Quadrotor Altitude Control using Recurrent Neural Network PID

Faisal Fajri Rahani, Phisca Aditya Rosyady
Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

ARTICLE INFORMATION

Article History:

Submitted 24 May 2023
Revised 02 July 2023
Accepted 04 July 2023

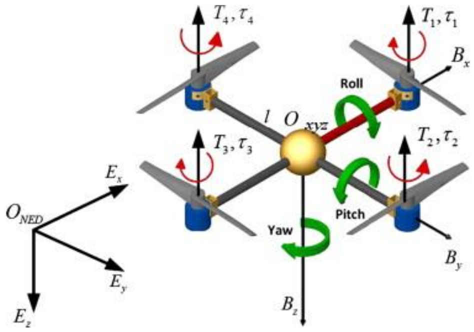
Keywords:

UAV;
RNN;
PID;
Quadrotor

Corresponding Author:

Faisal Fajri Rahani,
Department of Informatics,
Universitas Ahmad Dahlan,
Yogyakarta, Indonesia.
Email:
faisal.fajri@tif.uad.ac.id

ABSTRACT



Quadrotor is a type of Unmanned Aerial Vehicle (UAV) or unmanned flying vehicle. The Quadrotor can be operated by remote control or independently. Quadrotor control is a challenging problem because it takes into account complex things such as parametric protection, external interference, and so on. Conventional controls for quadrotor are widely used such as PID, state feedback, and so on. However, because the control is linear, then nonlinear control began to be developed. This paper will use PID control and a recurrent neural network for a quadrotor directional control system. The neural network system used has 3 layers for each network. The architecture used is 2 nodes in the input layer, 3 nodes in the hidden layer, and 1 node in the output layer. The input from this network is the error value from the system. The results of this control test show that the combination of PID and RNN for altitude control has a faster response time of 0.21 seconds, a smaller Steady state error value of 0.008 seconds and a faster settling time of 0.36 seconds than PID fine tuning. This shows that the response of this system is better than conventional PID.

Document Citation:

F. F. Rahani and P. A. Rosyady, “Quadrotor Altitude Control using Recurrent Neural Network PID,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 2, pp. 279-290, 2023, DOI: 10.12928/biste.v5i2.8455

1. INTRODUCTION

Quadorotor is a type of Unmanned Aerial Vehicle (UAV) or known as an unmanned flying vehicle [1]. In recent years, the quadrotor has attracted a lot of attention from many researchers around the world, due to the fact that it is inexpensive and has excellent maneuverability [2]. The Quadrotor can be operated with a remote controller and a standalone or autonomous program control device. This can enable the quadrotor to perform autonomous flight in unmanned mode [3]. This quadrotor type UAV vehicle has many advantages such as high flexibility, can fly autonomously, very small runway requirements, simple maintenance and low maintenance costs [4]. This Quadrotor is widely used in various fields such as: military reconnaissance, agricultural monitoring, civil aerial photography, and so on [3]. Even now quadrotor can complete several types of dangerous and difficult missions such as search and rescue missions, surveillance, inspection, mapping and aerial cinematography [5].

Quadrotor control is a challenging problem because it considers various complex issues such as parametric uncertainty, external disturbances, motor failures, and so on [6]. At the spatial level, three linear degrees of freedom along the three axes and three rotating degrees of freedom along the three axes are used as the six degrees of freedom for a quadrotor. Meanwhile, attitude angles play an important role in changing position coordinates which have a direct effect on control [3].

In recent years, there has been an emerging trend of renewal in control theory. Researchers have devised a number of theoretical algorithms for controlling the quadrotor. Conventional control for quadrotor that has been widely used such as PID [7], state feedback control system [8], etc. However, due to the linear nature of the control, many non-linear controls have been developed, for example using an adaptive sliding mode control system [9] fuzzy method [10] and control by combining a liner control system with an artificial intelligence system [8].

One of the controls that has been widely used in the industrial world is PID control [11][12]. The performance of conventional PID controllers fails in non-linear and uncertain applications [8] while almost all processes in quadrotor control are non-linear and uncertain.This kind of application requires real time adaptation of online control.This causes PID control in such situations to be impossible to generate control conventional tuning methods such as Zeigler-Nichols cannot be applied to non-linear, and uncertain robotic controllers [13]. In addition, the 2 degrees of freedom (2-DOF) robotic controller is a paired system and requires two controllers to be set up [14]. With advances in microprocessor computing capabilities, some researchers have developed natural optimization methods such as Genetic Algorithm (GA) [15], Multi-Objective GA [16][17], Tabu Search [18][19], Particle Swarm Optimization (PSO) [10] and CSA [20], artificial neural networks and others.

The Genetic Algorithm (GA) method is part of the Evolutionary Algorithm, which is an algorithm that imitates the natural evolutionary process where the main concept is that the most superior individuals will survive, while the weak individuals will become extinct [21]. The Particle swarm optimization (PSO) method is a population-based optimization technique inspired by the movements of fish and birds. The system is initialized with a population of random solutions, and the optimal solution is searched by updating the value [22].

One of the optimization methods used is an artificial neural network (ANN) [23]. The application of ANNs in the field of dynamic control systems has recently witnessed tremendous growth due to their excellent learnability, adaptability and generalization [24]. However ANN implementation requires a large number of parameters to be defined off line before implementation directly [25]. Due to the unavailability of faster training algorithms for ANNs the implementation of ANNs in on-line setup of controllers is still under-explored [26]. Cong and Liang proposed a controller-like local mixed neural network-based PID for motion control systems [25] using a conventional backpropagation algorithm [20].

