

Genetically Tuned Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor

Ali Tahir KARAŞAHİN

Karabük University, Faculty of Engineering, Department of Mechatronics Engineering, Karabük, Türkiye, tahirkarasahin@karabuk.edu.tr 

Abstract

In this paper, a linear quadratic regulator (LQR) controller operating according to the genetically tuned inner-outer loop structure is proposed for trajectory tracking of a quadrotor. Setting the parameters of a linear controller operating according to the inner-outer loop structure is a matter that requires profound expertise. Optimization algorithms are used to cope with the solution of this problem. First, the dynamic equations of motion of the quadrotor are obtained and modelled in state-space form. The LQR controller, which will operate according to the inner-outer loop structure in the MATLAB/Simulink environment, has been developed separately for 6 degrees of freedom (DOF) of the quadrotor. Since adjusting these parameters will take a long time, a genetic algorithm has been used at this point. The LQR controller with optimized coefficients and a proposed LQR controller-based study in the literature are evaluated according to their success in following the reference trajectory and their responses to specific control inputs. According to the results obtained, it was observed that the genetically adjusted LQR controller produced more successful outcomes.

Keywords: Quadrotor; LQR Controller; Inner-Outer Loop; Trajectory-Tracking; Genetic Algorithm

1. INTRODUCTION

Unmanned aerial vehicles, especially multi-propeller vehicles, are widely used in many industries, such as search and rescue, reconnaissance and surveillance, mapping, and inspection of power lines [1–5]. Multicopters used in these applications are generally produced with four propellers. The desired position or angle is achieved by applying different rotational speeds to the brushless dc motors placed on this four-propeller aircraft. However, since the dynamics of the quadcopter are nonlinear and inherently unstable, it is a problem to control.

Model-based or model-free linear or nonlinear controllers are being developed for quadcopters to follow the references given to them. Nonlinear controllers such as feedback linearization, adaptive sliding mode control [6] and backstepping [7] have been proposed to match the non-linear characteristics of quadcopters. However, to develop a nonlinear controller, quadcopters must be modelled to be suitable for all operating conditions. Since this is not possible in practice, adaptive control techniques [8]–[10] are used to solve the stated problem. Adaptive controllers incorporate approaches that can adapt to changes in the working environment. But to do this, they depend on the system model to be highly accurate. It has been shown that the attitude controller, which is developed with the model

reference adaptive control technique, successfully performs this operation on the hardware [11]. Reinforcement learning techniques are also used, which iteratively perform learning processes independently of the system model. In fact, there are studies to carry out the control process with those that remain intact against engine failures that occur in the aircraft with these techniques [12]. However, reinforcement learning processes cannot guarantee that the system will always produce stable responses. Various fuzzy logic controllers are also frequently used, regardless of the system model. Interval type-2 fuzzy logic controllers are used to develop a controller that is resistant to changes in system parameters or disturbances in the working environment [13]–[14]. Although the controllers developed with model-based or model-free techniques have some unique advantages, it is expected that they will be able to show the expected performance on the hardware. The control algorithm that will work on the hardware should be considered in some criteria, such as not bringing a high processing load, short response times, and short development processes.

For the reasons stated, proportional-integral-derivative (PID) and LQR techniques can be said to be advantageous at this point. In addition, in flight control software (PX4, etc.) commonly used for drones, the PID controller comes by default. The adjustment of the coefficients in the PID technique used in the quadcopter's position controller was

carried out by different optimization algorithm [15]–[20]. However, many of the control techniques developed for quadcopters are recommended for a part of the control system (usually as a replacement for the position controller). This practice is even more common, especially when performed experimentally on hardware. It is thought that there are areas with improvement potential due to this approach for systems with many problems, such as quadcopters.

Open-source autopilot software such as PX4 [21] or ArduPilot [22] developed for quadcopters has been developed with cascade PID controllers according to the inner-outer loop structure. The coefficients in the PID control technique are set to a certain degree by default. These coefficients should be adjusted by an expert who knows the behavior of the quadcopter according to the application in which it will operate. Adjusting the specified coefficients is both a matter requiring expertise and a serious time-consuming process.

