

Trajectory Tracking Control of Quadrotor using LQ-Servo Control with SimMechanics

Fahmizal

Department of Electrical and Information Engineering, Engineering Faculty, Universitas Gadjah Mada

Hanung Adi Nugroho

Department of Electrical and Information Engineering, Engineering Faculty, Universitas Gadjah Mada

Adha Imam Cahyadi

Department of Electrical and Information Engineering, Engineering Faculty, Universitas Gadjah Mada

Ardiyanto, Igi

Department of Electrical and Information Engineering, Engineering Faculty, Universitas Gadjah Mada

<https://doi.org/10.5109/7160931>

出版情報 : Evergreen. 10 (4), pp.2412-2422, 2023-12. 九州大学グリーンテクノロジー研究教育センター

バージョン :

権利関係 : Creative Commons Attribution 4.0 International

Trajectory Tracking Control of Quadrotor using LQ-Servo Control with SimMechanics

Fahmizal^{1,2}, Hanung Adi Nugroho^{1,*}, Adha Imam Cahyadi¹, Igi Ardiyanto¹

¹Department of Electrical and Information Engineering, Engineering Faculty, Universitas Gadjah Mada, Indonesia

²Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Indonesia

*E-mail: adinugroho@ugm.ac.id

(Received December 13, 2022; Revised October 17, 2023; accepted October 22, 2023).

Abstract: The design of a Linear Quadratic (LQ) - Servo for trajectory tracking control of a Quadrotor is presented in this work. In this study, Newton's and Euler's laws are utilized to develop a mathematical model of how the movement of a Quadrotor is simulated. We get a linearized version of the model, so we can make a linear controller called LQ-Servo. The success of LQ-Servo control was tested in two scenarios, the first trajectory tracking at a circular position and the second with the position of the trajectory number "8". The simulation results show that the proposed LQ-Servo controller successfully overcame inertial disturbances when the UAV Bicopter performs trajectory tracking. Based on the results, it is evident that the LQ-Servo control can still maintain Quadrotor trajectory tracking with only a slight increase in RMSE when subjected to inertial disturbances. Finally, the SimMechanics are used to observe the Quadrotor's behavior under the proposed control strategies in virtual reality.

Keywords: Trajectory tracking; Quadrotor; LQ-Servo; SimMechanics

1. Introduction

Due to its maneuverability, the Quadrotor has gained popularity. Quadrotors, 6-DOF (Degrees of Freedom) UAVs, are perfect for search and rescue¹⁻³, surveillance⁴⁻⁶, and long-distance inspection⁷⁻⁹. A Quadrotor is a rotorcraft with four rotors as lift-producing propellers. Quadrotors use a variety of sensors to provide excellent stability and control, allowing them to move in tight spaces. Due to the thrust and moments generated by four rotors, Quadrotors can perform 6 DOF movements. However, controlling Quadrotors remains difficult in terms of reliability, precision, safety, and productivity.

The Quadrotor is capable of linear movement in addition to its other motions, such as rotating. The axis of rotation of the Quadrotor is the point at which the roll, pitch, and yaw motions take place. Furthermore, translational motion is a movement resulting from rotational motion. This motion is comprised of both lateral and longitudinal motion. The three main stages of longitudinal motion are takeoff, hovering, and landing, whereas lateral motion consists of horizontal movement from one location to another (waypoint)^{10,11}. Changes in the Quadrotor's angle of rotation allow for the possibility of lateral movement during hover if the Quadrotor can retain stability. Changes in roll angle will cause translation along the Y axis, whereas changes in pitch angle will

cause translation along the X axis. Thus, lateral Quadrotor motions require an initial rotating motion. Therefore, Quadrotor trajectory tracking control is an essential topic for researchers.

To get the greatest degree of maneuverability, a control system is essential. The flight control system can be divided into three main subcategories when seen from the perspective of the Quadrotor control approach. These categories are learning-based flight control, non-linear flight control, and linear flight control. Numerous articles¹²⁻²¹ have discussed linear flight control methods for Quadrotors.

Several researchers have conducted research on trajectory tracking Quadrotor, such as on Abdelhay et al.²². A cascaded PID controller tracks the trajectory. The chosen model hovers at a linearized nonlinear model altitude. Additionally, the linear and nonlinear model responses are evaluated, and the PID controller for the nonlinear model is constructed and analyzed.

The research in²³ focuses on the simulation of a Quadrotor. A linear quadratic regulator-based optimum control is designed for the position and yaw control of the Quadrotor. The dynamics of a Quadrotor define its behavior in three-dimensional space. The created LQR-based controller was applied to the longitudinal, lateral, and vertical locations of the Quadrotor, as well as its

orientation in the yaw direction.

Elkhatem et al.²⁴⁾ mentioned that wind and other environmental disturbances can destabilize UAVs. LQR controllers eliminate state trajectory deviations with minimal control effort due to their great performance and resilience. An innovative method using the Quadrotor's state variables matrix and a preference factor automatically adjusts the weighting matrices in this paper. The designer penalizes or rewards a Quadrotor state to compensate for model perturbations. The proposed strategy for picking these matrices is tested for LQR and LQR with a PI controller. Both control solutions stabilized the Quadrotor with superior performance and robustness. The LQR-PI control system performed better in tracking than LQR alone.

Kuantama et al.²⁵⁾ utilized feedback linearization and LQR with a PID controller to stabilize Quadrotor attitude in the trajectory. The results demonstrate that variable weight adjustment on LQR and coefficients on PID generate better Quadrotor dynamic motion stability and performance. Overall, the modeling control method was able to reject disturbance, despite a little error in the output trajectory tracking.

Accelerated computer-aided design (CAD) improvements in recent years have led to new paths in engineering system modeling, design, and process control. The CAD development program provides new opportunities for engineering students to test their thinking. CAD software has created more accurate 3D models than ever before. These models can be utilized to describe both simple and complex engineering systems. This method makes modeling easier because the designer no longer must make the equations of motion for the system. This stage has been carried out by several researchers, as seen in²⁶⁻²⁹⁾.