One part of the control in the quadrotor which is very important in carrying out its mission is direction control. This control aims to determine the direction facing the quadrotor in carrying out its mission so that it is in accordance with the final destination. In this directional control several parameters must be considered because the quadrotor can be subject to uncertain disturbances originating from external [2]. The research contribution to this study the PID control system of a recurrent neural network is applied to fix the existing linear control problems so that the system can adapt to non-linear environmental changes. So from this description, this study will discuss quadrotor control using PID recurrent neural networks.

2. METHODS

2.1. Research Flow

This research was conducted in several stages. The stages of research carried out in this study are depicted in Figure 1. The first stage is carried out by making a quadrotor model. This quadrotor model has been created by previous studies. The next stage is to calculate the hardware components used. The value of weight, volume, and location of the components of the center of mass will affect the simulation calculation. The results of these calculations will then be used for system simulation parameters. In the next stage, researchers calculated the value of each PID component with the Ziegler Nicholas method. The next step is to fine tune the PID to get a better response. If you have received a good response, then proceed to make a PID design with RNN. The calculation results of the three methods were then tested to obtain the response value for each control stage.

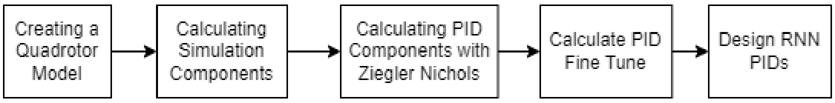

Creating a Quadrotor Model
Calculating Simulation Components
Calculating PID Components with Ziegler Nichols
Calculate PID Fine Tune
Design RNN PIDs

Figure 1. Research Diagram

2.2. Model Quadrotor

This study uses a quadrotor with an X-flying configuration which utilizes 4 motors which are divided into 2 front motors and 2 rear motors. The four motors in moving forward and backward are considered parallel to the earth's longitude axis, while movements to the right and left are considered to be parallel to the earth's latitude axis.

There are different directions of rotation of each rotor on the quadrotor. Two rotors rotate clockwise and the other two rotors rotate counterclockwise. Two rotors having the same direction of rotation are placed opposite each other. The rotation of each rotor on the quadrotor is shown in Figure 2 [27]. In the quadrotor X configuration, rotor 1 (M1) and rotor 3 (M3) rotate clockwise. M1 is placed on the front right, while M3 is placed on the rear left. The rotors M2 and M4 each rotate counterclockwise. M2 is placed on the front right and M4 is placed on the rear left of the Quadrotor. Each Mi rotor (with i = 1,2,3,4) will generate a force Fi which is proportional to the square of the motor speed. The lift force on the quadrotor is obtained from the sum of the upward thrust generated by each rotor. In this study the focus is on controlling the height on the Z axis.

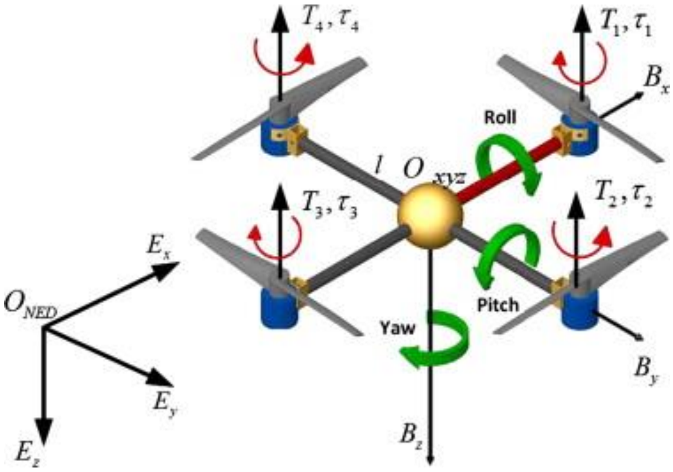


Figure 2. Quadcopter Model [27]

The system on the quadrotor has a lift (thrust) which is obtained through equation (1) [28].

$$Fi = b\omega_i^2, \quad i = 1,2,3,4 \tag{1}$$

Where F is the lift force on each rotor, i is the i-th rotor index while the value b is the constant value of the lifting force of the propeller used, and the value ω is the rotational speed of each propeller. The quadrotor translational motion is shown in equation (2) [29].

$$m\dot{v} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - {}^{o}R_B \begin{bmatrix} 0 \\ 0 \\ F_T \end{bmatrix} \tag{2}$$

The value of m is the mass of the quadrotor, $\dot{v}$ is the first derivative of velocity v, F_T is the total lift, ${}^{o}R_B$ are the earth frame coordinates, and g is the earth's gravitational velocity. The vertical motion of the quadcopter is affected by the total thrust 'T' and the weight 'w' along the z-axis shown in Figure 1. The rotating speed of the four rotating rotors is changed simultaneously to achieve vertical movement [30]. The large torque value that occurs in each propeller by the rotor will counteract the air friction force. The amount of torque value for the rotor surrounding the z-axis is obtained by the equation (3) [29].

$$\tau_i = k\omega_i^2 \tag{3}$$

Modelling on the quadrotor system is carried out using the Newton-Euler equation approach. Newton's second law is used to find the relationship between the thrust F and the acceleration a experienced by a center of mass. This is shown in equation (4). The lift force on the quadrotor is the result of the total lift force on each rotor on the quadrotor. The total force is obtained by equation (5) [31].

$$F = ma \tag{4}$$

$$F_T = u_1 = \sum_{i=1}^{4} F_i \tag{5}$$

The movement of the quadrotor on the earth's z-axis can be modelled using the rotation matrix in equation (4) and Newton's II law in equation (5), so that it is derived into three equations according to the force that occurs on each translation axis.

