.

Quadrotor height control system using LQR and recurrent artificial neural networks

Faisal Fajri Rahani[1*], Miftahurrahma Rosyida[2]

[1,2]Department of Informatics Engineering Education, Universitas Negeri Yogyakarta, Indonesia

ABSTRACT

The quadorotor is a type of unmanned flying vehicle known as Unmanned Aerial Vehicle (UAV). In recent years, quadrotors have attracted much attention from researchers around the world due to their excellent maneuverability. A good control system in this quadrotor system is needed for ease of use of this quadrotor. One control system that is often used is the Linear Quadratic Regulator (LQR) control system. This control system has challenges for dynamic system disturbances in quadrotor control. Researchers proposed a recurrent artificial neural network (RNN) system to address these challenges.RRN is used to change the value of the feedback component in the LQR control system. The nature of the feedback component in LQR, which is static, is changed based on the system error value based on changes in the error value entered into the RNN. The result of this RNN is a change in the value of the LQR feedback component based on the input of the system. The results of this research show that LQR control with RNN produces a faster system response of 0.075 seconds and a faster settling time of 0.221 seconds. Compensation for the system response speed produces a higher overshot value.

Corresponding Author:

Faisal Fajri Rahani,
Department of Informatics,
Universitas Ahmad Dahlan,
Jl. Ring RoadSelatan, Kragilan, Tamanan, Banguntapan, Bantul, Daerah Istimewa Yogyakarta, Indonesia
Email: faisal.fajri@tif.uad.ac.id
https://doi.org/10.52465/joscex.v5i2.379

1. INTRODUCTION

Quadorotor is a type of unmanned flying vehicle or known as an Unmanned Aerial Vehicle (UAV) [1]. In recent years, quadrotors have attracted much attention from many researchers around the world, due to the fact that their maneuverability is excellent [2]. The quadrotor is capable of carrying out autonomous flights in unmanned mode. This is because quadrotors can be operated with remote controllers and independent or autonomous program control devices [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, civilian aerial photography, etc. [5]. Even now quadrotors can carry out several types of dangerous and difficult missions such as search and rescue, surveillance, inspection, mapping, and aerial cinematography [6].

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

    In recent years, there has been a tendency to emerge novelty in control theory. Researchers have obtained several theoretical algorithms to control quadrotors. Conventional control for quadrotors that have been widely used, such as PID [7], state feedback control system [8], LQR [9] and so on. However, due to the linear nature of the control, many non-linear controls have been developed that have begun to be developed. Some of these controls are, for example, using an adaptive sliding mode control system [10], fuzzy method [11] and control by combining a liner control system with an artificial intelligence system [8]. With advances in the computing capabilities of microprocessors, some researchers are developing natural optimization methods such as Genetic Algorithm (GA) [12], Multi-Objective GA, Tabu Search, Particle Swarm Optimization (PSO) [11] and CSA [13], artificial neural networks [14] and others.

    The full-state feedback LQR method is a development of the pole placement method and the Hamiltonian Jacobi Bellman (HJB) method by setting the state of the system on a zero-value system by inserting the system as optimally as possible [15]. The setting has the goal of minimizing the overshoot values and obtaining a timely response according to the specified system criteria.

    The altitude control system with the full-state feedback LQR method on quadrotors has weaknesses caused by several basic things. This is due to the linearization of the control with LQR and the improper calculation of the control modeling parameters [16]. The value of the fixed feedback constant component will make the system response poor. This is because the quadrotor system receives many irregular disturbances, while the control system used cannot accommodate these disturbances [17]. Due to this, a system is needed to regulate the value of the feedback constant so that the system can maintain the height of the quadrotor system.

    From several method studies above, one of the good methods for time series data is the method Recurrent Neural Networks (RNN). The application of RNNs in the field of dynamic control systems has recently witnessed tremendous growth due to its remarkable learning, adaptability, and generalization. However, RNN implementation requires a large number of parameters to be defined offline before live implementation. Due to the unavailability of faster training algorithms for RNNs, the application of RNNs in on-line tuning of controllers is still poorly explored.

2. METHOD
Quadrotor Model Determination

    At this stage, the quadrotor system model was determined using Newton-Euler's concept obtained with previous research. This stage will simulate a quadrotor system model with a mathematical model approach for translational motion and quadrotor rotational motion.

