State Feedback Controller Design by Utilizing Linear Quadratic Regulator and Backstepping Control for UAV Quadrotor

Purwadi Agus Darwito[1](✉), Linggar Handy Swandana[1], Hermawan Nugroho[2], and Totok Ruki Biyanto[1]

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

[2] Department of Electrical and Electronic Engineering, University of Nottingham Malaysia, Selangor, Malaysia
hermawan.nugroho@nottingham.edu.my

Abstract. One of the popular UAV applications in the logistics sector is the use of UAVs as couriers to deliver goods to customers' homes. In this application, the UAV must be able to follow the given trajectory while avoiding the existing obstacles to get to the customer's house. To fulfill this, UAVs require an autonomous trajectory tracking and obstacle avoidance system. One of the control systems commonly used in UAVs is to use a linear-quadratic regulator (LQR) and backstepping control (BSC) and utilize LiDAR as an obstacle detector. With the help of MATLAB, simulations of open loop, close loop, trajectory tracking and obstacle avoidance were carried out. The close loop test shows a steady state error of 0.000024 m and 0 m for orientation control and position control. The performance of the LQR-BSC control system in the trajectory tracking simulation shows the MAE performance values of 0.1429, 0.0015 and 0.1267 for position control. While the simulation of obstacle avoidance using the LQR-BSC control system and the LiDAR sensor as detection shows that the quadcopter can avoid the obstacles given.

Keywords: Backstepping Control · LiDAR · Linear-Quadratic Regulator · Obstacle Avoidance · Quadcopter

1 Introduction

Unmanned Aerial Vehicle (UAV) is one of the classifications of aircraft that can fly autonomously without the need for a pilot. UAV is a system consisting of aircraft components, sensors, and ground control station. The flight attitude of the UAV can be controlled by electronic devices embedded in the UAV itself automatically or controlled through the ground control station. At first UAV was more often used for military needs.

S. Sugiman et al. (Eds.): MIMSE-M-E 2022, AER 216, pp. 89–101, 2023.
https://doi.org/10.2991/978-94-6463-078-7_11

However, along with the times, UAVs are widely used for various needs in various fields including logistics, agriculture, remote sensing, to smart city applications. One of the popular factors in the use of UAVs is because UAVs are autonomous. The UAV has an embedded control system so that the UAV can operate without any human intervention. The UAV can control its flight attitude based on the sensors installed on the UAV in question. One of these sensors is the Inertia Measurement Unit (IMU) which is used to monitor the flying attitude of the UAV. With only the IMU sensor, with the help of the kalman filter, the parameters needed for UAV control can be estimated. In addition, the UAV can also detect obstacles such as billboards when flying in the air. Detection of these obstacles can be done by using a monocular camera [1]. In addition to utilizing monocular cameras, the use of LiDAR for UAV environmental sensing is also commonly used where the LiDAR sensor has the advantage of distance estimation and can operate in low light environments [2]. In previous research on quadcopter-type UAV control systems, one of the commonly used control systems is the linear-quadratic regulator (LQR), which is the optimal control by utilizing the quadratic cost function [3]. In practice, control systems using LQR can also be combined with other control systems. The control system that utilizes a combination of LQR and sliding mode control shows better performance in controlling quadrotor UAVs compared to using only LQR [4]. In addition to using LQR, the practice of using backstepping control (BSC) is also quite commonly used in quadcopter control because the BSC is robust and can guarantee system stabilization [5].

2 System Modelling and Control System Design

2.1 Quadcopter Modelling

System modeling is carried out to obtain a mathematical model of the quadcopter to be controlled. In the proposed control system, a state-space model of quadcopter dynamics is needed in which the general equation of the state-space model itself is formulated in Eq. (1) [6].

$$\dot{x} = Ax + Bu \tag{1}$$

Where A and B are state matrices, input matrices. While the matrix x, and matrix u are state vectors and input vectors.