Translational motion on the z-axis can be described by equation (6) [32]. Lifting force (thrust) or F_{total} obtained from equation (5). Mark $R_{B_z}^{E_y}$ obtained from equation (7). In translational motion on the z-axis of the earth, the value of the force F is reduced by the weight of the quadrotor. At the time of translational movement on the y-axis of the earth, the angle that changes are only the angle ψ (yaw). Angle θ $(pitch)$ and ϕ $(roll)$ considered unchanged so that it is considered to have a value of 0. This makes equation (7) can be simplified into equation (8). Equation (9) shows the translational motion acting on the z axis [9].

$$F_z = -F_T . R_{B_z}^{E_z} + m.g \tag{6}$$

$$m\ddot{z} = -(\cos\phi \cos\theta)F_T + mg \tag{7}$$

$$m.\ddot{z} = -F_T + m.g \tag{8}$$

$$\ddot{z} = -\frac{1}{m}F_T + g \tag{9}$$

2.3. State Space Representation

A state space representation is a mathematical model of a physical system as a set of input, output, and state variables related to first-order differential equations [30]. A "state space" is a space that has a state variable as its axis. Vectors are used in state space to indicate the state of the system. In general, the state space representation of a linear system with input p, output q and variable n state is written in the form of equations (10) and (11) [31]. $\dot{\mathbf{x}}(t)$ is the State Vector, $\mathbf{y}(t)$ is the Output Vector, $\mathbf{u}(t)$ is the Input (or control) Vector, **A** is the System Matrix, **B** is the Input Matrix, **C** is the Output Matrix, **D** is the Feed forward Matrix.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{Ax}(t) + \boldsymbol{Bu}(t) \tag{10}$$

$$\boldsymbol{y}(t) = \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) \tag{11}$$

We start by selecting the system state. z is the total upward force on the u_2 quadrotor along the y-axis obtained from $(F_T - mg)$. So, from equations (5) and (9) we get equations (12) and (13).

$$\dot{z} = \dot{z} \tag{12}$$

$$\ddot{z} = -u_1/m \tag{13}$$

The following two states are practically suitable for quadcopters at 1 DOF, for altitude control, obtained from equations (10) to (13), obtained equations (14) and equations (15). z is the Position along the z-axis (height), v_z is the Speed along the z-axis The y output of this system consists of a vertical position $z(z)$.

$$\underbrace{\begin{bmatrix} \dot{z} \\ \dot{v}_z \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} z \\ v_z \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{\boldsymbol{B}} \underbrace{[u_1]}_{\boldsymbol{u}} \tag{14}$$

$$\underset{\boldsymbol{y}}{[z]} = \underset{\boldsymbol{C}}{[1\quad 0]}\underset{\boldsymbol{D}}{[0]}\underset{\boldsymbol{x}}{\begin{bmatrix} z \\ v_z \end{bmatrix}} \tag{15}$$

2.4. System Inertial Components

The stage of determining the parameters of this system is the calculation of each component characteristic used in the quadrotor system. The characteristics of these components describe the character of the component if simulated or modeled on a simulation system.

The mechanical frame used in this study uses the DJI F450 frame. The frame is 450mm diagonally from each rotor center. The quadcopter frame is made of plastic for the arms and circuit board material for the center frame. Each rotor is placed in an X configuration with a 45° tilt at the end of the quadcopter arm. ESC as a rotor speed regulator is placed under the quadcopter arm. The rotor unit used is a brushless rotor type and size X2212 with a power of 980 kV. Each rotor has a diameter of 285mm, propellers pitch 114mm. By using rotor specifications and propeller sizes of each 10×45 propeller-mounted brushless motor, a maximum lifting force of 0.7 kg can be achieved for each rotor. The results of the four rotors are used to lift a quadcopter weighing 1300 grams with a rotor speed of 50% of its maximum capacity. The mechanical picture of this study is shown in Figure 3.

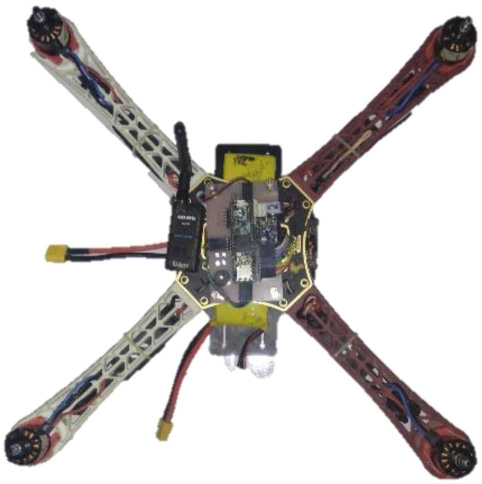


Figure 3. Quadrotor Mechanics Top View

The matrix **B** contains the value of the moment of inertia of the quadrotor for each axis on the quadrotor. The moment of inertia is denoted, $I_{xx}I_{yy}I_{zz}$, and is the moment of I_{xx} I_{yy} inertia acting on the axis and frame on the I_{zz} quadrotor x, y, z with kgm units respectively. The inertial value of the quadrotor is shown in equation (16) [32].

$$I_{zz} = \sum_{i=1}^{n}\left(I_{G_{zz_i}} + m_i(x_i^2 + y_i^2)\right) \tag{16}$$

One of the determinations of the specified parameters is in the form of system inertia parameters. The inertial components are determined on each of the x-axis, y-axis, and y-axis. The inertial components of this system are searched based on the dimensions and mass of each component. The component dimensions are shown in Table 1 and the component masses are shown in Table 2.