    This research uses a quadrotor with an X configuration that utilizes four rotors. These four rotors are divided into two front rotors and two back rotors. This pair of two pairs of rotors rotate in different directions. Two rotors rotate in a clockwise direction, whereas the remaining two rotate in a counterclockwise direction. Each arm of the quadrotor is equipped with rotors with the same rotation direction [18]. This can be seen in Figure 1.
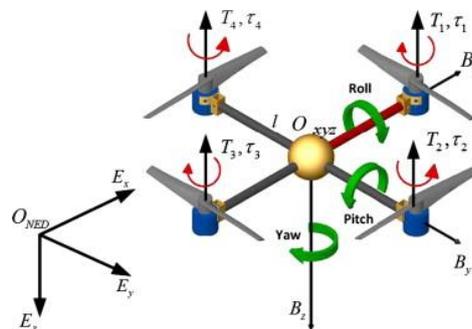

Figure 1. Quadcopter model [19]

.

Each Mi rotor with i = 1, 2, 3, 4 will generate a force Fi that is proportional to the square of the motor speed. The lift force on the quadrotor is calculated as the sum of the upward thrust generated by each rotor. This study focuses on controlling the height of the Z-axis. The quadrotor has a lift, which is obtained through Equation (1) [18].

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

Where F is the lifting 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 is the rotational speed of each propeller.

Modeling of the quadrotor system is carried out using the Newton- Euler equation approach. Newton's alternate law is used to find the relationship between thrust F and the acceleration a endured by a center of mass. This is shown in Equation (2). The lift force on the quadrotor is the result of the total lift force on each rotor on the quadrotor. The total force is reached by equation (5) [18].

$$F = ma \tag{2}$$

$$F_T = u_1 = \sum_{i=1}^{4} F_i \tag{3}$$

The movement of the quadrotor on the earth's z-axis can be modeled using the rotation matrix in equation (2) and Newton's II law in equation (3), so that it is derived into three equations according to the force that occurs on each translation axis.

The translational motion on the z-axis can be described by Equation (4) [13]. Lifting force (thrust) or F_{total} obtained from equation (5). 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 is only the angle $\psi(yaw)$. Angle $\theta(pitch)$ and $\phi(roll)$ considered unchanged so that it is considered to have a value of 0. This makes Equation (5) can be simplified into equation (6). Equation (7) shows the translational motion acting on the z-axis [20].

$$F_z = -F_T.R_{B_z}^{E_z} + m.g \tag{4}$$

$$m\ddot{z} = -(cos\phi cos\theta)F_T + mg \tag{5}$$

$$m.\ddot{z} = -F_T + m.g \tag{6}$$

$$\ddot{z} = \frac{-1}{m}F_T + g \tag{7}$$

Determination of Control System Parameters

This stage will be carried out to take parameters on the quadrotor system in accordance with the actual situation. This is done to go beyond the mathematical models that have been made. Parameter values are obtained from various sources such as measurements, data sheets, etc. The components that are taken parameters are motors, quadrotor frames, electronic systems, and so on.

The stage for determining system parameters is the calculation of system characteristics which are influenced by each component used in the quadrotor system. The character of these components describes the system if it is modeled in a simulation system.

The mechanical frame used in this research is 450 mm diagonally from each rotor center. The quadrotor frame is made of plastic for the arms and circuit board material for the middle frame. Each rotor is placed in an X configuration with a 45 ° tilt at the end of the quadcopter arm. The ESC as a rotor speed controller 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 285 mm, propeller pitch 114mm. Using the rotor specifications and propeller size of each brushless motor mounted on a 10×45 propeller, 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. This character of the system uses the system character used in previous research [18].

The moment of inertia of this system is described by, $I_{xx}I_{yy}I_{zz}$, 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 (8) [18].

$$I_{zz} = \sum_{i=1}^{n} \left(I_{G_{zz_i}} + m_i(x_i^2 + y_i^2) \right) \tag{8}$$

The component dimensions are shownTable 1 and the component masses are shown in Table 2.