The state x vector in quadcopter modeling itself is shown in Eq. (2).

$$x = (\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z})^T \tag{2}$$

While the input vector u itself is shown in Eq. (3).

$$u = \left(U_1, U_2, U_3, U_4, U_x, U_y\right)^T \tag{3}$$

Based on the quadcopter dynamics that have been developed previously, the state-space model of the quadcopter can be formulated as in Eq. (4) [7].

$$\dot{x} = \begin{cases} \ddot{\phi} = \frac{lU_2}{I_{xx}} - \frac{\dot{\theta} J_r}{I_{xx}} w_r + \dot{\psi}\dot{\theta}\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right) \\ \ddot{\theta} = \frac{lU_3}{I_{yy}} - \frac{\dot{\phi} J_r}{I_{yy}} w_r + \dot{\psi}\dot{\theta}\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right) \\ \ddot{\psi} = \frac{U_4}{I_{xx}} + \dot{\phi}\dot{\theta}\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right) \\ \ddot{x} = \frac{U_1}{m} U_x \\ \ddot{y} = \frac{U_1}{m} U_y \\ \ddot{z} = \frac{U_1}{m}(\cos(\phi)\cos(\theta) - g \end{cases} \tag{4}$$

The mathematical model in Eq. (4) is then linearized by determining the stable point of quadcopter operation [8]. The 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 [9]. When in a stable condition, the quadcopter experiences a state as shown in Eq. (5).

$$\left\{ \begin{array}{c} \theta = \phi = \psi = \dot{\theta} = \dot{\phi} = \dot{\psi} = \ddot{\theta} = \ddot{\phi} = \ddot{\psi} = 0 \\ \dot{x} = \ddot{x} = \dot{y} = \ddot{y} = \dot{z} = \ddot{z} = 0 \\ w_r = w_{r,\text{hover}} \\ \dot{w}_r = \ddot{w}_r = 0 \end{array} \right\} \tag{5}$$

Furthermore, to apply linearization to the state vector, an approach is carried out by substituting the conditions in Eq. (5) to Eq. (4) so that a linear model of the quadcopter in Eq. (6) is produced [10].

$$\dot{x} = \begin{cases} \ddot{\phi} = \frac{lU_2}{I_{xx}} \\ \ddot{\theta} = \frac{lU_3}{I_{yy}} \\ \ddot{\psi} = \frac{U_4}{I_{xx}} \\ \ddot{x} = \frac{U_1}{m} U_x \\ \ddot{y} = \frac{U_1}{m} U_y \\ \ddot{z} = \frac{U_1}{m} - g \end{cases} \tag{6}$$

Furthermore, to minimize calculations, state reduction techniques are also used to simplify mathematical calculations [10]. With this technique the state vector in Eq. (6) will be reduced to so that only the orientation vector is represented by $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})^T$, So that the state-space equation for the orientation control system in the format of Eq. (1)

Table 1. Quadcopter Modelling parameter.

Parameter	Values	Unit
Arm length (l)	0.23	Meter
Inertia coefficient (l_{xx})	0.01557	Kg. m^2
Inertia coefficient (l_{yy})	0.01557	Kg. m^2
Inertia coefficient (l_{xx})	0.02098	Kg. m^2
Mass (m)	1.3	Kg
Gravity (g)	9.81	m/s^2

will be Eq. (7).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{t} & 0 & 0 \\ l_{xx} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{l}{t_{yy}} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{l_{yn}} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{7}$$

The values of the modeling parameters used alone are presented in Table 1 as follows.

2.2 Control System Design

LQR requires a diagonal matrix Q and R, each of which serves to determine the weight of the state as well as the control signal on the cost function as Eq. (8) [11].

$$I = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{8}$$

In this study, the Q matrix is an identity matrix that has 6 × 6 dimensions as in Eq. (9). While the matrix R is also an identity matrix which has 3 × 3 dimensions as in Eq. (10).

$$Q = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,10\,0 \\ 0\,0\,0\,0\,0\,1 \end{bmatrix} \tag{9}$$

$$R = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} \tag{10}$$

For the backstepping control, the height of the quadcopter will be controlled with a U1 signal. The dynamic equation for the quadcopter height is taken from Eq. (6) so that Eq. (11).

$$\begin{cases} \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \frac{U_1}{m} - g \end{cases} \tag{11}$$

In addition, it also takes the error of the height e11 which is defined as Eq. (12).

$$e_{11} = x_{11d} - x_{11} \tag{12}$$

Where x_{11d} is the quadcopter altitude setpoint.