In this paper, a genetically tuned LQR controller is proposed to increase the quadcopter's ability to follow the reference trajectory. The controller proposed for the quadcopter is designed according to the inner-outer loop structure, and the coefficients used in the LQR controller are optimized by a genetic algorithm (GA).

As a contribution to this paper, the integrative action and LQR-based controller coefficients working according to the inner-outer loop structure were optimized, and gains were obtained in the performance criteria determined according to the current study in the literature. In addition, the dynamic model of the quadcopter with the X configuration is derived simply and straightforwardly and linearized according to certain approaches, making it suitable for model-based controller development.

The article is organized as follows: In Section 2, the motion equations acting on the quadcopter are obtained according to the Newton-Euler formulas, the obtained motion equations are linearized according to the determined approaches, and the motion equations are presented in the state-space form by with the quadcopter inner-outer loop structure. In Section 3, the design of the LQR-based controller with integrative action according to the inner-outer loop structure has been carried out. In Section 4, the quadcopter is modelled in a MATLAB/Simulink environment according to the parameters of Parrot AR. Drone 2.0. The developed controller responses were tested according to the specified control inputs and following the reference trajectory. Additionally, the results of the simulation tests performed are shared. In Section 5, the test results are evaluated, and future work topics are mentioned.

2. MATERIALS AND METHODS

In this section, the equations of motion of the quadcopter will be derived according to Newton-Euler formulas. The equations of motion required for quadcopter body-fixed $\{B\}$ and inertial $\{I\}$ frames are shown in Figure 1. According to the inertial frame specified here, the position vector of the center of gravity of the quadcopter is expressed as $p =$

$[x \ y \ z]^T$. The Euler angle vector $n = [\varphi \ \theta \ \psi]^T$ in the body-fixed frame is denoted as the roll, pitch, and yaw angle, respectively. The angular velocity component is expressed as $\omega = [p \ q \ r]^T$. The quadcopter's equations of motion are expressed as follows [23]:

$$m\ddot{p} = -mg\vec{a}_3 + {}^I R_B F \quad (1)$$

$$I\dot{\omega} = -\omega \cdot I\omega + \tau \quad (2)$$

It is the rotation matrix used in the transformation from the ${}^I R_B$ body-fixed frame to the inertial frame in Equation 1.

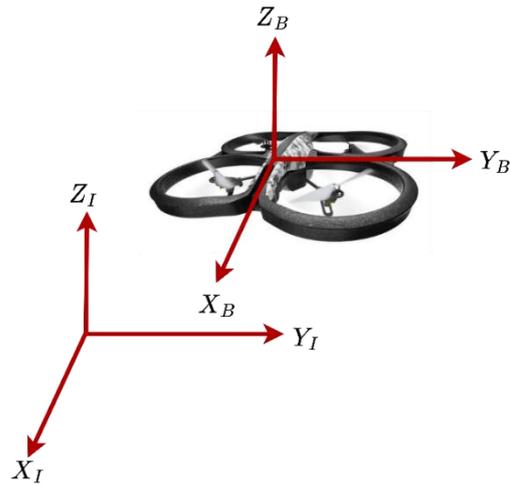


Figure 1. Reference frames are defined for the quadcopter model [24].

The unit vector in the body-fixed frame is $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ and in the inertial frame it is $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$. m shows the total mass of the quadcopter, g shows the gravity acceleration value, F shows the force produced by the motors, I show the 3×3 inertia matrix defined in the body-fixed frame, and τ shows the moments generated from the quadcopter. The rotation matrix used in the transformation from the body-fixed frame used in Equation 1 to the inertia frame. This rotation matrix performs the transformation between the specified axis sets [25]:

$${}^I R_B = \begin{bmatrix} c\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ c\theta s\psi & s\varphi s\theta s\psi - c\varphi c\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (3)$$

In Equation 3, s and c denote sine and cosine, respectively. It should be noted that the ${}^B R_I$ matrix [25] used in the controller design is different from the one specified in Equation 3. Angular rates of change $\dot{n} = [\dot{\varphi} \ \dot{\theta} \ \dot{\psi}]^T$ are obtained from the rotation rates occurring in the quadcopter body as follows [26]:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi s\theta & c\varphi s\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4)$$