Table 1. Component Dimensions

No	Component	Long	Wide	Tall	Radius	Unit
1	Component centers	0.1395	00443	0.03125	-	meter
2	Quadrotor arm	0.215	0.03155	0.01025	-	meter
3	ESC	0.04525	0.025	0.0087	-	meter
4	Rotor	-	-	0.028	0.01375	meter
5	Baling- propeller	-	-	0	0.127	meter
6	Battery	0.14	0.05	0.03	-	meter

Table 2. Massa and Shape Type Components

No	Component	Mass	Unit	Kind
1	Component centers	0.68	kg	Beam
2	Quadrotor arm	0.027	kg	Beam
3	ESC	0.035	kg	Beam
4	Rotor	0.06	kg	Cylinder
5	Baling-propeller	0.008	kg	Cylinder
6	Battery	0.392	kg	Beam

Each component is then determined by the value of the moment of inertia based on the size of the component, the mass of the component, and the position of the component with respect to the center of mass. The center of mass on this quadrotor system is determined at the midpoint of the battery components. The resulting inertial value on each component is then summed on each axis. The result of calculating the total value of inertia on the axis x is: 1.24×10^2kgm^2, the total value of inertia on the axis y is 1.2×10^2kgm^2, While the total value of inertia on the axis z is $\mathbf{2.26 \times 10^2}$**kgm^2**.

2.5. PID Control Design with Recurrent Neural Network

The control system used in this study can be seen in Figure 4. The system gets input in the form of a reference of the specified value and then compares it with the reading values of several sensors. In this study, researchers focused on altitude sensors that can be in the form of barometers / sonars. The result of the difference in values is an error value. The error value of enter system will be entered in the control system PID RNN. The output of the control results will be converted into signal values which are then sent to the actuator in the form of a motor. The output of the motor is the movement of the quadrotor which is then read by the sensor again to enter the next system.

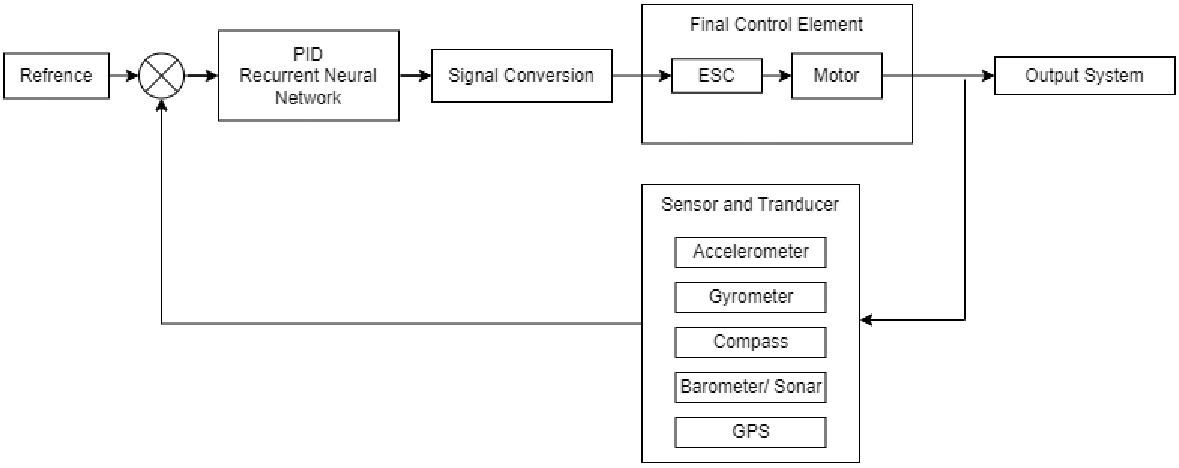


Figure 4. Diagram System Control

The structure of the PID-JST control diagram in Figure 5 is control using the principle of artificial neural networks. The neural network used in this system uses an online learning system. Artificial neural networks use three structural parts, namely:

- Input layer
- Hidden layer
- Output layer

The network used has 2 nodes on the input network, 3 nodes on the hidden network, and 1 node on the output network. This simple network requires real time output and the system can be implemented directly on the microcontroller. In the input network there are two inputs which are the error value and the previous error value. In the insert layer there are 3 nodes that will affect the value of the constants P, I, and D. The values of the constants P, I, and D resulting from the tuning of the simulation system are used as reference values in this inner layer. The result of this hidden node output will be the new values of the constants P, I, and D.

In this artificial neural network, surgery is performed to find new weight values in each network. This new weighting value is implemented with the principle of taking the smallest possible weight value with the smallest possible error value and error delta. The search for weight values on the network is carried out continuously based on input errors in the neural network system. This is done using derivatives in each calculation. This new weight calculation is performed at each iteration as part of the reverse propagation learning.