In backstepping control, the Lyapunov equation is needed which in this study is chosen the positive Lyapunov equation which is formulated in Eq. (13) [12].

$$\mathrm{V}_{11} = \frac{1}{2} e_{11}^2 \tag{13}$$

Which if the Lyapunov equation is derived with respect to time, we get Eq. (14).

$$\dot{\mathrm{V}}_{11} = e_{11}(\dot{x}_{11d} - x_{12}) \tag{14}$$

To make the system stable, the derivative of the Lyapunov equation must be negative [13]. To meet these requirements, a virtual input x_{12} is created which is formulated in the Eq. (15).

$$x_{12} = \dot{\mathrm{x}}_{11\mathrm{d}} + c_{11} e_{11} \tag{15}$$

Where c is a positive constant. With the virtual control input in Eq. (15), to make the system stable the derivative of the Lyapunov function in Eq. (14) must meet the requirements as inequality (16).

$$\dot{\mathrm{V}}_1 = \mathrm{e}_{11}(\dot{x}_{11d} - x_{12}) \leq -\mathrm{c}_{11} e_{11}^2, \quad c_{11} > 0 \tag{16}$$

e_{12} error is also defined which is formulated as equation

$$e_{12} = x_{12} - \dot{x}_{11d} - c_{11} e_{11} \tag{17}$$

With e_{12}, Eq. (14) can be redefined into Eq. (18).

$$\dot{\mathrm{V}}_{11} = -\mathrm{e}_{11} e_{12} - c_{11} e_{11}^2 \tag{18}$$

Then the second Lyapunov equation is defined which is formulated in Eq. (19) [14].

$$V_{12} = V_{11} + \frac{1}{2} e_{12}^2 \tag{19}$$

$$\dot{V}_{12} = -e_{11} e_{12} - c_{11} e_{11}^2 + e_{12}(\dot{x}_{12} - \ddot{x}_{11d} - c_1 e_{11}) \tag{20}$$

If Eq. (11) is substituted into Eq. (20), then Eq. (21) will be obtained as follows

$$\dot{V}_{12} = -e_{11}e_{12} - c_1 e_{11}^2 + e_{11}\left(\frac{U_1}{m} - g - \ddot{x}_{11d} - c_{11}e_{11}\right) \tag{21}$$

From Eq. (21), the control input U1 can be taken to satisfy the inequality (22) [14].

$$\dot{V}_{12} = -c_{11}e_{11}^2 - c_{12}e_{12}^2 < 0 \tag{22}$$

So that the equation U_1 is obtained which is written in Eq. (23).

$$U_1 = m(e_{11} + g + \ddot{x}_{11d} + c_{12}e_{12} + c_{12}(\dot{x}_{11d} - x_{12})) \tag{23}$$

For controlling the X position and Y position, the same thing as Eq. (11) to (23) is carried out so that the U_x and U_y equations are produced as in Eqs. (24) and (25).

$$U_x = \frac{m}{U_1}(e_7 + \ddot{x}_{7d} + c_8 e_8 + c_7(\dot{x}_{7d} - x_8)) \tag{24}$$

$$U_y = \frac{m}{U_1}(e_9 + \ddot{x}_{9d} + c_{10}e_{10} + c_9(\dot{x}_{9d} - x_{10})) \tag{25}$$

2.3 Obstacle Avoidance Algorithm Design

The obstacle avoidance algorithm used in this study is based on the detection of obstacles using the LiDAR sensor. When lidar emits a laser, when there is an obstacle, lidar will detect the obstacle based on the reflection of the laser beam back. From the reflected data, tcloud points will be generated which contain information on the position of the obstacle relative to the LiDAR sensor. Based on this information, the obstacle avoidance algorithm is designed based on the estimated length and width of the obstacles detected by LiDAR. The flow of the data processing process using the LiDAR sensor is shown in the flow chart in Fig. 1.