Table 1. Component dimensions [18]

No	Component	Long	Wide	Tall	Radius	Satuan
1	Component centers	0,1395	0,0443	0,03125	-	metre
2	*Quadrotor* arm	0,215	0,03155	0,01025	-	metre
3	ESC	0,04525	0,025	0,0087	-	metre
4	Rotor	-	-	0,028	0,01375	metre
5	Baling- propeller	-	-	0	0,127	metre
6	Battery	0,14	0,05	0,03	-	metre

Table 2. Components of mass and shape type [18]

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 element is also determined by the value of the moment of indolence grounded on the size of the element, the mass of the element, and the position of the element with respect to the center of mass. The inertial value on each element is also added on each axis. The result of calculating the total value of indolence on the axis: $2{,}26 \times 10^2 \textbf{kgm}^2$. This number was also used in previous research [18].

Design of the quadrotor control

At this stage, the design of the classic LQR control and the LQR control of artificial neural networks is carried out. The design of this control system pays attention to several parameters that will affect the existing control system. The parameters that are the reference for the artificial neural network system are the system response read from the input angle and the angle velocity input processed from the system simulation test.
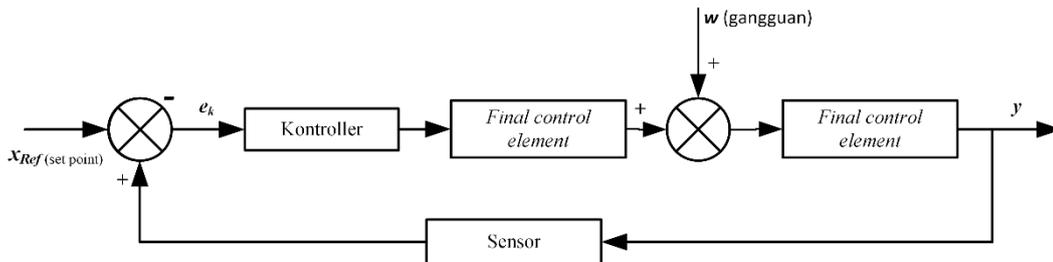

Figure 2. Closed loop control system for the quadrotor [8]

Control of the quadrotor is generally shown in Figure 2. The value of entering the system is in the form of $Xref$ set points. The input can be in the form of values that have been determined by the program or from the control system. The reference value is then compared to the sensor reading that shows the position of the quadrotor relative to the ground. The result of the difference is called the error value. The error value will be a reference of the controller so that the controller can control according to the error value obtained. The controller issues a control signal which then becomes the input to the final block of control elements. The block serves to change the position of the quadrotor so that its position matches the desired reference value. The result of the control is then read back by the sensor and then re-compared with the reference value.

State Space Representation

.

A state space representation is a mathematical model of a system in the form of input, output, and state variables related to first-order differential equations [21]. 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(9)10) [20].

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{9}$$

$$y(t) = Cx(t) + Du(t) \tag{10}$$

$\dot{x}(t)$ is *State Vector*, $y(t)$ is *Output Vector*, $u(t)$ is *Input Vector*, **A** is *System Matrix*, **B** is *the input matrix,* **C** is *the output matrix,* and **D** is the feed *forward matrix.* We start by selecting *the state of the system.* The total upward force on the u_2 *quadrotor* along the y-axis obtained from $(F_T - mg.$ (3)11(7))(11)(12).

$$\dot{z} = \dot{z} \tag{11}$$

$$\ddot{z} = -u_1/m \tag{12}$$

The following two *states* are practically suitable for *quadcopters* at 1 DOF, for altitude control, obtained from equations (9) to (12(9)14(12)(14) *where z* is a position along the z-axis (height) and v_z is a speed along the z-axis. The y output of this system consists of a vertical position.

$$\begin{bmatrix} \dot{z} \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ v_z \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} [u_1] \atop \dot{x} \, A \, x \, B \, u \tag{13}$$

$$[z] = [1 \quad 0][0] \begin{bmatrix} z \\ v_z \end{bmatrix} \atop y \, C \, D \, x \tag{14}$$

Design Control with LQR with Recurrent Artificial Neural Network

At this stage, the implementation of LQR classical control is carried out with the value of the tuned feedback paremeter and the results of tuning the artificial neural network. This control simulation stage is carried out by encoding LQR controls using MATLAB. In this simulation, it is used to observe the response of the system. At this stage, the transient response of the system is observed. The system's transient response is in the form of overshoots, steady-state errors, and so on.