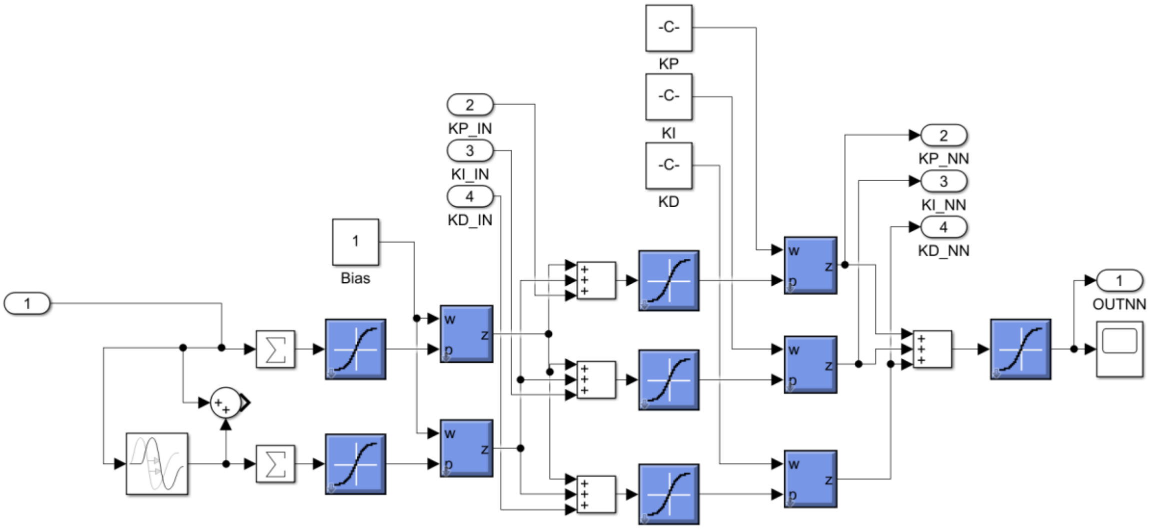


Figure 5. Neural Network Diagrams

The use of artificial neural networks in this study is shown in Figure 6. The use of repetitive neural networks is made using Simulink MATLAB diagrams. The result of the output of the PID constant value of the first neural network is recalculated by the second neural network. This is done repeatedly until 4 blocks are used. The output of the last block is entered as the value of the new PID constant from repeated recurrent neural network.

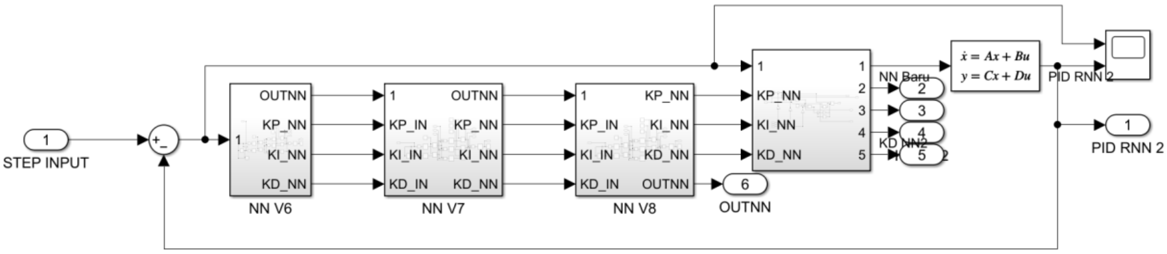


Figure 6. Block Diagram of the Use of Recurrent Neural Networks

3. RESULT AND DISCUSSION

3.1. Use of the Ziegler Nichols Method

System testing begins by finding the value of the PID constant using the Ziegler Nichols method. The use of this method begins by entering system parameters to determine the physical size of the quadrotor. This value will affect the result of determining the PID constant. In Ziegler Nichols tuning, the system must be able to oscillate rhythmically when obtaining the value of the initial KP constant. If the system can oscillate well, then the initial value can be a reference for the value of the PID constant. The results of the initial tuning to find oscillations of this system are shown in Figure 7. Figure 7 shows the system oscillating at amplitude 2 and repeating the oscillation time every 6 seconds.

As a result of the previous initial value, the system calculation gets the initial value of the PID constant, which is the P value of 0.0136, the I value is 0.0043, and the D value is 0.0106. The results of these constants are then tested using the step response in the simulation. The graph of the simulation results is shown in Figure 5. In Figure 8 it can be seen that the system can respond well to step test interference. Although when viewed with the response results, namely Rise Time of 1.2306 seconds, Overshoot 69.7894%, Steady state error of ±0.6979, Settling time with a time of 59.0160 seconds.

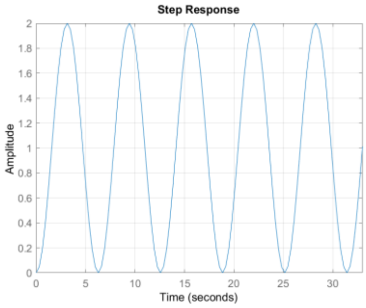


Figure 7. System Response to the ZN Method

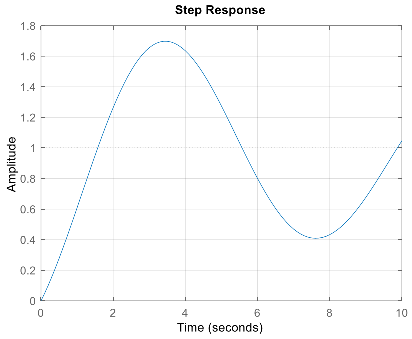


Figure 8. PID ZN Tuning Results

The results of the system have not been good results for a quadrotor system. This is shown in the results of the system response that is much more than one second. This shows that the control system is still not responsive. A very large overshoot result will make the quadrotor system oscillate up and down repeatedly and will be dangerous for quadrotor users. With a Steady state error value of about 1.6 meters from the reference value of 1 meter, it shows that the quadrotor is still up and down very far. This will make the quadrotor unable to perform its function properly.

Furthermore, researchers conducted a re-tuning based on the results of tuning using Ziegler Nichols. That initial reference value is used to get the value of the new PID constant. The search for the value of this new constant is based on the response of the system with an overshoot value of less than 1% so that the hope is that the system can work according to the purpose for which this system was created.

The system calculation gets the initial value of the PID constant, which is the P value of 0.2441, the I value is 0.0647, and the D value is 0.1846. The results of these constants are then tested using the step response in the simulation. The graph of the simulation results is shown in Figure 9. In Figure 9, it can be seen that the system can respond well to step test interference. Although when viewed with the response results, namely Rise Time of 0.2618 seconds, Overshoot 0.9296 %, Steady state error of ±0.0093, Settling time with a time of 0.4364 seconds.