Grau et al.²⁸⁾ describe a CAD-based method for identifying Quadrotor parameters. A global linearized control model is proposed. These systems use an LTI controller and a linear control model around each operational point to construct the controller. Raheemah et al.²⁷⁾ designed a Quadrotor CAD with SolidWorks software and converted the CAD to MATLAB. Furthermore, the PID control is used to regulate the Quadrotor's attitude so that it can follow the trajectory of the conical helix. From the literature review, we can conclude that by converting Quadrotor CAD to MATLAB, several stages of research can be done, such as identifying the Quadrotor and designing the desired control system.

Meanwhile, this paper's main contribution was to develop a simple Quadrotor model using SimMechanics simulation in MATLAB using a Simscape Multibody environment³⁰⁾. The CAD modelling results will provide almost accurate real-world system parameters, allowing us to continue with the control design. Then, using SimMechanics simulation in MATLAB, we solve the problem of Quadrotor trajectory tracking using full state feedback such as LQ-Servo for optimal control. The goal

is to evaluate the controllers' performance in assuring good trajectory tracking on a circular trajectory and trajectory tracking of the number "8".

This paper's remaining sections are organized as follows: Section II covers the dynamics of Quadrotors and describes the design of the LQ-Servo linear controller. Numerous simulation results are presented in Section III to demonstrate the value and effectiveness of the proposed methodologies. Section IV concludes the paper with observations and future work recommendations.

2. Method

2.1 Dynamics Modelling of a Quadrotor

Quadrotor modeling can be observed on a frame of a Cartesian diagram with a three-dimensional axis (x, y, z). The Cartesian diagram frame is divided into two, namely the earth frame, which is not moving, and the Quadrotor body frame that moves rotation and translation as presented in Fig. 1. Consider now a North-East-Down (NED) inertial frame fixed to earth and a body frame fixed to the Quadrotor. Linear position of Quadrotor (σ^E) is determined from the vector coordinates between the origin body frame and the origin of the earth frame with respect to the earth frame. Quadrotor angular position (θ^E) is determined based on how the body frame is oriented in relation to the earth frame. Linear position and angular position are found in Eq. 1 – Eq. 2. Furthermore, the Quadrotor rotates on the axis (x, y, z) so that it can be expressed as a rotation matrix in Eq. 3. Roll, pitch, and yaw indicate the Quadrotor's Euler angles with denoted ϕ, θ and ψ respectively.

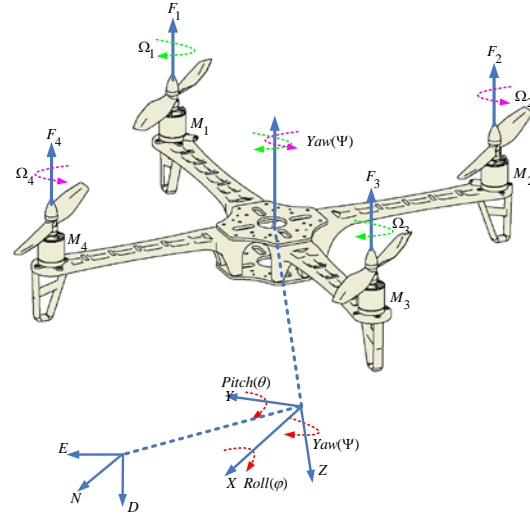


Fig. 1: Quadrotor reference frames

$$\sigma^E = [x \ y \ z]^T \quad (1)$$

$$\theta^E = [\phi \ \theta \ \psi]^T \quad (2)$$

$$R\theta = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (3)$$

Quadrotor speed consists of linear velocity $(\dot{x} \ \dot{y} \ \dot{z})$ and angular velocity $(\dot{\phi} \ \dot{\theta} \ \dot{\psi})$, so that when represented in the form of state space Quadrotor equations have 12 (twelve) states as in Eq. 4 – Eq. 5.

$$x_{1-6} = [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z}]^T \quad (4)$$

$$x_{7-12} = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T \quad (5)$$

Based on Fig. 1, the total lift (thrust) and moment of the Quadrotor can be obtained from the input u written in Eq. 6 – Eq. 7.

$$u = [u_1 \ u_2 \ u_3 \ u_4]^T \quad (6)$$

$$u_1 = K_c(\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4)$$

$$u_2 = K_c(-\Omega_2 + \Omega_4) \quad (7)$$

$$u_3 = K_c(\Omega_1 + \Omega_3)$$

$$u_4 = K_m(\Omega_1 - \Omega_2 + \Omega_3 - \Omega_4)$$

K_c and K_m are aerodynamic force constants and moment constants³¹⁾. Ω_i is the rotational speed of the rotor

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \left(\begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi \\ -s\theta & s\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \right) - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (10)$$

Next, if assumed $v = [\dot{x} \ \dot{y} \ \dot{z}]^T$ and $\omega = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ equal to zero, so we get the linear acceleration of Quadrotor (\ddot{v}) in Eq. 11 – Eq. 13.

$$\ddot{x} = -\frac{1}{m}(s\phi s\psi + c\phi s\theta c\psi)u_1 \quad (11)$$

$$\ddot{y} = -\frac{1}{m}(-s\phi s\psi + c\phi s\theta c\psi)u_1 \quad (12)$$

$$\ddot{z} = g - \frac{1}{m}(c\phi c\theta)u_1 \quad (13)$$

For Quadrotor rotations in Newton-Euler equations can be written as Eq. 14. M_B is the moment acting on the Quadrotor and can be obtained from the Eq. 15 where u_2, u_3, u_4 are the torque of the roll (ϕ), pitch (θ), and yaw (ψ). And J is the Quadrotor diagonal inertia matrix presented in Eq. 16.

$$M_B = J\omega + \dot{\omega} \times J\omega \quad (14)$$

$$M_B = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (15)$$

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (16)$$

By substituting Eq. 15 – Eq. 16 into Eq. 14, we get Eq. 17. The x-axis has a moment of inertia denoted by I_{xx} , the y-axis has a moment of inertia denoted by I_{yy} , and the z-

axis has a moment of inertia denoted by I_{zz} . Based on Eq. 17, it is obtained Eq. 18.

