Design Quadcopter Automatic Control System for Obstacle Avoidance Using Linear Quadratic Regulator (LQR) with LiDAR Sensor

Purwadi Agus Darwito[1(✉)], Muhammad Tabayyun Yudhistira[1], Hermawan Nugroho[2], and Totok Ruki Biyanto[1]

[1] Department of Engineering Physics, Sepuluh Nopember Institute of Technology Surabaya, Surabaya, Indonesia
{padarwito,trb}@ep.its.ac.id

[2] Department of Electrical and Electronic Engineering, University of Nottingham Malaysia Selangor, Semenyih, Malaysia
Hermawan.Nugroho@nottingham.edu.My

Abstract. Unmanned Aerial Vehicle (UAV) Quadcopter has many functions and uses as logistics transportation with the advantages of transportability efficiency as a means of air transportation. Barriers to UAVs in carrying out their functions as air transportation are obstacles. UAV requires an automatic control system that can fly following the trajectory, detect and avoid obstacles. The obstacle avoidance control system in this study uses the Linear Quadratic Regulator (LQR) method with a LiDAR sensor. The point cloud method in LiDAR acts as a determinant of obstacle avoidance decisions. The 6 DOF UAV control system using LQR control method with LiDAR sensor has been well designed. The quadcopter UAV can fly following the trajectory and detect and avoid obstacles autonomously by means of simulations in Matlab software. The results of the simulation state that the quadcopter control system can fly following the trajectory, also detect and avoid obstacles with the LiDAR sensor as an obstacle detecting sensor, where the avoidance path is formed due to the new coordinates obtained from the avoidance algorithm.

Keywords: quadcopter · linear quadratic regulator · obstacle avoidance

1 Introduction

Unmanned Aerial Vehicle (UAV) and better known as a flying machine system vehicle that can fly in the air with remote control controlled by an operator or pilot or is able to control itself which has been programmed by a programmer [1]. Unmanned Aerial Vehicle (UAV) can be divided into two types based on the shape of the wings, namely fixed wings and multi rotary wings. Fixed wings are a type of UAV that has wings, with both sides of the aircraft having wings or ailerons that function as aircraft guides at a certain height [2]. As well as fixed wings have rudder and elevator parts that function for aircraft to maneuver. And the UAV type multi rotary wings is a UAV that does not

S. Sugiman et al. (Eds.): MIMSE-M-E 2022, AER 216, pp. 193–206, 2023.
https://doi.org/10.2991/978-94-6463-078-7_21

have wings but has many rotors and more than one propeller. The UAV type multi rotary wings is intended for access to locations or terrain that cannot be reached by winged aircraft. Multi-rotary wings UAVs have different configurations on various motor parts, Tricopter which has three motor and arm configurations, Quadcopter which has four motor and arm configurations, Hexacopter which has six motor and arm configurations and Octocopter which has eight motor and arm configurations [1].

As previously mentioned, apart from being a vehicle for aviation in the military field, UAVs are now starting to be designed as an ecosystem to help fulfill human needs. In the industrial sector, the UAV is further developed as a flight vehicle for logistics transportation. UAV has advantages in transportability efficiency as a means of transportation that can pass through the air, can make its own path and can perform time efficiency. In the development of UAV technology as a means of logistics transportation, a UAV system that has the capability and autonomous system is designed. Autonomous UAV can be interpreted as a flight vehicle that can operate without human intervention as the controller of the UAV, because autonomous UAV is designed by an application system to be able to operate without human intervention as a UAV controller [3].

Previous research stated that quadcopters that use LiDAR sensors have a better ability to analyze obstacles that will be faced [12] [13] [14]. So in this final project, an automatic control system for the UAV quadcopter was developed with the application of LiDAR to avoid obstacles. It is hoped that this research can be applied to quadcopters as a function of logistics transportation and other functions based on community needs.