To get the maximum and minimum obstacle positions, the actual quadcopter positions of xact and yact are needed. The calculation of the obstacle position is carried out with Eq. (26) to Eq. (29) as follows.

$$x_{obs,min} = x_{pt,min} + x_{act} \tag{26}$$

$$x_{obs,max} = x_{pt,max} + x_{act} \tag{27}$$

$$y_{obs,min} = y_{pt,min} + y_{act} \tag{28}$$

$$y_{obs,max} = y_{pt,max} + y_{act} \tag{29}$$

With this value, the value of the new setpoint can be generated to avoid obstacles by using Eq. (30) and Eq. (31).

$$x_{av} = \begin{cases} x_{abs,min} - 3, x_{act} - x_{abs,min} \leq x_{abs,max} - x_{act} \\ x_{abs,max} + 3, x_{act} - x_{abs,min} \geq x_{abs,max} - x_{act} \end{cases} \tag{30}$$

$$x_{av} = \begin{cases} x_{abs,min} - 3, x_{act} - x_{abs,min} \geq x_{abs,max} - x_{act} \\ x_{abs,max} + 3, x_{act} - x_{abs,min} \leq x_{abs,max} - x_{act} \end{cases} \tag{31}$$

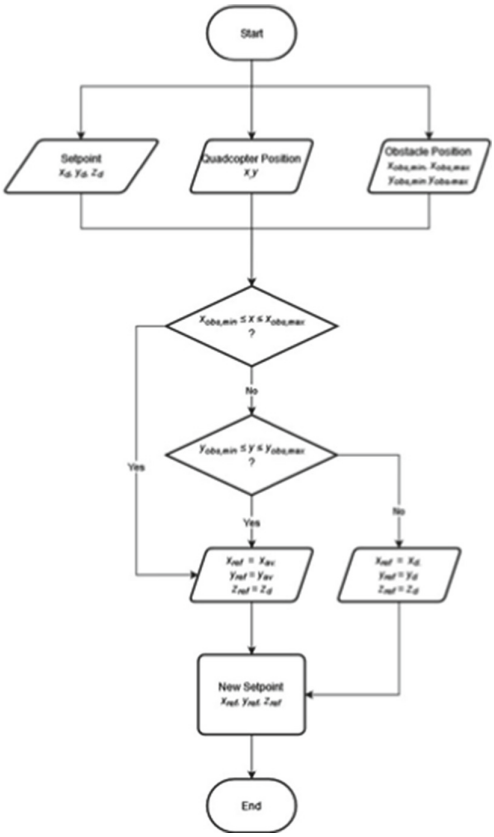

Fig. 1. Obstacle avoidance flowchart.

3 Result

3.1 Close Loop Test

Close loop testing is carried out to review the performance of the proposed control system. The input step is given as the desired setpoint for the orientation and position of the quadcopter. In the close loop test for position control, the test is carried out with variations in the gain values of c7, c9, and c11 so that it can be seen the effect of differences in performance from each difference in the value of the gain constant of the proposed position control system. Figure 2 is a position response graph from the close loop test results with the test performance shown in Table 2.

In the close loop test for the quadcopter orientation controller, a test variation is given in the form of weighting the orientation state on the Q matrix as in Eq. (9) which is given a weighting variation on the values of Q(1,1), Q(3,3) and Q(5, 5) of 1, 5, and 10 so that it can also be seen the effect of differences in performance from each difference in the state weight values on the Q matrix for the LQR controller of the proposed orientation control

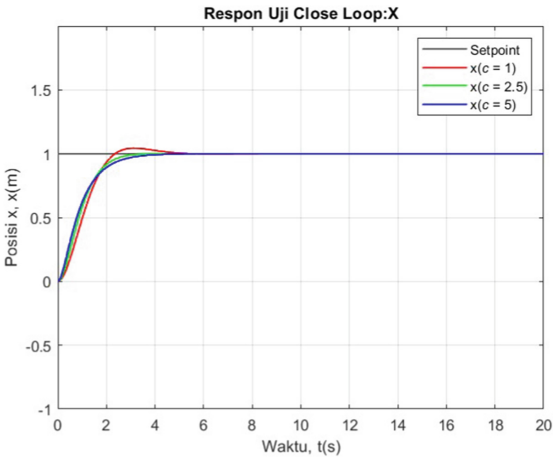

Fig. 2. The response graph of the quadcopter *X* position of the close loop test results.

Table 2. Performance of quadcopter attitude control of close loop test results.

State	Performance Criteria								
	Rise time (s)			*Settling Time* (rad)			*Steady State Error* (rad)		
	$Q = 1$	$Q = 5$	$Q = 10$	$Q = 1$	$Q = 5$	$Q = 10$	$Q = 1$	$Q = 5$	$Q = 10$
Φ	2	1.09	0.8	4.12	4.56	1.44	0.00007	-0.45	-0.69
Θ	2.31	1.09	0.8	4.12	4.56	1.44	0.00007	-0.45	-0.69
Ψ	2.19	0.98	0.69	3.93	1.77	1.25	0	-0.55	-0.69

Table 3. Performance of quadcopter position control of close loop test results.