$$F_G - F_{thrust} = m\dot{v} + \omega \times mv \quad (8)$$

With F_G is gravity, F_{thrust} is a rotation matrix of vertical thrust $([0 \ 0 \ u_1]^T)$, m is the Quadrotor mass, $v = [\dot{x} \ \dot{y} \ \dot{z}]^T$ represents the linear velocity of the Quadrotor, $\dot{v} = [\ddot{x} \ \ddot{y} \ \ddot{z}]^T$ is Quadrotor's linear acceleration and $\omega = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ represents the angular velocity of the Quadrotor. So, Eq. 8 becomes Eq. 9.

$$\begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R\theta \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} = m \left(\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \right) \quad (9)$$

If Eq. 3 is substituted into Eq. 9, then it is obtained Eq. 10.

axis has a moment of inertia denoted by I_{zz} . Based on Eq. 17, it is obtained Eq. 18.

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} I_{xx}\ddot{\phi} + (I_{zz} - I_{yy})\dot{\theta}\dot{\psi} \\ I_{yy}\ddot{\theta} + (I_{xx} - I_{zz})\dot{\phi}\dot{\psi} \\ I_{zz}\ddot{\psi} + (I_{yy} - I_{xx})\dot{\phi}\dot{\theta} \end{bmatrix} \quad (18)$$

Thus, the angular acceleration of the Quadrotor ($\ddot{\omega}$) in Eq. 19 – Eq. 21.

$$\ddot{\phi} = \frac{u_2 + (I_{yy} - I_{zz})\dot{\theta}\dot{\psi}}{I_{xx}} \quad (19)$$

$$\ddot{\theta} = \frac{u_3 + (I_{zz} - I_{xx})\dot{\phi}\dot{\psi}}{I_{yy}} \quad (20)$$

$$\ddot{\psi} = \frac{u_4 + (I_{xx} - I_{yy})\dot{\phi}\dot{\theta}}{I_{zz}} \quad (21)$$

The state space representation of a Quadrotor can be expressed as in Eq. 22, after being rearranged according to the state order in Eq. 4 – Eq. 5.

$$\begin{aligned}
 \dot{x}_1 &= \dot{x} = x_2 \\
 \dot{x}_2 &= \ddot{x} = -\frac{1}{m}(s_\phi s_\psi + c_\phi s_\theta c_\psi)u_1 \\
 \dot{x}_3 &= \dot{y} = x_4 \\
 \dot{x}_4 &= \ddot{y} = -\frac{1}{m}(-s_\phi c_\psi + c_\phi s_\theta s_\psi)u_1 \\
 \dot{x}_5 &= \dot{z} = x_6 \\
 \dot{x}_6 &= \ddot{z} = g - \frac{1}{m}(c_\phi c_\theta)u_1 \\
 \dot{x}_7 &= \dot{\phi} = x_8 \\
 \dot{x}_8 &= \ddot{\phi} = \frac{u_2 + (I_{yy} - I_{zz})\dot{\theta}\dot{\psi}}{I_{xx}} \\
 \dot{x}_9 &= \dot{\theta} = x_{10} \\
 \dot{x}_{10} &= \ddot{\theta} = \frac{u_3 + (I_{zz} - I_{xx})\dot{\phi}\dot{\psi}}{I_{yy}} \\
 \dot{x}_{11} &= \dot{\psi} = x_{12} \\
 \dot{x}_{12} &= \ddot{\psi} = \frac{u_4 + (I_{xx} - I_{yy})\dot{\phi}\dot{\theta}}{I_{zz}}
 \end{aligned} \tag{22}$$

2.2 Linearization

To develop a linear controller, the nonlinear model expressed by Eq. 22 must be linearized about an operational point. We can simplify our analysis by applying the following approximations if we assume the Quadrotor is in a hovering state in Eq. 23. The yaw angle (ψ) is ignored for simplicity's sake. This means that the pitch angle (θ) will control the x position, while the roll angle (ϕ) will control the y position.

$$\begin{aligned}
 u_1 &\simeq mg \\
 \sin(\phi) &\simeq \phi \rightarrow \phi \simeq 0, \cos(\phi) \simeq 1 \\
 \sin(\theta) &\simeq \theta \rightarrow \theta \simeq 0, \cos(\theta) \simeq 1 \\
 \sin(\psi) &\simeq 0 \rightarrow \cos(\psi) = \sqrt{1 - \sin^2(\psi)} = 1
 \end{aligned} \tag{23}$$

Using these approximations, Eq. 22 can now be written under the linear form in Eq. 24. It is possible to write Eq. 24 in the following state space as in Eq. 25.

2.3 LQ-Servo Scheme

Using full state feedback control, we can ensure that all eigenvalues of the closed-loop system lie in the left half of the complex plane. This is accomplished by stabilizing a particular system. Consider a dynamic linear system described by the form $\dot{x} = Ax + Bu$ in the state space. By utilizing the full state feedback, which represents a

linear combination of the state variables, that is $u = -Kx$ in order for the closed-loop system, which is given Eq. 26.

$$\begin{aligned}
 \dot{x} &= (A - BK)x \\
 y &= Cx
 \end{aligned} \tag{26}$$

One type of full-state feedback optimal control is the linear quadratic regulator (LQR). The optimality criterion for LQR is specified by the cost function in Eq. 27. Matrix Q and Matrix R indicate the cost of each state x and control input u for a linear state space system. Calculating the control inputs that will yield the lowest possible value of cost function J is referenced in Eq. 28. The continuous Ricatti equation has a solution of P , as in Eq. 29.