2 Methodology

The process carried out in this study begins with problem identification and literature study on the quadcopter which is the object of research. After the data is obtained, mathematical modeling of the quadcopter will then be carried out which will then be used for input into the LQR control system. Thapan then carried out the design of the LiDAR point cloud. The next stage is the integration of LQR and obstacle avoidance. The last stage is the analysis of the simulation results and discussion.

2.1 Quadcopter Assembly and Data Collection

In accordance with the previous note that the Quadcopter hardware that has been assembled will still be conveyed in this study. The hardware assembly aims to obtain the model parameters of the Quadcopter. The following are the specifications of the Quadcopter that have been assembled.

The measuring Quadcopter mass using hanging scales. Quadcopter mass measurement is useful for simulations to be implemented. It is known that the mass of the Quadcopter is one kilogram (Figs. 1 and 2).

The value of the moment of inertia on each axis can be calculated assuming the mass of the ball Ms at the center or radius R and the point mass mm at the end of the arm at a distance l from the center of mass [15] (Fig. 3).

Fig. 1. Quadcopter Mass Measurement

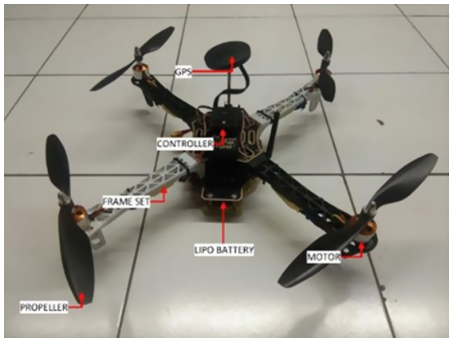

Fig. 2. Quadcopter Body

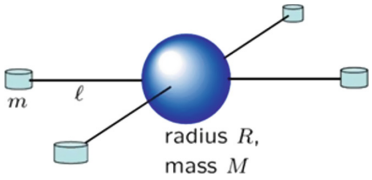

Fig. 3. Inertia Calculation Schematic [7]

So that the calculation of the moment of inertia on the x, y, and z axes is obtained as follows

$$\begin{aligned} I_x &= \frac{2MR^2}{5} + 2ml^2 \\ I_y &= \frac{2MR^2}{5} + 2ml^2 \\ I_z &= \frac{2MR^2}{5} + 4ml^2 \end{aligned} \tag{1}$$

The value of the modeling parameters used in the Quadcopter is obtained from the calculation of the specifications of each component used. Here are the parameters.

Table 1. Assembled Quadcopter Specifications.

No.	Parameter	Symbol	Value
1.	Massa *Quadcopter*	m	1.3 kg
2.	Panjang lengan	l	0.23 m
3.	Koefisien Inersia Sumbu x	I_{xx}	0.01557 kg. m^2
4.	Koefisien Inersia Sumbu y	I_{yy}	0.01557 kg. m^2
5.	Koefisien Inersia Sumbu z	I_{zz}	0,02098 kg. m^2
6.	Koefisien *thrust*	b	9.56×10^{-6} N. s^2
7.	Percepatan gravitasi	g	9.81 m/s^2
8.	Rotor Inertia	Jr	1.713e-5 N. ms^2

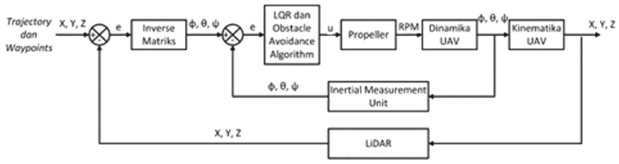

Fig. 4. Quadcopter Control System Block Diagram

2.2 Obstacle Avoidance Control System Design on Quadcopter

The design of the obstacle avoidance control system on the Quadcopter is carried out to integrate the LQR Quadcopter control method with the Obstacle Avoidance Algorithm. In designing the obstacle avoidance control system on the Quadcopter, it is divided into two stages, namely the design of the control system and the design of the algorithm for Obstacle Avoidance (Fig. 4).