State	Performance Criteria											
	Rise time (s)			Settling Time (s)			Max. overshoot (%)			Steady State Error (m)		
	$c = 1$	$c = 2.5$	$c = 5$	$c = 1$	$c = 2.5$	$c = 5$	$c = 1$	$c = 2.5$	$c = 5$	$c = 1$	$c = 2.5$	$c = 5$
x	1.52	2	1.837	4.214	2.724	3.332	4.315	0.0246	0	0	0	0
y	1.52	2	1.837	4.214	2.724	3.332	4.315	0.0246	0	0	0	0
z	1.52	1.629	1.837	4.214	2.724	3.332	4.321	0.0246	0	0	0	0

system. One of the responses from the close loop test results for orientation control is depicted in the graph in Fig. 3 with the performance shown in Table 3.

3.2 Trajectory Tracking Test

This trajectory test is carried out to determine the ability of the proposed position controller to follow the setpoint which will continue to change over time. The setpoint will form a pattern which is shown in Fig. 4. In this test, a variation of the value of the constant c is given which is the same as the close loop test with the test results shown in Fig. 5 and Table 4.

Table 4. Performance of trajectory tracking test results.

State	MAE		
	$c = 1$	$c = 2.5$	$c = 5$
x	0.4416	0.2546	0.1429
y	0.006	0.0029	0.0015
z	0.231	0.1639	0.1267

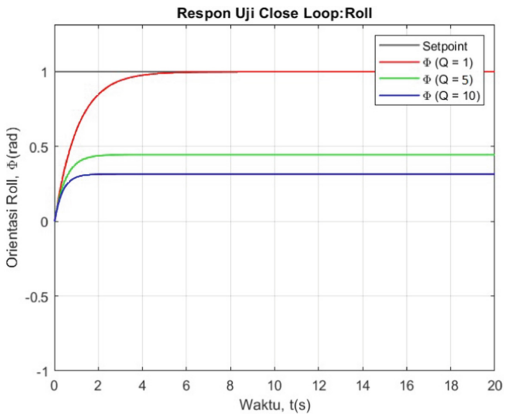

Fig. 3. The response graph of the quadcopter rolls orientation of the close loop test results.

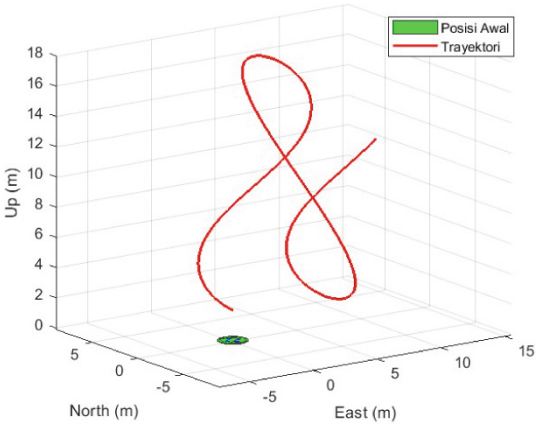

Fig. 4. Quadcopter test trajectory.

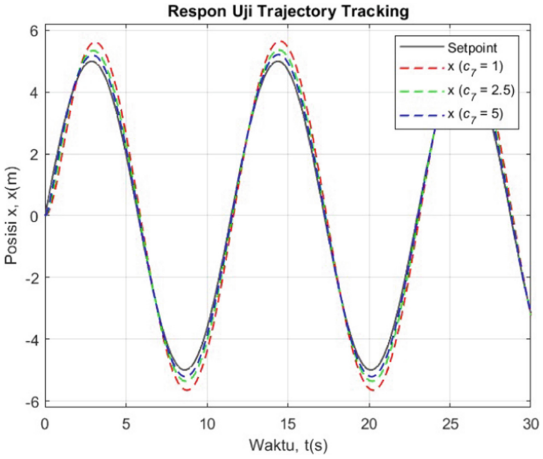

Fig. 5. The response graph of the Quadcopter X position of the trajectory tracking test results.