$$J = \int_0^\infty [x(t)^T Q x(t) + u(t)^T R u(t)] dt \tag{27}$$

$$u^* = -K_{LQR}x = -R^{-1}B^T P x \tag{28}$$

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{29}$$

The LQR controller performs well in regulation tasks. Nonetheless, it cannot guarantee the tracking of dynamic references. This is due to the controller's lack of integral actions. Figure 2 shows a LQ-Servo formulation of the control problem. We found $\dot{x}_N = e = r - Cx$ from the LQ-Servo controller and augmented the state equations, yielding in Eq. 30. Finally, the reference for the control input u^* is calculated using the extended system state representation in Eq. 31.

$$\begin{bmatrix} \dot{x} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_N \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \tag{30}$$

$$y = [C \quad 0] \begin{bmatrix} x \\ x_N \end{bmatrix}$$

$$\begin{aligned}
 \dot{x}_1 &= \dot{x} = x_2 \\
 \dot{x}_2 &= \ddot{x} = -g\theta \\
 \dot{x}_3 &= \dot{y} = x_4 \\
 \dot{x}_4 &= \ddot{y} = +g\phi \\
 \dot{x}_5 &= \dot{z} = x_6 \\
 \dot{x}_6 &= \ddot{z} = g - \frac{u_1}{m} \\
 \dot{x}_7 &= \dot{\phi} = x_8 \\
 \dot{x}_8 &= \ddot{\phi} = \frac{u_2}{I_{xx}} \\
 \dot{x}_9 &= \dot{\theta} = x_{10} \\
 \dot{x}_{10} &= \ddot{\theta} = \frac{u_3}{I_{yy}} \\
 \dot{x}_{11} &= \dot{\psi} = x_{12}
 \end{aligned} \tag{24}$$

$$\dot{x}_{12} = \ddot{\psi} = \frac{u_4}{I_{zz}}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \quad (25)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

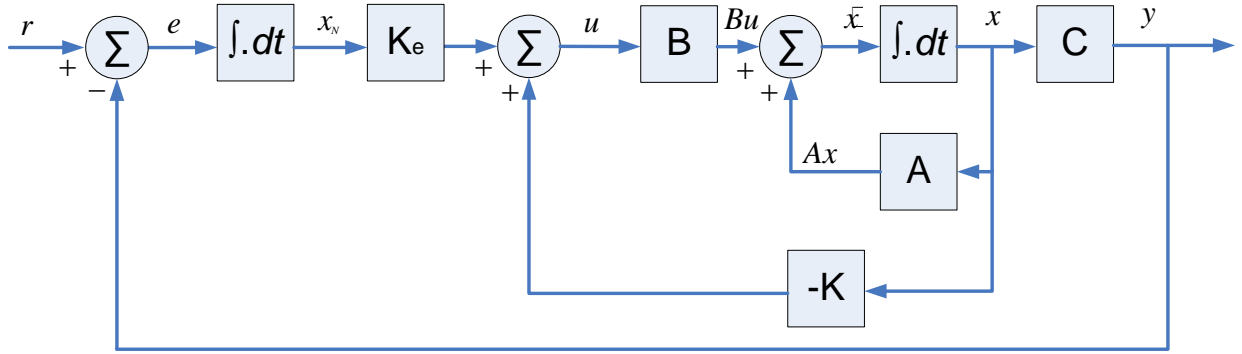


Fig. 2: Block diagram for LQ-Servo scheme

$$u^* = -Kx + K_e x_N = -[K \quad K_e] \begin{bmatrix} x \\ x_N \end{bmatrix} \quad (31)$$

2.4 Trajectory Tracking Control of Quadrotor

The purpose of the control system is for the Quadrotor to follow a specified trajectory established by a trajectory planner to develop an adequate flight plan. Quadrotor control can also be accomplished with multi loops as well

as a PID controller, as shown in Fig. 3. The loop is composed of two loops: one for rotational motion (inner loop) and one for translational motion (outer loop). The inner loop control rotates the Quadrotor. The rotating motion system must settle quickly to support the translational motion system as an outside loop.

The outer loop controls Quadrotor translational motion. Position control has tracking issues. The translational motion system must follow a reference signal and handle

interruptions. The control input u_1 is designed to control the Quadrotor's altitude. The control inputs u_2 , u_3 , and u_4 are used to control the Quadrotor's attitude.

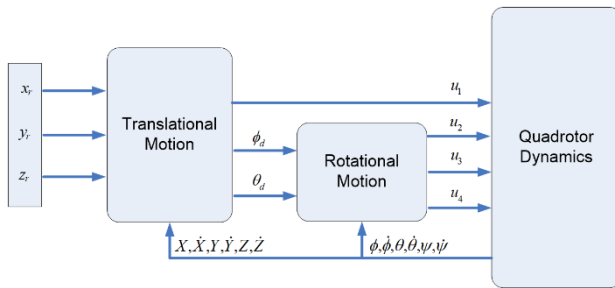


Fig. 3: Block diagram for multi loops control of Quadrotor

2.5 SimMechanics (Simscape Multibody™)

Simscape Multibody™ (formerly SimMechanics™) is a 3D mechanical system simulation environment that includes mechanical of robots. This software models a system by arranging its mechanical model rather than a mathematical model and demonstrates it visually, saving the designer time and effort in identifying a mathematical model of the system.

SimMechanics offers unique solver technology for simulating multibody mechanical systems. SimMechanics can import a CAD system model, including mass, inertia, joints, constraints, and 3D geometry. The following diagram illustrates how to load a CAD model into SimMechanics: First, download SimMechanics Link® and create an account. Second, Create an XML file based on the CAD model. Third, import the XML file into SimMechanics (Simscape Multibody™).



Fig. 4: Block diagram for converting CAD model into SimMechanics

In this paper, we design a Quadrotor CAD design using Autodesk Inventor Student version. Once we get the XML file of the Quadrotor CAD then we can get the Simulink block which represents the Quadrotor dynamics with apply PID control as shown in Fig. 5. From this Simulink we can apply the suitable control design. Figure 6 shown the Quadrotor already used in SimMechanics.