In the design of the Quadcopter control system, it is carried out to assemble the control system starting from collecting all the data in the circuit such as the specification of the Quadcopter shape, the specification of the motor as a driver and the specification of the control circuit. The system programming in the Quadcopter control was continued in this study using the LQR method as the control on the Quadcopter. After data collection and control modeling on the Quadcopter, a series of auxiliary electrical components is carried out on the Quadcopter body, this can also be run in parallel in the design of the Quadcopter control system. Then a trial operation of the Quadcopter control system was carried out, this can also be run in parallel. In designing the algorithm for the Obstacle Avoidance Algorithm Quadcopter, it aims to be a decision-making system in the Quadcopter control system so that the Quadcopter can make decisions when there are obstacles. In the design of this system, sensor specification data is collected, and data is collected for programming algorithms. After the control system has been designed and integrated, then the quadcopter control test is carried out so that it can avoid obstacles with decision- making algorithms and methods that have been designed for the Quadcopter control system. The Quadcopter LQR controller will provide U2,

U3 and U4 input signals, respectively, which are signals for the Quadcopter's roll, pitch, and yaw control.

2.3 State Space Quadcopter Modeling

In preparing the Quadcopter modeling, the researchers obtained from Newton's second law, the equations of motion, kinematics and dynamics of the Quadcopter. Quadcopter modeling requires a state space model where the state space is a representation of input and output. Here is the state vector of the Quadcopter.

$$x = \underbrace{\begin{bmatrix} \phi & \theta & \psi & p & q & r & u & v & w & x & y & z \end{bmatrix}^T}_{\in \mathbb{R}^{12}} \tag{2}$$

$$\dot{x} = \underbrace{\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} & \dot{p} & \dot{q} & \dot{r} & \dot{u} & \dot{v} & \dot{w} & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T}_{\in \mathbb{R}^{12}} \tag{3}$$

$$\begin{cases} \dot{\phi} = p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - r[s(\phi)] \\ \dot{\psi} = r\dfrac{c(\phi)}{c(\theta)} + q\dfrac{s(\phi)}{c(\theta)} \\ \dot{p} = \dfrac{I_y - I_z}{I_x} rq + \dfrac{T_x + T_{\omega x}}{I_x} \\ \dot{q} = \dfrac{I_z - I_x}{I_y} pr + \dfrac{T_y + T_{\omega y}}{I_y} \\ \dot{r} = \dfrac{I_x - I_y}{I_z} pq + \dfrac{T_z + T_{\omega z}}{I_z} \\ \dot{u} = rv - q\omega - g[s(\theta)] + \dfrac{J_{\omega x}}{m} \\ \dot{v} = p\omega - ru + g[s(\phi)c(\theta)] + \dfrac{f_{\omega y}}{m} \\ \dot{\omega} = qu - pv + g[c(\theta)c(\phi)] + \dfrac{f_{\omega z} - f_t}{m} \\ \dot{x} = \omega[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)] \\ \dot{y} = v[c(\phi)s(\psi) + s(\phi)s(\psi)s(\theta)] - \omega[c(\psi)s(\phi) - c(\phi)c(\psi)] \\ \dot{z} = \omega[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \end{cases} \tag{4}$$

In this study, the Quadcopter does not have interference from both external and internal Quadcopter forces so that air disturbance and drag factor are ignored. So that we can get the linearization of the model as follows:

$$\dot{x} = Ax + Bu + Dd \tag{5}$$

$$y = Cx + Du \tag{6}$$

It is known that *A*, *B*, *C and D* are state matrix, input matrix, output matrix, and direct transmission matrix, while *x*, *u*, *and y* are state vector, input vector and output vector, respectively. The state *x* vector in quadcopter modeling is taken in the equation as follows

$$u = \left(U_1, U_2, U_3, U_4, U_x, U_y\right)^T \tag{7}$$

$$U_x = \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \tag{8}$$

$$U_y = \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \tag{9}$$

Then, some of the state vectors in the equation can be transformed into $[p\ q\ r]\ T = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]\ T$, then lowered with time to $[\dot{p}\ \dot{q}\ \dot{r}]\ T = [\ddot{\phi}\ \ddot{\theta}\ \ddot{\psi}]$ and external disturbance forces are ignored. Thus, the state-space model of the quadcopter dynamic system is obtained at the reference coordinates of the earth-fixed frame which is shown by the following equation.