The artificial neural network architect used in this study can be seen in Figure 3. The system in this architecture gets an input from the error value of the position of the system altitude. This error is obtained from the difference between the reference value and the sensor reading value. The results then became input from the first artificial neural network. The output of this first artificial neural network is then input from the second artificial neural network. The output of this second artificial neural network becomes the input of the third artificial neural network. The results of this third artificial neural network are taken from the results of the hidden layer output of the third artificial neural network. This value will be the value of the feedback constant (K) of the system used.
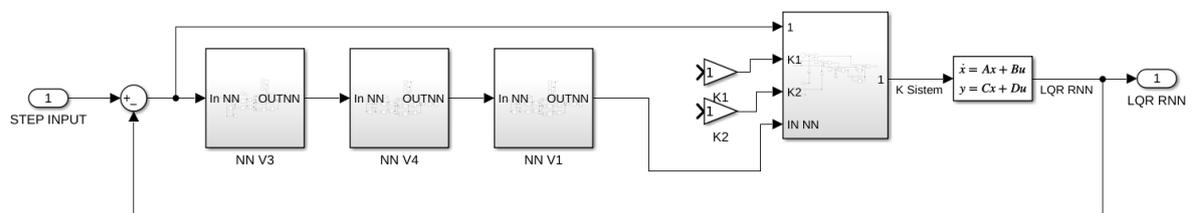

Figure 3, Recurrent neural network diagrams

In this part is the architecture of the artificial neural network used in the artificial neural network series in Figure 3. In the repetitive artificial nerve network, there is an artificial nerve network in it. This artificial neural network consists of 1 input node, 3 hidden nodes, and 1 output node. Input from the system network is the system error value on the first neural network which is then processed into the current error value and the previous error value. Then the recorded value is inserted into the artificial neural network. Each of these nodes uses the sigmoid activation function. The artificial neural network used in this system is shown in Figure 4.
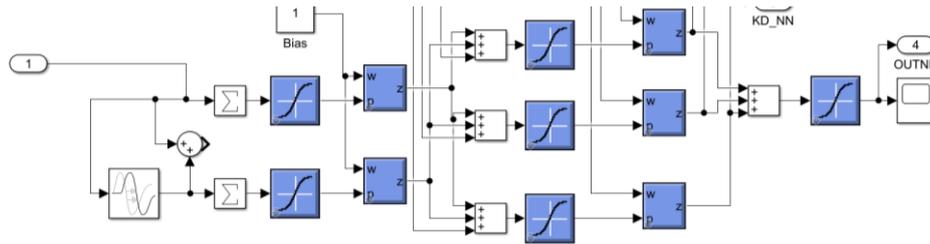
Figure 4. Artificial neural network architecture in RNNs

3.   RESULTS AND DISCUSSIONS

The results obtained from this initial stage are the results of the calculation of inertia for each axis of this system. The result of the calculation of the total value of inertia on the z-axis is $2,26 \times 10^2$ **kgm²**. These results are then used for the simulation parameters of the system.

Calculating the Controllability System

Controlability is a test to determine how many states the system can be controlled with. The results of this stage show that on the Z axis 2 control states can be determined.

```
m=1.2;
p = 1/m;
zz = 0.0226;

A = [0 1;
     0 0;];
B = [0 0;
     p 0;];
C = [1 0;];
D = [0 0];

states = {'z' 'vz'};
input = {'u4'};
outputs = {'z';};

sys_ss =
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
poles = eig(A);
co = ctrb(sys_ss);
controllability = rank(co)
```

Output:

```
ANS:

controllability
=

      2
```

Calculating the value of the Feedback K Component

In this section, the researcher focuses on the Z-axis control system. The program listing used for Z-axis control. The parameters used are shown in Figure 5. The parameters that play a role in the control of the Z axis are the mass of the *quadrotor* and the influence of gravity.