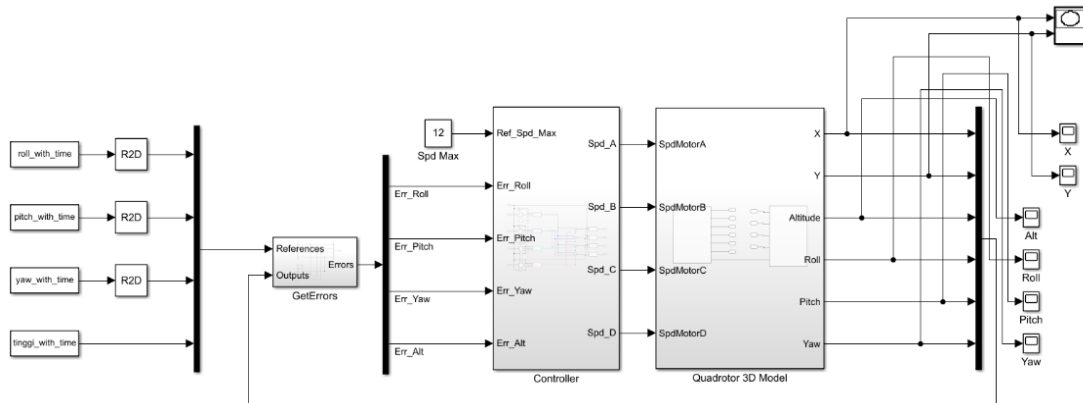


Fig. 5: Simulink of PID controller and Quadrotor dynamics models from CAD into SimMechanics

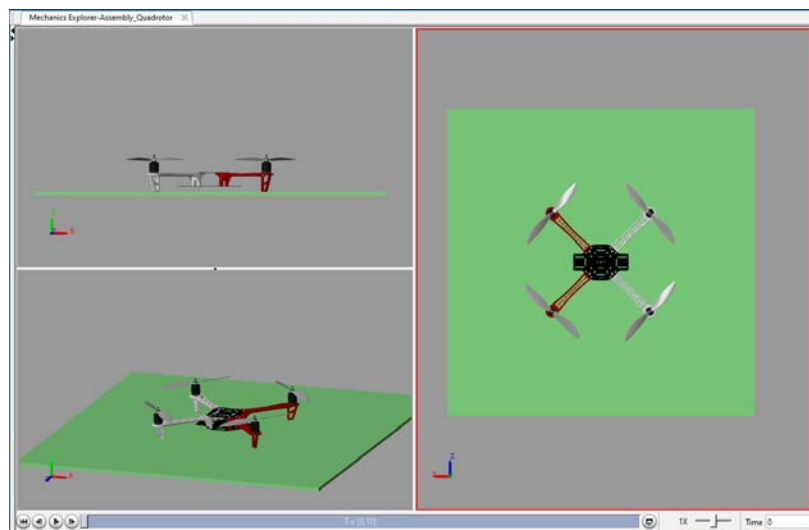


Fig. 6: SimMechanics Quadrotor results

3. Simulation Results and Discussion

To validate the effectiveness of the proposed control, a simulation is presented in MATLAB using Simulink and SimMechanics. Table 1 shows the Quadrotor parameters for simulations. Initial testing was carried out by observing the response of the Quadrotor in hovering conditions, with an assigned altitude, z_d of 2 meters. When employing an LQ-Servo control, the parameters of the Q Matrix and R Matrix significantly influence the feedback gain, K. During the initial stage of determining the Q Matrix and R Matrix parameters, the tuning process for Q and R weightings is performed using the values $Q = C^T C$ and $[eye(4) * 0.001]$. Subsequently, by adjusting the weighting matrix Q, variations in the feedback gain K can be obtained, for gain K_e (integral action) is tuned by

experimental fine-tuning process. This research presents five variants in the weighting matrix Q values, as indicated in Table 2.

Table 1. Simulation parameters

Parameter	Symbols	Value	Unit
Mass Quadrotor	m	1.3	Kg
Gravitational	g	9.81	m/s^2
The x-axis moment of inertia	I_{xx}	0.0081	$Kg.m^2$
The y-axis moment of inertia	I_{yy}	0.0081	$Kg.m^2$
The z-axis moment of inertia	I_{zz}	0.0142	$Kg.m^2$

Table 2. Response characteristics at altitude Quadrotor using LQ-Servo control

Characteristics	Q weighting matrix parameters				
	1 st $Q = C^T C$	2 nd $Q = [eye(12) * 0.1]$	3 rd $Q = [eye(12) * 10]$	4 th $Q = [eye(12) * 300]$	5 th $Q = [eye(12) * 700]$
RiseTime (s)	2.0138	1.0619	2.1381	2.1717	2.1696
SettlingTime (s)	3.8157	9.7043	3.8557	3.8699	3.8673
SettlingMin	1.4725	0.7311	1.7024	1.7855	1.7919
SettlingMax	1.6258	0.8206	1.8816	1.9782	1.9857
Overshoot (%)	0.1095	1.3647	0.0227	0	0
RMSE	0.6339	1.2574	0.5016	0.4722	0.4705

Figure 7 presents the altitude response of the Quadrotor with variations in the Q Matrix weighting. From this altitude response graph, it is evident that the 5th Q Matrix weighting parameter produces the smallest root mean square error (RMSE) compared to the others and does not result in overshoot.

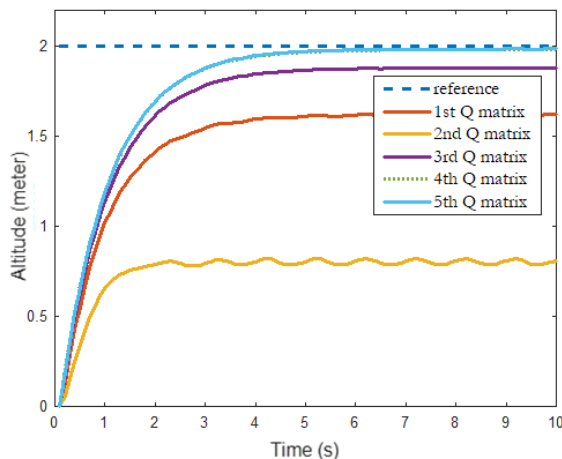


Fig. 7: Altitude response when heading to position $z_d = 2$ with variations in the Q Matrix weighting

As a comparison, a Quadrotor altitude control test was

conducted using a PID controller as in Fig. 8. The PID parameters used were: $K_p = 3$, $K_i = 0.02$, and $K_d = 5$, which were determined through experimental tuning. When applying PID control, the system has an overshoot of 11.67 %. Because the results of PID control produced with LQ-Servo generate a more appropriate response, the scenario of trajectory tracking with LQ-Servo control is investigated further in the next test.