$$\begin{cases} \ddot{X} = -\frac{f_t}{m}\left[C_\phi S_\theta C_\psi + S_\phi S_\psi\right] \\ \ddot{y} = -\frac{f_t}{m}\left[S_\psi S_\theta C_\phi - C_\psi S_\phi\right] \\ \ddot{z} = g - \frac{f_t}{m}\left[C_\theta C_\phi\right] \\ \ddot{\phi} = \left[\frac{I_{yy}-I_{zz}}{I_{xx}}\right]\dot{\theta}\dot{\psi} + \frac{\tau_{Mx}}{I_{xx}} \\ \ddot{\theta} = \left[\frac{I_{zz}-I_{xx}}{I_{yy}}\right]\dot{\psi}\dot{\phi} + \frac{\tau_{My}}{I_{yy}} \\ \ddot{\psi} = \left[\frac{I_{xx}-I_{yy}}{I_{zz}}\right]\dot{\phi}\dot{\theta} + \frac{\tau_{Mz}}{I_{zz}} \end{cases} \tag{10}$$

This mathematical model was then linearized with the aim of obtaining an effective LQR control design, referring to research [17]. The linearization process is carried out by selecting a stable point in the quadcopter operation [18]. This stable point is chosen based on the hovering position of the quadcopter which is the quadcopter's position when it can hover with a lift force equal to the force caused by the acceleration of gravity [19]. When in a stable condition, the quadcopter experiences the following conditions.

$$\left\{ \begin{array}{c} \theta = \phi = \psi = \dot{\theta} = \dot{\phi} = \dot{\psi} = \ddot{\theta} = \ddot{\phi} = \ddot{\psi} = 0 \\ \dot{x} = \ddot{x} = \dot{y} = \ddot{y} = \dot{z} = \ddot{z} = 0 \\ w_{\mathrm{r}} = w_{\mathrm{r,\ hover}} \\ \dot{w}_{\mathrm{r}} = \dot{w}_{\mathrm{r}} = 0 \end{array} \right\} \tag{11}$$

Furthermore, to apply linearization to the state vector in the equation, an approach is carried out by substituting the conditions in the equation into equation [19]. So that

the quadcopter state vector is obtained as follows.

$$\dot{x} = \begin{cases} \ddot{\phi} = \frac{lU_2}{I_{xx}} \\ \ddot{\theta} = \frac{lU_3}{I_{yy}} \\ \ddot{\psi} = \frac{U_4}{I_{xx}} \\ \ddot{x} = \frac{U_1}{m} U_x \\ \ddot{y} = \frac{U_1}{m} U_y \\ \ddot{z} = \frac{U_1}{m} - g \end{cases} \tag{12}$$

Furthermore, to minimize calculations, state reduction techniques are also used to simplify mathematical calculations (Okyere et al., 2019). With this technique the state vector in the equation will be reduced to 3 parts, namely the orientation control represented by the vector $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$, position control represented by vector $(x, \dot{x}, y, \dot{y})^T$, and height control which is represented by vector $(z, \dot{z})$ as in the following equation.

$$x = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = (\text{Orientasi, Posisi})^T \tag{13}$$

So the state-space equation for the orientation control system in equation format will be:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{l}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{l}{I_{yy}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{14}$$