.

```
1    m=1.2;
2    g=9.8;
3    u=m*g;
4    % u=1;
5    p = u/m;
```

Figure 5. System parameters

Next, the state system for the Z axis is with values A, B, C, and D. This state will only affect the Z-axis. The state system is shown in Figure 6.

```
1    A = [0 1;
2          0 0;];
3    B = [0;
4          p;];
5    C = [1 0;];
6    D = [0];
```

Figure 6. State system Sb Z

From the state of the system above, the system is then tested with a step test on Matlab. The results of this test are shown in **Error! Reference source not found.**. The figure shows that the system response has r eached the desired height set point value. However, in this figure the value of the system output returns to a high error value towards zero again.
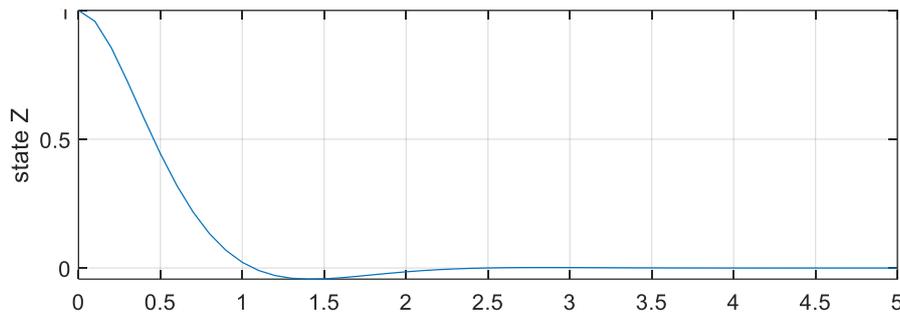

Figure 7. Results of the Step test for Sb Z

Next, the feedback component tuning was carried out for Sb Z. The value of the K component obtained was K(1) = 31.6228 and K(2) = 2.5404. The results of the system response for these components are shown in **Error! Reference source not found.**. The results show that the system response can be stable in less than 0.6 s econds. However, in **Error! Reference source not found.** it can be seen that the final value of the system response has not reached 1 which is the final goal of the system. The results of the test obtained a rise *time value* of 0.112, *an overshoot value* of 4.32%, *a settlement time* of 0.339, and a *final value* of 0.0316. This shows a good response for a system, but with the *final value* that is not yet suitable, this system cannot be considered good.
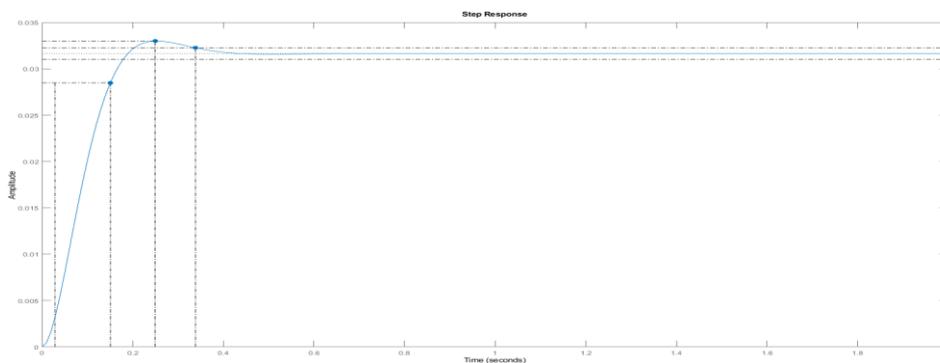

Figure 8. Results of Sb. Z's response

To improve the system response so that it can reach the appropriate final response, an additional Nbar system is given. This system uses the formula -inv(C*inv(A-B*K)*B). The result of this calculation is entered for the value component B. The response result of this new component is shown in. In the figure, it is shown that the final value can go to the final value of 1 for the system response to the step. The results of the response in **Error! Reference source not found.**show that the final value target is in accordance with the target (1). In o ther words, the value of steady-state error in the system is close to zero. The analysis of the system response in this method is shown in **Error! Reference source not found.**. In **Error! Reference source not found.**, the *rise time value is* 0.112, *the overshoot* is 4.32%, the *settling time* is 0.339, and *the final value* is 1. The components of *rise time*, *overshoot,* and *Settling time* produce the same value, but the biggest difference is the final value, in the *final value* which gets a value of 1. This shows that the system can work according to the desired set-point value.
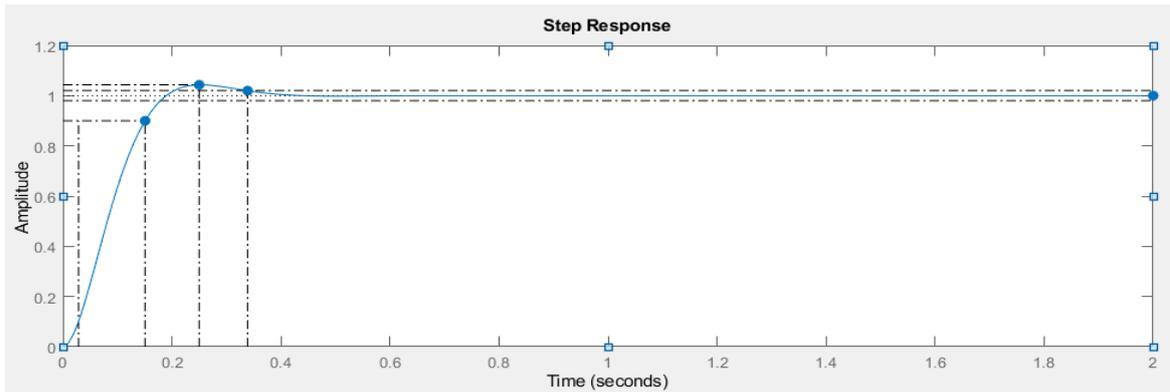