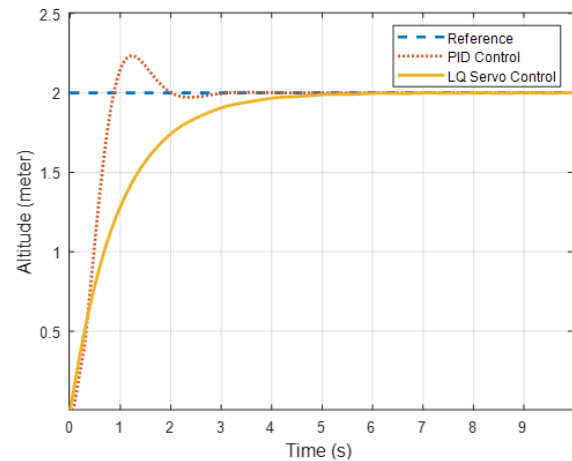


Fig. 8: Altitude response under PID control and LQ-Servo control

The second test involves providing trajectory tracking parameters. The Quadrotor had to follow the desired circular trajectory defined in Eq. 32.

$$\begin{aligned} x_d &= 1.4 \sin(\pi * t), y_d = 1.4 \cos(\pi * t) \\ z_d &= 2, \psi_d = 0 \end{aligned} \quad (32)$$

By utilizing Q Matrix weighting parameters of $[eye(12) \times 700]$ and R Matrix weighting of $[eye(4) \times 0.001]$, the resulting response in following the circular trajectory is presented in Fig. 9.

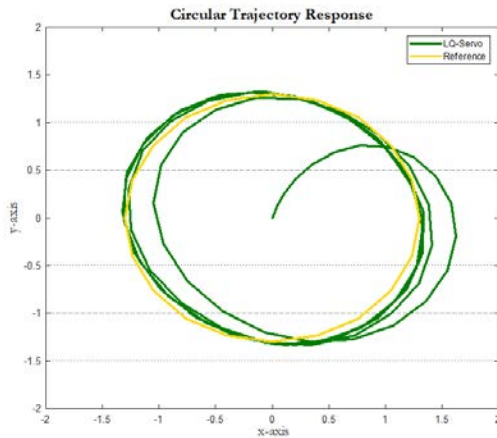


Fig. 9: Circular trajectory response using LQ-Servo control

Next, testing was conducted by introducing inertial disturbances into the Quadrotor dynamics system. This inertial disturbance takes the form of a payload that Quadrotor carries while following a given trajectory. It is assumed that the Quadrotor's payload is a solid cubic-shaped piece filled with liquid as shown in Fig. 10 with the specifications in Table 3. And the payload does not affect the system's center of mass.

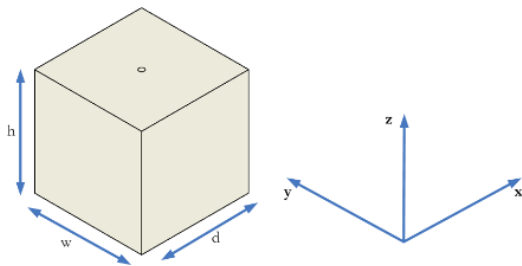


Fig. 10: Models of payloads

Table 3. Payload specifications

Parameter	Value	Units
Mass payload (m_d)	0.2	Kg
Length (d)	0.08	m
Width (w)	0.08	m
Height (h)	0.08	m

To facilitate the analysis, the inertial matrix is separated into two types: the inertial matrix at nominal conditions (J_0) according to Eq. 33 and the inertial matrix that results from the presence of payload (ΔJ). The value of ΔJ is affected by the products' mass, shape, and position of payload. The addition of the inertial matrix that appears due to the presence of payload referred to as the perturbed inertia matrix, allowing the inertial matrix to be written as Eq. 34.

$$J_0 = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (33)$$

$$J_\delta = J_0 + \Delta J \quad (34)$$

Because the payload is in the form of blocks, the inertial disturbance matrix is a symmetric matrix with zero off-diagonal elements as shown in Eq. 35 with the matrix elements described in Eq. 36.

$$\Delta J = \begin{bmatrix} \Delta I_{xx} & 0 & 0 \\ 0 & \Delta I_{yy} & 0 \\ 0 & 0 & \Delta I_{zz} \end{bmatrix} \quad (35)$$

$$\Delta I_{xx} = \frac{m_d}{12} (d^2 + h^2)$$

$$\Delta I_{yy} = \frac{m_d}{12} (d^2 + w^2) \quad (36)$$

$$\Delta I_{zz} = \frac{m_d}{12} (h^2 + w^2)$$

Based on Eq. 24 and the payload mass and dimensions in Table 3, the Quadrotor inertial dynamics are as described in Table 4.

Table 4. Quadrotor inertia after being given a payload

Parameter	Initial Inertia	Payload Inertia	Total (J_δ)	Unit
I_{xx}	$1.16E-04$	$2.13E-04$	$3.29E-04$	$Kg.m^2$
I_{yy}	$4.08E-05$	$2.13E-04$	$2.54E-04$	$Kg.m^2$
I_{zz}	$1.056E-04$	$2.13E-04$	$3.18E-04$	$Kg.m^2$

Figure 10 shows the results of the response after being given an inertial disturbance for circle trajectory tracking. Table 5 presents the RMSE value when the UAV Bicopter follows a circular trajectory.