Meanwhile, the state-space equation for the position control system in the equation format will be

$$\begin{bmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix} = \begin{bmatrix} 0\,1\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{U_1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{U_1}{m} \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix} \tag{15}$$

Then the state-space equation for the altitude control system in the equation format will be:

$$\begin{bmatrix} \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} 0\,1 \\ 0\,0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [U_z] \tag{16}$$

2.4 Design of Linear Quadratic Regulator (LQR) Control

In this study, researchers used a trial and error method on the weights used to get a quadcopter that was able to reach the set point. In this study, the Q matrix has a dimension of 6 × 6 and is determined through the trial and error method to obtain a satisfactory response. The R matrix itself has a dimension of 3 × 3 where the value is set to a diagonal matrix with a value of 1. The following is a matrix of one of the Q matrix candidates used for the LQR controller in this study.

$$Q_{attitude} = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,10\,0 \\ 0\,0\,0\,0\,0\,1 \end{bmatrix}, R_{attitude} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} \tag{17}$$

from state space modeling in the equation so that the K matrix is obtained with the following values.

$$K = \begin{bmatrix} 1 & 1.066 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.1623 & 1.1951 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1.0208 \end{bmatrix} \tag{18}$$

2.5 Obstacle Avoidance Design

In this study, Obstacle Avoidance was designed using an obstacle avoidance algorithm based on obstacle detection using a LiDAR sensor. LiDAR emits laser light to the area in front of it, when there is an obstacle detected by the returning laser beam. From the reflected data, a point cloud will be generated that contains information on the position of the obstacle relative to the LiDAR sensor [20]. The Obstacle Avoidance Algorithm is designed based on the estimated length and width of the obstacles detected by LiDAR.

To get the maximum and minimum position of the obstacle area, the actual quadcopter position is needed (*xact*, *yact*) in the plane of inertia and relative distance (*xpt*, *ypt*) obstacle obtained from the point cloud generated by the LiDAR sensor. The calculation of the obstacle position is carried out with the following equation:

$$x_{obs,min} = x_{pt,min} + x_{act} \tag{19}$$

$$x_{obs,max} = x_{pt,max} + x_{act} \tag{20}$$

$$y_{obs,min} = y_{pt,min} + y_{act} \tag{21}$$

$$y_{obs,max} = y_{pt,max} + y_{act} \tag{22}$$

With this value, the value of the new setpoint can be calculated using the following equation:

$$x_{sp,baru} = \begin{cases} x_{obs,min} - 3, \; x_{pt,min}^2 > x_{pt,max}^2 \\ x_{obs,max} + 3, \; x_{pt,min}^2 \leq x_{pt,max}^2 \end{cases} \tag{23}$$

$$y_{sp,baru} = \begin{cases} y_{obs,min} - 3, \; y_{pt,min}^2 > y_{pt,max}^2 \\ y_{obs,max} + 3, \; y_{pt,min}^2 \leq y_{pt,max}^2 \end{cases} \tag{24}$$

2.6 Simulation of Control System and Obstacle Avoidance Quadcopter

In this study, a simulation was carried out using MATLAB/Simulink 2022a software with a mathematical model based on Eqs. (18) and (19) with the values of the quadcopter parameters listed in Table 1. There are several scenarios given for the control system simulation as follows. Testing by providing input setpoint step, Testing trajectory tracking in the form of giving setpoints that will continue to change over time the simulation. Tests to avoid obstacles by providing several destination waypoints and obstacles between those waypoints.

In MATLAB/Simulink 2022a the UAV toolbox is used which can provide a simulation environment that meets needs with an intuitive use. In the UAV toolbox, there is a Simulation 3D Scene Configuration block that can configure the simulation by providing a quadcopter model and providing obstacles for simulation needs. With block simulation 3D scene configuration, giving quadcopter models and obstacles can be done with the uav Plat form and add Mesh commands in MATLAB. The following is the simulation configuration that will be used in this study (Fig. 5).