In Equation 4: s , c , se and t denote sine, cosine, secant, and tangent, respectively. After a reference is given to the quadcopter, the relationship between the yaw moment and

thrust force obtained from the motors when it is in steady state can be defined as follows [27]:

$$T_i = c_{T_i} \Omega_i^2 \quad (5)$$

$$\tau_i = c_{\tau_i} \Omega_i^2 \quad (6)$$

The expression T_i in Equation 5 represents the thrust force produced by any motor, and c_{T_i} is used as the coefficient that realises the conversion between angular velocity and thrust force. The Ω value used in the same equation represents the angular velocity of the motors. Similarly, the expression τ_i in Equation 6 represents the yaw moment produced from any motor, c_{τ_i} is the coefficient that converts between angular velocity and yaw moment, and these coefficients can be calculated experimentally. In addition, these coefficients vary according to the propeller type, number of blades, profile, and air density. Equations 5 and 6 in the case of analyzing the thrust force and yaw moment on the z-axis [27]:

$$\tau_{\psi_i} = \frac{c_{\tau_i}}{c_{T_i}} T_i = c_i T_i \quad (7)$$

can be written as. The total thrust force (T) obtained from the motors on the quadcopter and the moments (τ_i) equations of the x, y and z axes are shown in Equation 8 [24]:

$$\begin{bmatrix} T \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ l & -l & -l & l \\ -l & -l & l & l \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (8)$$

In Equation 8, the rows represent the total thrust force obtained from the four motors and the other rows represent the moments on the x, y, and z axes, respectively. The l value indicates the distance from the centre of gravity of any motor in the quadcopter.

2.1. Linearization of the Quadcopter Model

When the sets of equations specified in Equations (1)-(8) are examined, it is seen that they contain nonlinear expressions. Since linear control techniques will be used in this study, these equation sets should be linearized. The equilibrium point for linearization was determined as the quadcopter's hover position ($p = [x \ y \ z]^T, n = [0 \ 0 \ 0]^T$). This is preferred for simplicity. With the small angles approach, the cosine values are assumed to be 1, and the sine and tangent Euler angles are accepted as themselves.

Within the scope of this study, an LQR-based controller with integrative action was developed according to the inner-outer loop structure. Therefore, six different controllers need to be designed. These controllers are the position in the x, y, and z axes designed to follow the angle references on φ, θ and ψ axes. Therefore, for each controller, its systems are expressed with a state-space approach [24]. T in Equation 9 represents the total thrust obtained from all motors. Equations of motion to consider when developing the height controller in the z axis [26]:

$$\ddot{z} \cong \frac{1}{m} (T - mg) \quad (9)$$

The state variables and inputs determined for this system are as follows:

$$x_z = [z \ \dot{z}]^T, u_z = T - mg \quad (10)$$

The state-space representation is as follows:

$$\dot{x}_z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_z + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u_z \quad (11)$$

$$y_z = [1 \ 0] x_z \quad (12)$$

Changes in the x and y axes of the quadcopter also have effects on the roll and pitch angles in the inertia frame. We can express these changes as follows:

$$\ddot{\theta} \cong \frac{\tau_\theta}{I_y} \quad (13)$$

According to the specified motion equation, the state variable and inputs are as follows:

$$x_\theta = [\theta \ \dot{\theta}]^T, u_\theta = \tau_\theta \quad (14)$$

The state space representation consists of the following:

$$\dot{x}_\theta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\theta + \begin{bmatrix} 0 \\ 1/I_y \end{bmatrix} u_\theta \quad (15)$$

$$y_\theta = [1 \ 0] x_\theta \quad (16)$$

The acceleration equation of motion occurring in the body-fixed frame is:

$${}^B a = \frac{F}{m} - {}^B R_1 g \vec{a}_3 - \omega \cdot {}^B v \quad (17)$$

In Equation 17, ${}^B v = [u \ v \ w]^T$ denotes the velocity value in the body-fixed frame, $\omega \cdot {}^B v$ denotes centripetal acceleration. If the x and y directions in the body-fixed frame are linearized, and the yaw angle ψ is assumed to be zero:

$${}^B a_x \cong \theta g \quad (18)$$

$${}^B a_y \cong -\varphi g \quad (19)$$

As per the definition ${}^B v$ stated in Equation 17:

$$\dot{u} \cong \theta g \quad (20)$$

$$\dot{v} \cong -\varphi g \quad (21)$$

can be edited. According to the obtained equations, the state variables to be designed for the x position can be arranged as follows:

$$x_x = [{}^B x_l \ u]^T, u_x = \theta \quad (22)$$

State-space representation to be used for reference position control on the X-axis:

$$\dot{x}_x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_x + \begin{bmatrix} 0 \\ g \end{bmatrix} u_x \quad (23)$$

$$y_x = [1 \ 0] x_x \quad (24)$$

After linearizing the equations of motion occurring in the y-axis under certain assumptions, they can be expressed as follows:

$$\ddot{\phi} \cong \frac{\tau_\phi}{I_x} \quad (25)$$

The state variable and inputs, according to the specified motion equation are as follows:

$$x_\phi = [\phi \ \dot{\phi}]^T, u_\phi = \tau_\phi \quad (26)$$

Its representation in state-space form is:

$$\dot{x}_\phi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\phi + \begin{bmatrix} 0 \\ 1/I_x \end{bmatrix} u_\phi \quad (27)$$

$$y_\phi = [1 \ 0] x_\phi \quad (28)$$

According to Equations (19) and (21), the state variables and input for the controller that will perform the position control on the y axis can be used as follows:

$$x_y = [{}^B y_l \ v]^T, u_y = \phi \quad (29)$$

State-space representation to be used for reference position control in the y axis is:

$$\dot{x}_y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_y + \begin{bmatrix} 0 \\ -g \end{bmatrix} u_y \quad (30)$$

$$y_y = [1 \ 0] x_y \quad (31)$$

The angular acceleration occurring in the yaw axis can be linearized as follows:

$$\ddot{\psi} \cong \frac{\tau_\psi}{I_z} \quad (32)$$

State variable and input for the controller on this axis:

$$x_\psi = [\psi \ \dot{\psi}]^T, u_\psi = \tau_\psi \quad (33)$$

It can be expressed as. State-space representation to be used in the controller on this axis:

$$\dot{x}_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\psi + \begin{bmatrix} 0 \\ 1/I_z \end{bmatrix} u_\psi \quad (34)$$

$$y_\psi = [1 \ 0] x_\psi \quad (35)$$

It can be used as. Thus, a quadcopter dynamic model is derived with six different state space representations. With the dynamic model obtained, it has become usable in understanding flight dynamics with some fixed inputs in the MATLAB/Simulink environment.

3. LQR CONTROLLER

The linear quadratic regulator is an optimal linear controller technique that performs optimization over system dynamics after all states are taken as feedback. If a design is to be made with the LQR controller, the system to be controlled should be modelled as follows:

$$\dot{x} = Ax + Bu \quad (36)$$

$$y = Cx + Du \quad (37)$$

The K matrix determined because of the optimization process is as follows:

$$u(t) = -Kx(t) \quad (38)$$

is implemented. Optimized function in LQR controller:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (39)$$

is defined as. Q refers to the weight matrix, and R refers to the control matrix specified in Equation 39. Calculation of the optimal coefficient is done as follows:

$$K = R^{-1} B^T P \quad (40)$$

The steady-state of the P matrix in Equation 40 is determined according to the Riccati equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (41)$$

An integrator is added to the LQR controller structure to eliminate perturbations and steady-state error. Thus, a more robust and steady-state error is eliminated against uncertainties. The *ref* value in Equation 42 represents the reference given to the controller. Integrator entry in LQR structure with integrative action:

$$\dot{\xi} = ref - y = ref - Cx \quad (42)$$

is implemented as. The integrator output is expressed as ξ . By adding the integrator to the LQR controller, the controller response is written as:

$$u(t) = -Kx(t) + k_1 \xi \quad (43)$$

State-space representation after adding the integrator process to the LQR controller [24]:

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} ref \quad (44)$$

is becoming. According to the representation in the newly formed state-space form [24]:

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad (45)$$

It is arranged in the form of matrix A and B. The optimal gain matrix is as follows [24]:

$$\bar{K} = [K \quad -k_1] \tag{46}$$

is expressed. In case the quadcopter is required to follow the position and angle reference given, the LQR controller is designed according to the inner-outer loop structure with the designed integrative action. The controller designed according to the inner-outer loop structure is shown in Figure 2.