Table 5. RMSE value when following a circular trajectory tracking

Position	LQ-Servo with	
	LQ-Servo	inertial disturbance
Position x	0.6447	0.6630
Position y	0.5382	0.5426

The next test is by providing trajectory parameters in the form of the number "8" using Eq. 37. Figure 11 shows the results of the response after being given an inertial disturbance for trajectory tracking of the number "8" and Table 6 presents the RMSE value.

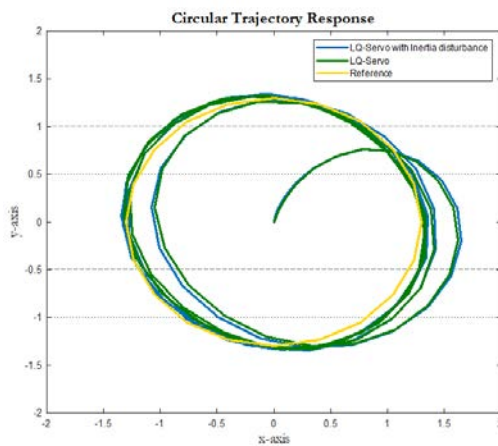


Fig. 10: Circular trajectory tracking response using LQ-Servo control with inertial disturbance

$$\begin{aligned} x_d &= \sin(0.25\pi * t), y_d = \sin(0.5\pi * t) \\ z_d &= 2, \psi_d = 0 \end{aligned} \quad (37)$$

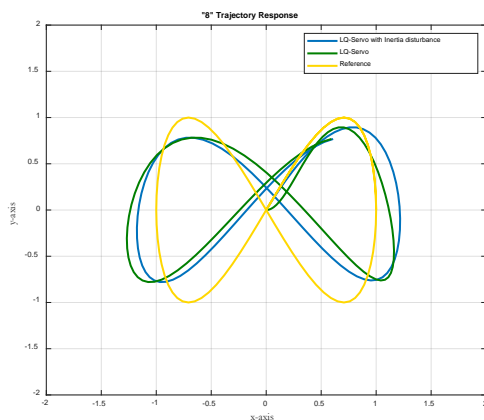


Fig. 11: "8" trajectory tracking response using LQ-Servo control with inertial disturbance

Table 6. RMSE value when following a "8" trajectory tracking LQ-Servo with

Position	LQ-Servo	inertial disturbance
Position x	0.7118	0.7395
Position y	0.8189	0.9037

From Table 5, it can be observed that with the presence of an inertial disturbance, the LQ-Servo controller experiences a significant increase in RMSE at position x and a slight increase in RMSE at position y . Similarly, in the case of following the trajectory of the number '8'. From Table 6, it is evident that the RMSE value for position x and position y also increases in the presence of inertial disturbance. Based on the results of this test, it is evident that the LQ-Servo control can still maintain Quadrotor trajectory tracking with only a slight increase in RMSE when subjected to inertial disturbances.

4. Conclusion

This study begins by applying Newton's and Euler's laws to Quadrotor dynamics. Subsequently, the model is linearized, resulting in the derivation of a linear controller known as the LQ-Servo. Simulation results of Quadrotor trajectory tracking using the LQ-Servo controller demonstrate the suppression of inertial interference, in the form of payload, during trajectory tracking. The success of LQ-Servo control was tested in two scenarios: the first involving trajectory tracking in a circular position, and the second involving trajectory tracking in the form of the number '8'.

In conclusion, the proposed control strategies were implemented in SimMechanics to allow observation of the Quadrotor's behavior in virtual reality. Potential future work could involve enhancing a Quadrotor's utility by employing nonlinear control methodologies like adaptive feedback linearization and sliding mode control.

Acknowledgements

This work was supported by the Indonesian Postgraduate Domestic Education Scholarship (BPPDN) with contract number 2974/UN1.P.IV/KPT/DSDM/2019.

References

- 1) N.H. Mai, and L.Q. Huy, "Design a Neural Controller to Control Rescue Quadcopter in Hang Status," in: *Int. Conf. Comput. Collect. Intell.*, Springer, pp. 301–313, (2020).
- 2) K. Choutri, M. Lagha, and L. Dala, "A fully autonomous search and rescue system using Quadrotor uav," *Int. J. Comput. Digit. Syst.*, **10** (01) 403–414 (2021).
- 3) G.A. Cardona, J. Ramirez-Rugeles, E. Mojica-Nava, and J.M. Calderon, "Visual victim detection and