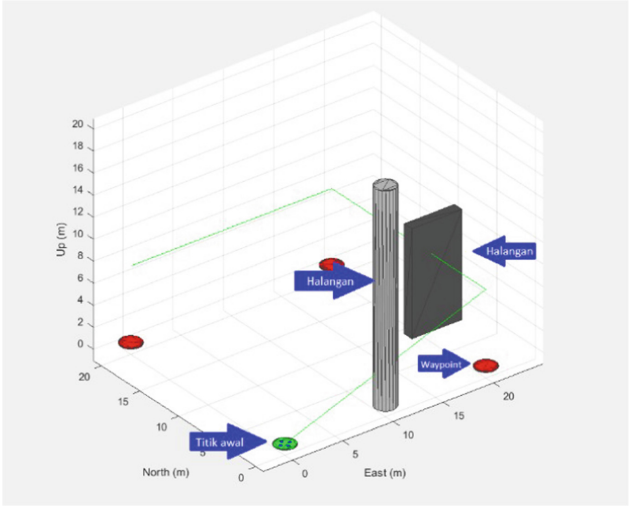

Fig. 6. Obstacle avoidance test environment.

3.3 Obstacle Avoidance Test

In this test, a simulation is carried out to see the performance of the proposed avoidance algorithm. Obstacle avoidance testing is carried out by providing several destination waypoints and obstacles between these waypoints as depicted in Fig. 6.

The test is carried out by varying the value of the backstepping constant c with the same value as in the previous tests. In addition, a variation of the weighting of the matrix Q (5,5) with values of 1, 50, and 125 is also given. The following Fig. 7 shows the response of the quadcopter position as a result of obstacle avoidance testing with a value

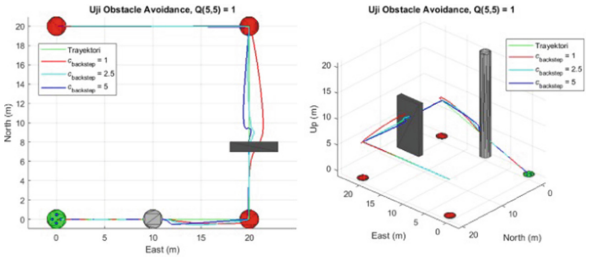

Fig. 7. The result of obstacle avoidance test, Q (5,5) = 1.

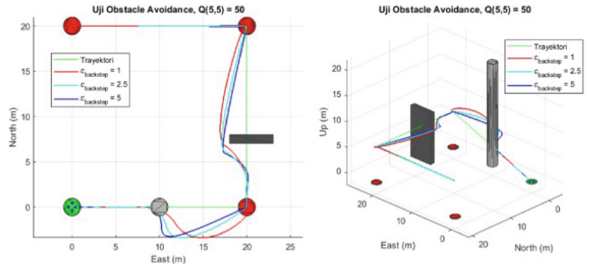

Fig. 8. The result of obstacle avoidance test, Q (5,5) = 50.

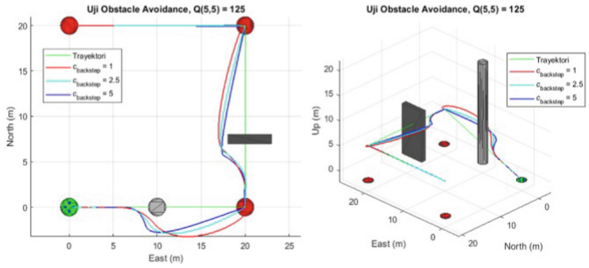

Fig. 9. The result of obstacle avoidance test, Q (5,5) = 125.

of Q (5,5) = 1. In the test with a weighting variation of the matrix Q with a value of 1, the quadcopter cannot avoid obstacles on time as shown in Fig. 7.

The second test with a weighting variation of the matrix Q (5,5) = 50 shown in Fig. 8 resulted in better dodging performance. This is indicated by the success of the quadcopter to avoid field obstacles for each variation of the backstepping constant. Due to the test with the weighted variation of the matrix Q (5,5) showing improved performance, the test was continued with the next variation, namely the weighting matrix Q (5,5) which was given a value of 125 where the response of the variation is shown in Fig. 9. On the weighted matrix Q (5,5) which is given a value of 125 variations of good performance in avoiding existing obstacles. The response of the weighting variation of the matrix Q (5,5) with a value of 125 shows the success of all variations of the position control backstepping constant to avoid obstacles in the path.

