read and L. C. Miller, "Neural network models of personality structure and dynamics," in *Measuring and Modeling Persons and Situations*, pp. 499–538, 2021, <https://doi.org/10.1016/B978-0-12-819200-9.00004-1>.
- [21] M. S. Hasibuan, L. E. Nugroho and P. I. Santosa, "Model detecting learning styles with artificial neural network," *Journal of Technology and Science Education*, vol. 9, no. 1, pp. 85–95, 2019, <https://doi.org/10.3926/jotse.540>.
- [22] J. Chu, X. Liu, Z. Zhang, Y. Zhang and M. He, "A novel method overcoming overfitting of artificial neural network for accurate prediction: Application on thermophysical property of natural gas," *Case Studies in Thermal Engineering*, vol. 28, p. 101406, 2021, <https://doi.org/10.1016/j.csite.2021.101406>.

- [23] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, 2019, <https://doi.org/10.1088/1742-6596/1168/2/022022>.
- [24] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks," in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 173–177, 2017, <https://doi.org/10.1109/INTELCIS.2017.8260032>.
- [25] N. C. Danh, "Neural networks application for the data of PID controller for Acrobot," *The Scientific World Journal*, vol. 2022, pp. 1–13, 2022, <https://doi.org/10.1155/2022/9162352>.
- [26] S. N. Larbi, N. Bourahla, H. Benchoubane and K. Choutri, "Developing a neural network algorithm as an additional online controller to the PID controller," in *16th World Conference on Earthquake*, pp. 1–6, 2017, <https://www.wcee.nicee.org/wcee/article/16WCEE/WCEE2017-975.pdf>.
- [27] R. Patel and V. Kumar, "Multilayer neuro PID controller based on back propagation algorithm," *Procedia Computer Science*, vol. 54, pp. 207–214, 2015, <https://doi.org/10.1016/j.procs.2015.06.023>.
- [28] A. Zribi, M. Chtourou and M. Djemel, "A new PID neural network controller design for nonlinear processes," *Journal of Circuits, Systems, and Computers*, vol. 27, no. 4, pp. 1–11, 2015, <https://doi.org/10.1142/S0218126618500652>.
- [29] E. Reichensdörfer, J. Günther and K. Diepold, "Recurrent neural networks for PID auto-tuning," *München*, Sep. 2017, <https://mediatum.ub.tum.de/doc/1381851/534530033346.pdf>.
- [30] M. K. Debnath, R. Agrawal, S. R. Tripathy and S. Choudhury, "Artificial neural network tuned PID controller for LFC investigation including distributed generation," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 33, no. 5, 2020, <https://doi.org/10.1002/jnm.2740>.
- [31] R. Hernández-Alvarado, L. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa and F. Fonseca-Navarro, "Neural network-based self-tuning PID control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, 2016, <https://doi.org/10.3390/s16091429>.
- [32] H. Cheng, Y. Zhang, L. Kong and X. Meng, "The application of neural network PID controller to control the light gasoline etherification," *IOP Conference Series: Earth and Environmental Science*, p. 012045, 2017, <https://doi.org/10.1088/1755-1315/69/1/012045>.
- [33] Y. Li, K. H. Ang and G. C. Y. Chong, "PID control system analysis and design: Problems, remedies, and future directions," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 32–41, 2006, <https://doi.org/10.1109/MCS.2006.1580152>.
- [34] D. Copot, M. Ghita and C. Ionescu, "Simple alternatives to PID-type control for processes with variable time-delay," *Processes*, vol. 7, no. 3, p. 146, 2019, <https://doi.org/10.3390/pr7030146>.
- [35] S. Cong, G. Li and B. Ji, "A novel PID-like neural network controller," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 121–126, 2005, <https://doi.org/10.3182/20050703-6-CZ-1902.00241>.
- [36] S. Cong and Y. Liang, "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3872–3879, 2009, <https://doi.org/10.1109/TIE.2009.2018433>.
- [37] W. Lu, J. Yang and X. Liu, "The PID controller based on the artificial neural network and the differential evolution algorithm," *J Comput (Taipei)*, vol. 7, no. 10, pp. 2368–2375, 2012, <https://doi.org/10.4304/jcp.7.10.2368-2375>.
- [38] S. C. Smithson, G. Yang, W. J. Gross and B. H. Meyer, "Neural networks designing neural networks," in *Proceedings of the 35th International Conference on Computer-Aided Design*, pp. 1–8, 2016, <https://doi.org/10.1145/2966986.2967058>.
- [39] K. Soltesz and A. Cervin, "When is PID a good choice?," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 250–255, 2018, <https://doi.org/10.1016/j.ifacol.2018.06.074>.
- [40] L. Mikova, I. Virgala and M. Klemen, "Speed control of DC motor," *American Journal of Mechanical Engineering*, vol. 4, no. 7, pp. 380–384, 2016, <https://doi.org/10.12691/ajme-4-7-27>.
- [41] Yong-Seok Lee and Dong-Won Jang, "Optimization of neural network-based self-tuning PID controllers for second order mechanical systems," *Applied Sciences*, vol. 11, no. 8002, pp. 1–12, 2021, <https://doi.org/10.3390/app11178002>.
- [42] M. Ma, "Research on parameter self-tuning PID control algorithm based on BP neural network," in *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp. 1215–1220, 2022, <https://doi.org/10.1109/ElConRus54750.2022.9755484>.