Figure 11. Results of the Z-axis response with the addition of Nbar
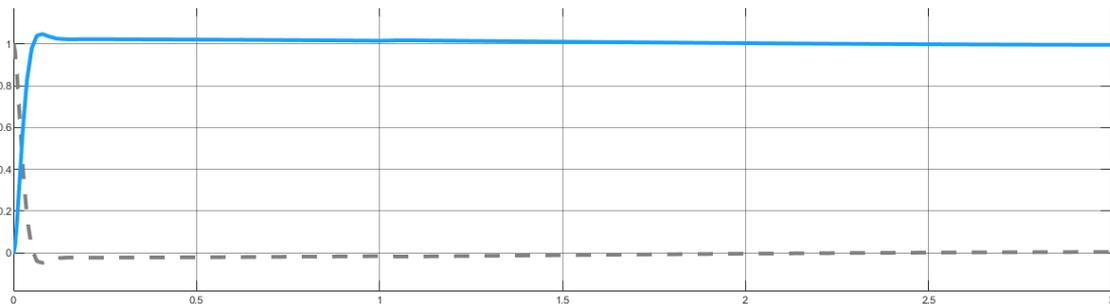

Figure 12. Results of the Z-axis response with RNN

In **Error! Reference source not found.**, the *rise time value is* 0.0371, *the overshoot* is 7.91%, *the s ettling time* is 0.118, and the *final value* is 1. The rise *time component* and *Settling time* produce better values, but greater overshoot values. This indicates that the system can run faster but results in a greater overshoot response.

Table 3 Comparison of the system response of each method

Responds	LQR	LQR RNN
Rise Time	0.112	0.037
Overshoot	4.32%,	7.91%
Settling time	0.339	0.118

In results shown in Table 3 it is shown that there is an improvement in the Rise time and Settling time response. However, the table also shows that improving this response causes the overshoot value to become higher. This is compensation for the faster response value.

.

4.   CONCLUSION

The results of this research were obtained from the comparison of the method proposed by the researcher compared with the conventional LQR control method, showing that the proposed method, namely LQR with RNN, produces a better system response. The result of the LQR rise time response is 0.112 seconds, which is a much larger value than the LQR RNN response time value of 0.0371. With a better response value, this will allow the to quickly adjust the system reference value of the system. The rapid improvement of the system response with the LQR RNN method must be compensated for by a greater system overshoot value. The results of overshoot in the LQR RNN control showed a value of 7.91%, which is a higher value than the overshoot value in the LQR control. However, the compensation of these values also results in better settlement time values in the LQR RNN. The results are shown in Table 3. The results of this research show that the LQR system with RNN produces a better response and settles time values. Increasing the overshoot value is the result of a faster response. From the results of this research, it can be concluded that the results of this method produce better system output for systems that require a faster response.

REFERENCES

[1]   L. Zhou and B. Zhang, "Quadrotor UAV Flight Control Using Backstepping Adaptive Controller," in *2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE)*, Jul. 2020, pp. 163–166. doi: 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," in *2019 Chinese Control Conference (CCC)*, Jul. 2019, pp. 8360–8364. doi: 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," *Proc. - 2018 33rd Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2018*, no. 15, pp. 88–92, 2018, doi: 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," *Proc. 30th Chinese Control Decis. Conf. CCDC 2018*, pp. 559–563, 2018, doi: 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," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 5860–5865, 2016, doi: 10.1109/CCDC.2016.7532046.