- Quadrotor-swarm coordination control in search and rescue environment,” *Int. J. Electr. Comput. Eng.*, **11** (3) 2079 (2021).
- 4) S. Zhang, H. Wang, S. He, C. Zhang, and J. Liu, “An autonomous air-ground cooperative field surveillance system with Quadrotor UAV and unmanned ATV robots,” in: 2018 IEEE 8th Annu. Int. Conf. CYBER Technol. Autom. Control. Intell. Syst., IEEE, pp. 1527–1532, (2018).
 - 5) P.S. Gohari, H. Mohammadi, and S. Taghvaei, “Using chaotic maps for 3d boundary surveillance by Quadrotor robot,” *Appl. Soft Comput.*, **76** 68–77 (2019).
 - 6) C. Montañez-Molina, J. Pliego-Jiménez, and C. Cruz-Hernández, “Chaotic velocity profile for surveillance tasks using a Quadrotor,” in: 2021 IEEE Conf. Control Technol. Appl., IEEE, pp. 323–328, (2021).
 - 7) L. Fahmani, J. Garfaf, K. Boukhdar, S. Benhadou, and H. Medromi, “Modelling of very high voltage transmission lines inspection’s Quadrotor,” *SN Appl. Sci.*, **2** (8) 1–11 (2020).
 - 8) W. Gu, D. Hu, L. Cheng, Y. Cao, A. Rizzo, and K.P. Valavanis, “Autonomous wind turbine inspection using a Quadrotor,” in: 2020 Int. Conf. Unmanned Aircr. Syst., IEEE, pp. 709–715, (2020).
 - 9) K. Takaya, H. Ohta, V. Kroumov, K. Shibayama, and M. Nakamura, “Development of UAV system for autonomous power line inspection,” in: 2019 23rd Int. Conf. Syst. Theory, Control Comput., IEEE, pp. 762–767, (2019).
 - 10) G.M. Qian, D. Pebrianti, L. Bayuaji, N.R.H. Abdullah, M. Mustafa, M. Syafrullah, and I. Riyanto, “Waypoint navigation of quad-rotor MAV using Fuzzy-PID control,” in: *Intell. Manuf. Mechatronics*, Springer, pp. 271–284, (2018).
 - 11) G.M. Qian, D. Pebrianti, Y.W. Chun, Y.H. Hao, and L. Bayuaji, “Waypoint navigation of quad-rotor MAV,” in: 2017 7th IEEE Int. Conf. Syst. Eng. Technol., IEEE, pp. 38–42, (2017).
 - 12) M.F. Shehzad, A. Bilal, and H. Ahmad, “Position & attitude control of an aerial robot (Quadrotor) with intelligent pid and state feedback lqr controller: A comparative approach,” in: 2019 16th Int. Bhurban Conf. Appl. Sci. Technol., IEEE, pp. 340–346, (2019).
 - 13) E. Okyere, A. Bousbaine, G.T. Poyi, A.K. Joseph, and J.M. Andrade, “LQR controller design for quad-rotor helicopters,” *J. Eng.*, **2019** (17) 4003–4007 (2019).
 - 14) L. Martins, C. Cardeira, and P. Oliveira, “Linear quadratic regulator for trajectory tracking of a Quadrotor,” *IFAC-PapersOnLine*, **52** (12) 176–181 (2019).
 - 15) M. Islam, and M. Okasha, “A Comparative Study of PD, LQR and MPC on Quadrotor Using Quaternion Approach,” in: 2019 7th Int. Conf. Mechatronics Eng., IEEE, pp. 1–6, (2019).
 - 16) S. Ibrahim, B. Alkali, A. Oyewole, S.B. Alhaji, and A.A. Abdullahi, “Structural integrity study for a quadcopter frame to be deployed for pest control,” *Evergreen* vol.8, no.3, pp.667–672, (2021). <https://doi.org/10.5109/4491843>
 - 17) T. Huang, D. Huang, Z. Wang, N. Qin, and A. Shah, “Robust control for a Quadrotor UAV based on linear quadratic regulator,” in: 2020 39th Chinese Control Conf., IEEE, pp. 6893–6898, (2020).
 - 18) B. Han, Y. Zhou, K.K. Deveerasetty, and C. Hu, “A review of control algorithms for Quadrotor,” in: 2018 IEEE Int. Conf. Inf. Autom., IEEE, pp. 951–956, (2018).
 - 19) G. Farid, M. Hongwei, S.M. Ali, and Q. Liwei, “A review on linear and nonlinear control techniques for position and attitude control of a Quadrotor,” *Control Intell. Syst.*, **45** (1) 43–57 (2017).
 - 20) T.N. Dief, and S. Yoshida, “System identification for quad-rotor parameters using neural network,” *Evergreen* vol.3, no.1, pp.6–11, (2016). <https://doi.org/10.5109/1657380>
 - 21) T.N. Dief, and S. Yoshida, “System identification and adaptive control of mass-varying quad-rotor,” *Evergreen* vol.4, no.1, pp58–66, (2017). <https://doi.org/10.5109/1808454>
 - 22) S. Abdelhay, and A. Zakriti, “Modeling of a quadcopter trajectory tracking system using pid controller,” *Procedia Manuf.*, **32** 564–571 (2019).
 - 23) F. Ahmad, P. Kumar, A. Bhandari, and P.P. Patil, “Simulation of the quadcopter dynamics with lqr based control,” *Mater. Today Proc.*, **24** 326–332 (2020).
 - 24) A.S. Elkhateem, and S.N. Engin, “Robust lqr and lqr-pi control strategies based on adaptive weighting matrix selection for a uav position and attitude tracking control,” *Alexandria Eng. J.*, **61** (8) 6275–6292 (2022).
 - 25) E. Kuantama, I. Tarca, and R. Tarca, “Feedback linearization LQR control for quadcopter position tracking,” in: 2018 5th Int. Conf. Control. Decis. Inf. Technol., IEEE, pp. 204–209, (2018).
 - 26) V. Shrivastava, V. Diwakar, M. Sehgal, M. Verma, and E. Neha, “Modelling and analysis of hexapod walking robot,” *Evergreen* vol.9, no.2, pp378–388, (2022). <https://doi.org/10.5109/4794162>
 - 27) F.A. Raheemah, and M.T. Hussein, “3D cadmodel for a Quadrotor system modeling and control,” *Diagnostyka*, **23** (2022).
 - 28) A. Grau, Y. Bolea, and A. Sanfeliu, “CAD-based Approach for Identification of UAVs,” in: MATEC Web Conf., EDP Sciences, pp. 1004, (2019)
 - 29) J.J. Brito, P. Toledo, and S. Alayón, “Virtual laboratory for automation combining inventor 3D models and Simulink control models: Virtual laboratory for automation,” in: 2018 IEEE Glob. Eng. Educ. Conf., IEEE, pp. 555–562, (2018).
 - 30) P. Noskievič, and D. Walica, “Design and Realisation of the Simulation Model of the Stewart Platform using the MATLAB-Simulink and the Simscape

- Multibody Library,” in: 21th Int. Carpathian Control Conf., IEEE, pp. 1–5, (2020).
- 31) Fahmizal, A. Surriani, M. Budiyanto, and M. Arrofiq, “Altitude control of Quadrotor using fuzzy self tuning PID controller,” in: 5th Int. Conf. Instrumentation, Control. Autom. (ICA), IEEE, pp. 67-72, (2017).
- 32) K.J. Åström, and R.M. Murray, “Feedback systems: an introduction for scientists and engineers,” Princeton university press, (2021).