4 Conclusion

After the text edit has been completed, the paper is ready the results of the close loop simulation for orientation control show that the performance of the orientation control system using LQR can have a very minimum steady state error of 0.000024 rad with a risetime of 2.31 s for weighting variations on the Q matrix equal to 1. However, it has the disadvantage of not achieving the setpoint if the weighting the Q matrix is given a value of more than 1. While for position control shows the performance of the position control system using the BSC has a steady state error value with a value of 0 m for all test variations with the smallest rise time of 1.52 s given by the variation of the constant c with a value of 1 which at the same time has the highest overshoot with a value of 4.3%. Obstacle avoidance simulation using the LQR-BSC control system and the LiDAR sensor as an obstacle detector shows that all variations of the c constant for backstepping control can avoid the obstacles given by the variation of the weighting matrix Q (5.5) with a value of 125. In future research, popular avoidance algorithms such as RRT* or VFH can be applied so that better obstacle avoidance performance can be obtained.

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. Dai, X., Mao, Y., Huang, T., Qin, N., Huang, D., Li, Y..: Automatic obstacle avoidance of quadrotor UAV via CNN-based learning, Neurocomputing, vol. 402, pp. 346–358 (2020). doi: https://doi.org/10.1016/j.neucom.2020.04.020.
2. Wang, P.: Research on Comparison of LiDAR and Camera in Autonomous Driving, J. Phys.: Conf. Ser., vol. 2093, no.1, p.012032 (2021). doi: https://doi.org/10.1088/1742-6596/2093/1/012032.
3. Ahmad, F., Kumar, P., Bhandari, A., Patil, P. P.: Simulation of the Quadcopter Dynamics with LQR based Control, Materials Today: Proceedings, vol. 24, pp. 326–332 (2020). doi: https://doi.org/10.1016/j.matpr.2020.04.282.
4. Tang, Y.-R., Xiao, X., Li, Y.: Nonlinear dynamic modeling and hybrid control design with dynamic compensator for a small-scale UAV quadrotor, Measurement, vol. 109, pp. 51–64 (2017). doi: https://doi.org/10.1016/j.measurement.2017.05.036.
5. Nguyen, A. T., Xuan-Mung, N., Hong, S.-K.: Quadcopter Adaptive Trajectory Tracking Control: A New Approach via Backstepping Technique, Applied Sciences, vol. 9, no. 18, Art. no. 18 (2019). doi: https://doi.org/10.3390/app9183873.
6. Ogata, K.: System dynamics / Katsuhiko Ogata., 4th ed. Upper Saddle River, NJ: Pearson/Prentice Hall (2004).
7. Elkholy, H. M.: Dynamic modeling and control of a Quadrotor using linear and nonlinear approaches (2014).
8. Wang, P., Man, Z., Cao, Z., Zheng, J., Zhao, Y.: Dynamics modelling and linear control of quadcopter, in 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 498–503 (2016). doi: https://doi.org/10.1109/ICAMechS.2016.7813499.

9. Poyi, G. T.: A novel approach to the control of quad- rotor helicopters using fuzzy-neural networks. University of Derby. [Online]. Available: http://hdl.handle.net/10545/337911 (2014).
10. Okyere, E., Bousbaine, A., Poyi, G. T., Joseph, A. K., Andrade, J. M.: LQR controller design for quad-rotor helicopters. The Journal of Engineering, vol. 2019, no.17, pp. 4003–4007 (2019). doi: https://doi.org/10.1049/joe.2018.8126
11. Ogata, K.: Modern Control Engineering. Prentice Hall (2010).
12. Bouabdallah, S., Siegwart, R.: Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor, in Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2247–2252 (2005). doi: https://doi.org/10.1109/ROBOT.2005.1570447.
13. Sepulchre, R., Jankovic, M., Janković, M., Kokotovic, P. V., Kokotović, P. V.: Constructive Nonlinear Control. Springer London (1997).
14. Tripathi, V. K., Behera, L., Verma, N.: Design of sliding mode and backstepping controllers for a quadcopter," in 2015 39th National Systems Conference (NSC), pp. 1–6 (2015). doi: https://doi.org/10.1109/NATSYS.2015.7489097.