[6]   M. K. Shaik and J. F. Whidborne, "Robust sliding mode control of a quadrotor," *2016 UKACC Int. Conf. Control. UKACC Control 2016*, 2016, doi: 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," *AIP Conf. Proc.*, vol. 1705, no. 1, p. 020013, Feb. 2016, doi: 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 J. Electron. Instrum. Syst.*, vol. 9, no. 1, pp. 21–32, Apr. 2019, doi: 10.22146/ijeis.37212.

[9]   F. F. Rahani and T. K. Priyambodo, "Implementasi Full State Feedback LQR dengan JST pada Kendali Ketinggian Quadrotor," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 4, p. 357, Nov. 2019, doi: 10.22146/jnteti.v8i4.536.

[10]   J. Fei and C. Lu, "Adaptive Sliding Mode Control of Dynamic Systems Using Double Loop Recurrent Neural Network Structure," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 4, pp. 1275–1286, 2018, doi: 10.1109/TNNLS.2017.2672998.

[11]   H. Housny, E. Chater, and H. El Fadil, "Multi-closed-loop design for quadrotor path-tracking control," *2019 8th Int. Conf. Syst. Control. ICSC 2019*, pp. 27–32, 2019, doi: 10.1109/ICSC47195.2019.8950659.

[12]   Z. Yan and Y. Zhou, "Application to Optimal Control of Brushless DC Motor with ADRC Based on Genetic Algorithm," *Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020*, pp. 1032–1035, 2020, doi: 10.1109/AEECA49918.2020.9213554.

[13]   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 Trans.*, vol. 62, pp. 258–267, 2016, doi: 10.1016/j.isatra.2016.01.016.

[14]   F. M. S. Al-Zwainy, F. M. Al-Zwainy, H. Ala, and H. F. Ibraheem, "Predicting productivity in construction industry utilizing multiple linear regression technique and artificial neural network technique: A review for research and applications," *Int. J. Res. Adv. Eng. Technol. 7 Int. J. Res. Adv. Eng. Technol.*, 2020.

[15]   W. K. Alqaisi, B. Brahmi, J. Ghommam, M. Saad, and V. Nerguizian, "Adaptive Sliding mode Control Based on RBF Neural Network Approximation for Quadrotor," *IEEE Int. Symp. Robot. Sensors Environ. ROSE 2019 - Proc.*, 2019, doi: 10.1109/ROSE.2019.8790423.

[16]   C. Sun, T. Lu, and K. Yuan, "Balance control of two-wheeled self-balancing robot based on Linear Quadratic Regulator and Neural Network," *2013 Fourth Int. Conf. Intell. Control Inf. Process.*, vol. 1, pp. 862–867, 2013, doi: 10.1109/ICICIP.2013.6568310.

[17]   P. Gautam, "Optimal control of Inverted Pendulum system using ADALINE artificial neural network with LQR," *2016 Int. Conf. Recent Adv. Innov. Eng.*, pp. 1–6, 2016, doi: 10.1109/ICRAIE.2016.7939523.

[18]   F. F. Rahani and P. A. Rosyady, "Quadrotor Altitude Control using Recurrent Neural Network PID," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 5, no. 2, pp. 279–290, Apr. 2023, doi: 10.12928/biste.v5i2.8455.

[19]   L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, *Quad Rotorcraft Control*, vol. 1. London: Springer London, 2013.

[20]   O. A. Dhewa and F. F. Rahani, "Peningkatan Kestabilan Quadrotor menggunakan Kendali Linear Quadratic Regulator dengan Kompensasi Integrator dalam Mempertahankan Posisi," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 4, no. 2, pp. 62–75, Nov. 2022, doi: 10.12928/biste.v4i2.6808.

[21]   Z. Tahir, "State Space System Modeling of a Quad Copter UAV," *Indian J. Sci. Technol.*, vol. 8, no. 1, pp. 1–5, Jan. 2015, doi: 10.17485/ijst/2016/v9i27/95239.