In addition, the UAV toolbox also provides a virtual LiDAR sensor that can be used for obstacle avoidance requirements in the simulation carried out. The virtual sensor can provide a point cloud output which can be if LiDAR detects an obstacle that provides a distance re configured to attach to the quadcopter model where the virtual sensor block is shown in the following (Fig. 6).

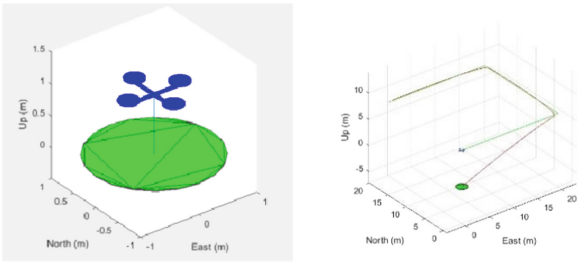

Fig. 5. Quadcopter model (blue) on simulation and simulation display

Fig. 6. Virtual LiDAR Sensor Block on MATLAB/Simulink

3 Results and Discussion

3.1 Trajectory and Obstacle Avoidance Test on Quadcopter

1. Quadcopter Trajectory Test

This test is conducted to review the performance of the proposed control system. The test is carried out by given step input as the state for the linear position of the *x*, *y*, *and z axes*. With trajectory conditions with obstacles. The response of the results of this step input test is illustrated by simulation variations in Figs. 7, 8, 9 and 10 of performance are shown in Tables 2, 3, 4 and 5: Variation 1 (Q (3, 3) = 30, 60 dan 90).

The results of the trajectory tracking test are in the table.

The MAE calculation results in Table 3 show that the variation of Q (3, 3) = 90 has a smaller error value compared to other variations with trajectory conditions with obstacles .Variation 3 (Q (6, 6) = 30, 60 dan 90).

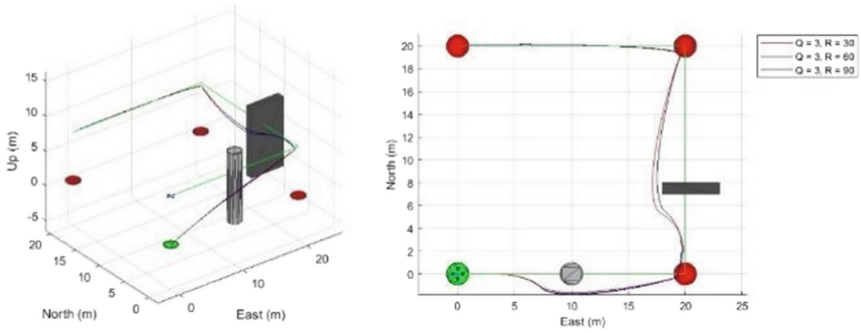

Fig. 7. Variation Simulation 1 (Q (3, 3) = 30, 60 dan 90) Trajectory

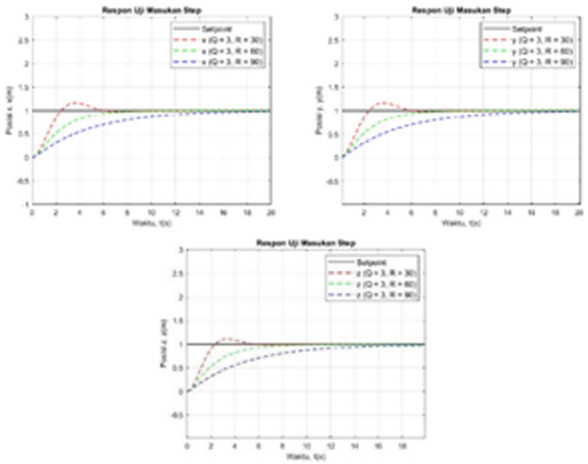

Fig. 8. Graph of Variation 1 response test trajectory step input position x, y and z