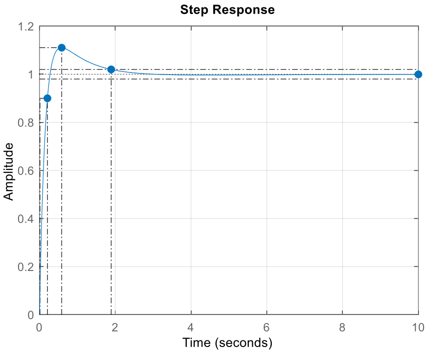


Figure 9. Respond PID Fine Tune

The results of the system's response show that the system is running better. The value of the parameters calculated in the response of the system makes the quadrotor control more stable. The value of the PID constant resulting from fine tuning will be the reference value of the PID constant on the artificial neural network in each block of the repeating neural network. The reference value will then be processed based on the error value resulting from changes in system reference on the neural network.

3.2. Recurrent Neural Network Control System

At the Recurrent Neural Network Control System stage, the value of the PID constant in the previous tuning is entered in the neural network system. This value will be the input of the refractive system in the neural network and the reference in the PID control system. A comparison of the system response results from the recurrent neural network system with the previous PID control is shown in Figure 10. In the results, it can be seen at a glance that the system response with PID RNN shown in orange dot has a faster response and lower overshoot when compared to the results of the PID response from the previous PID component tuning. In Figure 10 the blue line shows the input step for testing the response with a reference value of 1 meter.

The results of the response in Figure 9 produce a response Rise Time of 0.053 seconds, Overshoot 1.048%, Steady state error of 0.0093, Settling time with a time of ±0.075 seconds. The comparative results of each system's response are shown in Table 3. The result of the response shows that the system response value is much faster than the PID method alone. This will result in a good quadrotor response in carrying out its mission. The result of the fast response resulted in a larger system overshoot than the overshoot with a fine tune PID value. The greater overshot value is compensated for the system's extremely fast response. The results for the steady state error response and the settling time value produce a better response than the previous method. This small steady state error indicates that the system can maintain its position better. A small settling time value indicates a faster system to maintain its altitude faster. For the results of the response of the testing system with repeated interference can be seen in. In the Figure 11, it can be seen that the system can adjust changes in reference changes at any time.

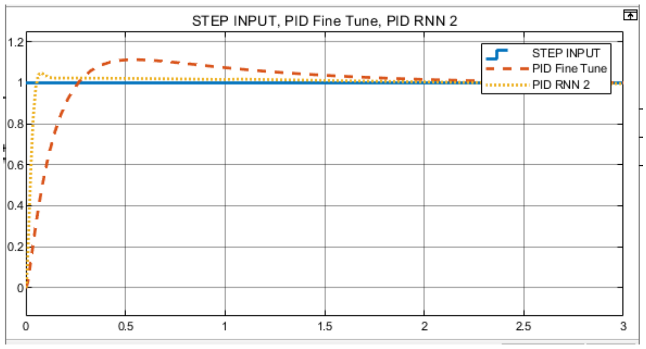


Figure 10. Comparison of Responses to Each Method

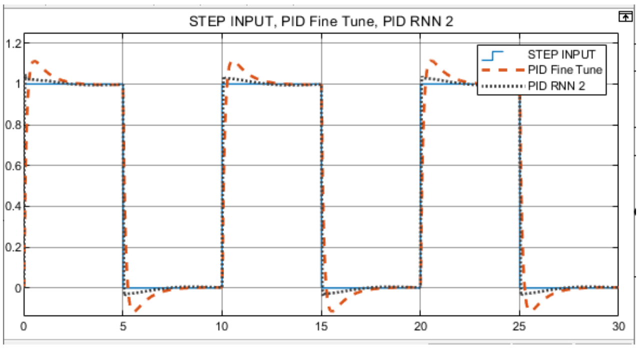
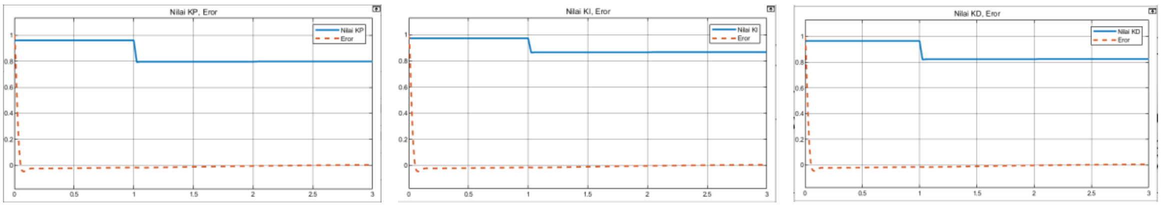


Figure 11. Response of Each Method with Changes in Reference Value

Table 3. Comparison of System Response of Each Method

Responds	PID ZN	PID Fine Tune	PID RNN
Rise Time	1.2306	0.2618	0.053
Overshoot	69.7894 %	0.9296 %	1.048%
Steady state error	±0.6979	±0.0093	±0.001
Settling time	59.0160	0.4364	0.075

The response result of the previous PID RNN system is the result of the PID RNN constant. The value of each component changes according to the error value in the system. The changes in each component are shown in for the Figure 12 value P, value I, and the constant D. The Figure 12 shows that the value of each component changes when there is a change in error. In the Figure 12, it can be seen that when the error value is high, the value of each PID component is high. However, when the error value has decreased, the value of each PID component drops to adjust.



(a) (b) (c)

Figure 12. (a) KP Value Changes with Errors, (b) Changes in KI Value with Errors, (c) Change in KD Value with Error

Changes in the value of the PID component can be seen in Table 4. In the Table 4, it can be seen that the PID RNN value changes in certain range values. The value of the RNN PID component is also relatively higher than the conventional PID method. This has an impact on the results of a faster system response. The value of the range of changes in the PID constant also shows that the system can adjust itself to the error values that occur. Changes in these values change based on the system error value that is included in the calculation of the RNN value in the system.

Table 4. Comparison of PID Parameter Values of each Method

	PID ZN	PD Fine Tune	PID RNN
KP	0.0136	0.2441	0.7969-0.9613
KI	0.0043	0.0647	0.8632-0.9734
KD	0.0106	0.1846	0.8183-0.9659

4. CONCLUSIONS

The results of this study obtained the results of the comparison of the method proposed by the researcher when compared with conventional control methods PID showed that the proposed method resulted in a better system response. Rise Time response results are 1.77 seconds faster than PID ZN and 0.2 seconds faster than PID fine tuning. The result of settling time is 58.941 seconds, faster than PID ZN and 0.3614 seconds faster than PID fine tuning. The overshoot result is 68.741% smaller than the ZN PID but 0.1184% larger than the fine tuning PID. This good response is needed by systems that require speed and precision. By using this method, the system can produce a faster response due to changes in the existing PID constant values. However, the response results also show a higher system overshoot. This is because the PID constant value is higher than conventional PID but it is not very fast enough to reduce the overshoot results. The result of the response is obtained from changes in each constant that can change according to changes in error values in the system.

ACKNOWLEDGEMENT

Thank you to LPPM Universitas Ahmad Dahlan for its role in funding this research.

REFERENCES

[1] L. Zhou and B. Zhang, “Quadrotor UAV Flight Control Using Backstepping Adaptive Controller,” *2020 IEEE 6th International Conference on Control Science and Systems Engineering, ICCSSE 2020*, pp. 163–166, 2020, https://doi.org/10.1109/ICCSSE50399.2020.9171967.

[2] Q. Jing, Z. Chang, H. Chu, Y. Shao, and X. Zhang, “Quadrotor attitude control based on fuzzy sliding mode control theory,” *Chinese Control Conference, CCC*, vol. 2019-July, pp. 8360–8364, 2019, https://doi.org/10.23919/ChiCC.2019.8865754.

[3] Q. Jiao, J. Liu, Y. Zhang, and W. Lian, “Analysis and design the controller for quadrotors based on PID control method,” *Proceedings - 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation, YAC 2018*, no. 15, pp. 88–92, 2018, https://doi.org/10.1109/YAC.2018.8406352.

[4] Y. Cheng, L. Jiang, T. Li, and L. Guo, “Robust tracking control for a quadrotor UAV via DOBC approach,” *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 559–563, 2018, https://doi.org/10.1109/CCDC.2018.8407194.

[5] C. Wang, Z. Chen, Q. Sun, and Z. Qing, “Design of PID and ADRC based quadrotor helicopter control system,” *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 5860–5865, 2016, https://doi.org/10.1109/CCDC.2016.7532046.

[6] M. K. Shaik and J. F. Whidborne, “Robust sliding mode control of a quadrotor,” *2016 UKACC International Conference on Control, UKACC Control 2016*, 2016, https://doi.org/10.1109/CONTROL.2016.7737529.

[7] T. K. Priyambodo, A. Dharmawan, and A. E. Putra, “PID self tuning control based on Mamdani fuzzy logic control for quadrotor stabilization,” in *AIP Conference Proceedings*, p. 020013, 2016, https://doi.org/10.1063/1.4940261.

[8] F. F. Rahani and T. K. Priyambodo, “Penalaan Mandiri Full State Feedback dengan LQR dan JST Pada Kendali Quadrotor,” *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 9, no. 1, p. 21, Apr. 2019, https://doi.org/10.22146/ijeis.37212.

[9] J. Fei and C. Lu, “Adaptive Sliding Mode Control of Dynamic Systems Using Double Loop Recurrent Neural Network Structure,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1275–1286, 2018, https://doi.org/10.1109/TNNLS.2017.2672998.

[10] H. Housny, E. Chater, and H. El Fadil, “Multi-closed-loop design for quadrotor path-tracking control,” *2019 8th International Conference on Systems and Control, ICSC 2019*, pp. 27–32, 2019, https://doi.org/10.1109/ICSC47195.2019.8950659.

[11] J. Li and W. Li, “On-Line PID Parameters Optimization Control for Wind Power Generation System Based on Genetic Algorithm,” *IEEE Access*, vol. 8, pp. 137094–137100, 2020, https://doi.org/10.1109/ACCESS.2020.3009240.

[12] C.-L. Lee and C.-C. Peng, “Analytic Time Domain Specifications PID Controller Design for a Class of 2nd Order Linear Systems: A Genetic Algorithm Method,” *IEEE Access*, vol. 9, pp. 99266–99275, 2021, https://doi.org/10.1109/ACCESS.2021.3093427.

[13] N. N. B. M. Mazlan, N. M. Thamrin and N. A. Razak, "Comparison Between Ziegler-Nichols and AMIGO Tuning Techniques in Automated Steering Control System for Autonomous Vehicle," *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 7-12, 2020, https://doi.org/10.1109/I2CACIS49202.2020.9140089.

[14] L. Shen and H. Xiao, “Delay-dependent robust stability analysis of power systems with PID controller,” *Chinese Journal of Electrical Engineering*, vol. 5, no. 2, pp. 79–86, Jun. 2019, https://doi.org/10.23919/CJEE.2019.000014.

[15] Y. Xiang, Z. Liu, and L. Wang, “Genetic-Algorithm-Optimization-Based Predictive Functional Control for Chemical Industry Processes Against Partial Actuator Faults,” *IEEE Access*, vol. 8, pp. 214586–214595, 2020, https://doi.org/10.1109/ACCESS.2020.3041015.