Table 2. Performance Variations 1 controller test results of obstacle avoidance

State	*Rise Time (s)*			*Settling Time (s)*			*Max Overshoot %*		
	Q (3, 3) = 30	Q (3, 3) = 60	Q (3, 3) = 90	Q (3, 3) = 30	Q (3, 3) = 60	Q (3, 3) = 90	Q (3, 3) = 30	Q (3, 3) = 60	Q (3, 3) = 90
x	1.73	4.72	9.97	8.12	8.42	16.22	0.21	0.0000	0.0000
y	1.73	4.72	9.97	8.12	8.42	16.22	0.21	0.0000	0.0000
z	1.66	4.96	10.14	5.29	8.89	16.37	0.16	0.0000	0.0000

Table 3. Performance Variations 1 controller test results and MAE obstacle avoidance

State	*MAE*		
	Q (3, 3) = 30	Q (3, 3) = 60	Q (3, 3) = 90
x	0.974	0.932	0.793
y	0.047	0.032	0.025
z	0.583	0.429	0.455

The results of the trajectory tracking test are in the table.

The MAE calculation results in Table 5 show that the second variation shows that the variation (Q (6, 6) = 90) has the lowest value compared to other variations. And compared to the first variation, Q (6.6) = 90 has the smallest error value compared to the other six variations.

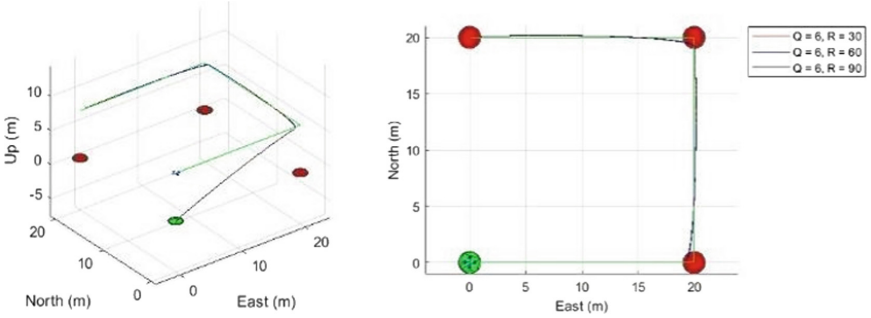

Fig. 9. Variation Simulation 2 (Q (6, 6) = 30, 60 dan 90) Trajectory

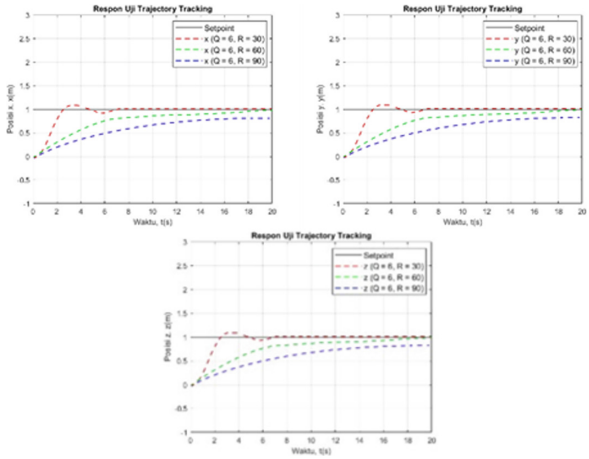

Fig. 10. Graph of Variation 2 response test trajectory step input position x, y and z

Table 4. Performance Variations 2 controller test results of obstacle avoidance

State	*Rise Time (s)*			*Settling Time (s)*			*Max Overshoot %*		
	Q (6, 6) = 30	Q (6, 6) = 60	Q (6, 6) = 90	Q (6, 6) = 30	Q (6, 6) = 60	Q (6, 6) = 90	Q (6, 6) = 30	Q (6, 6) = 60	Q (6, 6) = 90
x	1.16	5.02	11.07	5.94	8.32	16.34	0.11	0.0000	0.0000
y	1.16	5.02	11.07	5.94	8.32	16.34	0.11	0.0000	0.0000
z	1.21	5.16	11.16	5.87	8.39	16.37	0.06	0.0000	0.0000

Table 5. Performance Variations 2 controller test results and MAE obstacle avoidance

State		MAE	
	Q (6, 6) = 30	Q (6, 6) = 60	Q (6, 6) = 90
x	0.829	0.646	0.443
y	0.056	0.031	0.022
z	0.473	0.374	0.285

4 Conclusion

Based on the results of research, the researchers came to the following conclusions: A quadcopter UAV control system that can fly autonomously and can follow the trajectory well. UAV LiDAR Scenario as a LiDAR Sensor simulation allows the Quadcopter to detect obstacles in front of it with Max Range = 7 (meters), Azimuth Limits = [−179 179] and Elevation Limits = [−20 20], according to the basic settings of the Lidar Scenario UAV block. The quadcopter UAV can fly following the trajectory and detect and avoid obstacles autonomously. The performance of the LQR control system in the obstacle avoidance test variation of Q (6, 6) = 90 shows the rise time performance (s) $x = 1.21$, $y = 5.16$ and $z = 11.16$. Settling time (s) $x = 5.87$, $y = 8.39$ and $z = 16.37$. Overshoot (%) $x = 0.06$, $y = z = 0$. And MAE (%) $x = 0.443$, $y = 0.022$ and $z = 0.285$.

Acknowledgment. The authors gratefully acknowledge the contributions of the Instrumentation, Control, and Optimization Laboratory of Sepuluh Nopember Institute of Technology (ITS)'s team and parties for supporting any software and computer requirements to meet the needs of this research.

References

1. Fahlstrom, P., and Gleason, T.: Introduction to UAV systems. In: Fourth Edition, John Wiley & Sons (Eds.) (2012).
2. Yedavalli, R. K.: Flight Vehicle Dynamics and Control. John Wiley & Sons (Eds.), The Ohio State University, USA (2020).
3. Rezende, A. M. C., Miranda, V. R. F., Machado, H. N., Chiella, A. C. B., Gonçalves, V. M., and Freitas, G. M.: Autonomous System for a Racing Quadcopter. 2019 19th International Conference on Advanced Robotics (ICAR). pp. 1-6 (2019).
4. Perez, A., Bhandari, S., Green, N., Cobian, C., Abarca, M., and Moffatt, A.: Obstacle Detection and Avoidance System for a UAV using LIDAR (2020).
5. Dharmawan, A., Putra, A. E., Tresnayana, I. M., and Wicaksono, W. A.: The Obstacle Avoidance System in A Fixed-Wing. 2019 International Conference of Computer Engineering, Network, and Intelligent Multimedia (CENIM) (2019).
6. Aldao, E., González-deSantos, L. M., Michinel, H., and González-Jorge, H.: UAV Obstacle Avoidance Algorithm to Navigate in Dynamic Building Environments. Drones 2022, p. 6(1) (2022).

7. Beard, R. W.: Quadrotor dynamics and control. Brigham Young University 19, pp. 46-56 (2008).
8. Kumar, A., Sharma, S., and Mitra, R.: Design of type-2 fuzzy controller based on LQR mapped fusion function. International Journal of Intelligent Systems and Applications. (2012).
9. Wang, P., Man, Z., Cao, Z., Zheng, J., and Zhao, Y.: Dynamics modelling and linear control of quadcopter. In 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 498-503 (2016).
10. Okyere, E., Bousbaine, A., Poyi, G. T., Joseph, A. K., and Andrade, J. M.: LQR controller design for quad-rotor helicopters. The Journal of Engineering 2019, pp. 4003-4007 (2019).
11. Matlab: Matlab UAV Toolbox Documentation). 1 Apple Hill Drive Natick, MA 01760-2098: The MathWorks, Inc (2022).