[16] L. Lin, A. Li, C. Xu, and Y. Wang, “Multi-Objective Genetic Algorithm Based Coordinated Second- and Third-Order Harmonic Voltage Injection in Modular Multilevel Converter,” *IEEE Access*, vol. 8, pp. 94318–94329, 2020, https://doi.org/10.1109/ACCESS.2020.2995293.

[17] H. Wei *et al.*, “Unified Multi-Objective Genetic Algorithm for Energy Efficient Job Shop Scheduling,” *IEEE Access*, vol. 9, pp. 54542–54557, 2021, https://doi.org/10.1109/ACCESS.2021.3070981.

[18] D. Zhang, Z. Gao and Z. Lin, "An Online Control Approach for Forging Machine Using Reinforcement Learning and Taboo Search," in *IEEE Access*, vol. 8, pp. 158666-158678, 2020, https://doi.org/10.1109/ACCESS.2020.3020550.

[19] A. A. Saadi, A. Soukane, Y. Meraihi, A. B. Gabis, and A. Ramdane-Cherif, “A Hybrid Improved Manta Ray Foraging Optimization With Tabu Search Algorithm for Solving the UAV Placement Problem in Smart Cities,” *IEEE Access*, vol. 11, pp. 24315–24342, 2023, https://doi.org/10.1109/ACCESS.2023.3255793.

[20] R. Sharma, V. Kumar, P. Gaur, and A. P. Mittal, “An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload,” *ISA Transactions*, vol. 62, pp. 258–267, 2016, https://doi.org/10.1016/j.isatra.2016.01.016.

[21] J. Kennedy, “Review of Engelbrecht’s fundamentals of computational swarm intelligence,” *Genetic Programming and Evolvable Machines*, vol. 8, no. 1, pp. 107–109, 2007, https://doi.org/10.1007/s10710-006-9020-8.

[22] H. Lu, J. Chen, and L. Guo, “Energy Quality Management,” in *Comprehensive Energy Systems*, Elsevier Inc., 2018, pp. 258–314, https://doi.org/10.1016/B978-0-12-809597-3.00521-6.

[23] N. Bacanin *et al.*, "Artificial Neural Networks Hidden Unit and Weight Connection Optimization by Quasi-Refection-Based Learning Artificial Bee Colony Algorithm," in *IEEE Access*, vol. 9, pp. 169135-169155, 2021, https://doi.org/10.1109/ACCESS.2021.3135201.

[24] J. Sun, S. Sathasivam, and M. K. B. M. Ali, “Analysis and Optimization of Network Properties for Bionic Topology Hopfield Neural Network Using Gaussian-Distributed Small-World Rewiring Method,” *IEEE Access*, vol. 10, pp. 95369–95389, 2022, https://doi.org/10.1109/ACCESS.2022.3204821.

[25] S. Cong and Y. Liang, “PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems,” *IEEE Transactions on Industrial Electronics*, 2009, https://doi.org/10.1109/TIE.2009.2018433.

[26] S. J. Ho, L. S. Shu, and S. Y. Ho, “Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA,” *IEEE Transactions on Fuzzy Systems*, 2006, https://doi.org/10.1109/TFUZZ.2006.876985.

[27] R. Dasgupta, S. B. Roy, O. S. Patil and S. Bhasin, "A Singularity-free Hierarchical Nonlinear Quad-Rotorcraft Control using Saturation and Barrier Lyapunov Function," *2019 American Control Conference (ACC)*, Philadelphia, pp. 3075-3080, 2019, https://doi.org/10.23919/ACC.2019.8814632.

[28] Y. Lei and H. Wang, "Aerodynamic Performance of a Quadrotor MAV Considering the Horizontal Wind," in *IEEE Access*, vol. 8, pp. 109421-109428, 2020, https://doi.org/10.1109/ACCESS.2020.3002706.

[29] A. Ollero, “Aerial Robotic Manipulators,” in *Encyclopedia of Robotics*, pp. 1–8, 2019. https://doi.org/10.1007/978-3-642-41610-1_78-1.

[30] Z. Tahir, “State Space System Modeling of a Quad Copter UAV,” *Indian Journal of Science and Technology*, vol. 8, no. 1, pp. 1–5, Jan. 2015, https://doi.org/10.17485/ijst/2016/v9i27/95239.

[31] Y. Jiang, S. Yin, J. Dong and O. Kaynak, "A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes," in *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12868-12881, 2021, https://doi.org/10.1109/JSEN.2020.3033153.

[32] A. B. Zakaria and A. Dharmawan, “Sistem Kendali Penghindar Rintangan Pada Quadrotor Menggunakan Konsep Linear Quadratic,” *Indonesian Journal of Electronics and Instrumentation Systems*, vol. 7, no. 2, pp. 219–230, 2017, https://doi.org/10.22146/ijeis.25503.

AUTHOR BIOGRAPHY

Faisal Fajri Rahani the author completed his undergraduate education in the Electronics and Instrumentation Study Program at Universitas Gadjah Mada in 2016 and Master Education at the Master of Computer Science Study Program at Universitas Gadjah Mada in 2018. Currently, author 2 is a permanent lecturer at the Informatics Study Program, Ahmad Dahlan University. His research areas are control systems, robotics, and artificial intelligence.

Phisca Aditya Rosyady the author completed his undergraduate education in the Electronics and Instrumentation Study Program at Universitas Gadjah Mada in 2014 and Master Education at the Phisca Aditya Rosyady in 2017. Currently, author 2 is a permanent lecturer at the Electronic Engineering Study Program, Ahmad Dahlan University. His research areas are Wireless Sensor Network & Image Processing.