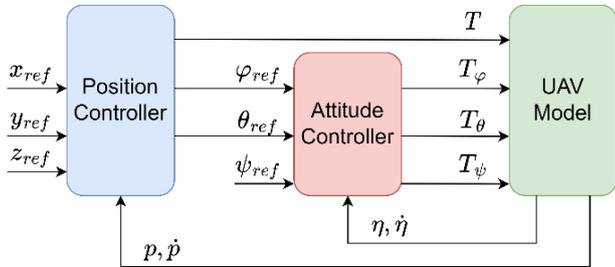


Figure 2. Inner-outer loop structure used for GA-LQR controller [24].

It is necessary to determine the Q and R matrix in the optimized function in the LQR controller. Determining this weight and control matrix by trial-and-error method takes considerable time. There are 18 coefficients in total that need to be set in the LQR controller for the x , y and z positions of the quadcopter and the roll, pitch, and yaw reference. These coefficients can be determined without losing time with optimization algorithms such as GA in the simulation environment.

3.1. Genetic Algorithm

The coefficients in the LQR controller, which is designed according to the inner-outer loop structure with integrative action, are adjusted with GA. GA is a search algorithm that tries to find the best result based on optimization. It was proposed by John Holland in 1975. It consists of natural selection, mutation, and crossover operators. GA works to find the values that will bring the given objective function to the best result. The flow diagram of the GA is shown in Figure 3.

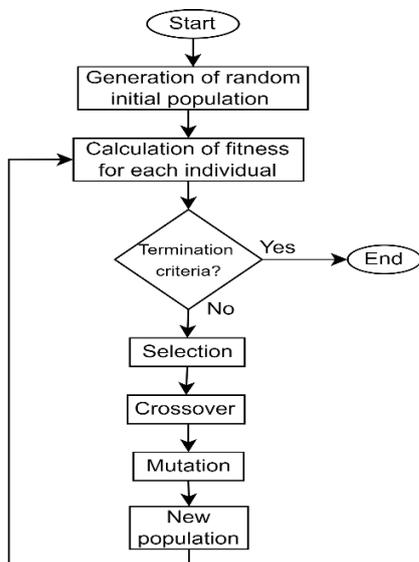


Figure 3. Genetic algorithm flow diagram.

A total of 18 coefficients that determine their success in tracking position and angle references have been optimized with GA. The gain matrix in Equation 46 is valid for position and angle controllers; the matrix K in this equation represents the state variables, and the matrix k_1 represents the integrative action value. For each controller, three coefficients must be used with two state variables and an integrative action value. Thus, 18 coefficients in total need to be determined for position and controllers. This optimization process was carried out using the dynamic model of the quadcopter in the MATLAB/Simulink environment. Root-mean-square error (RMSE) value of position errors is used for the fitness function used in GA. Function used for this operation:

$$f_{min} = rms(x) + rms(y) + rms(z) \tag{47}$$

In the GA working with the determined fitness function, the number of generations: 40, the number of populations: 20, the crossover ratio: 0.85, and the mutation rate: 0.20. GA was run on a computer equipped with AMD Ryzen 7 3700U processor and 16 GB RAM. With the specified GA parameters and hardware specifications, the run time was 907.42 seconds.

4. SIMULATION RESULTS AND DISCUSSION

The performance of the genetically tuned LQR controller was compared with a controller proposed in the literature [26]. The compared controller is designed with an inner-outer loop structure, LQR controller and integrative action. The coefficients of the controller were determined by trial-and-error method. In this study to evaluate the performance of the developed controller, its responses were observed by giving some references to the position and angle values. The parameters of the Parrot AR. 2.0 aircraft modelled in the MATLAB/Simulink environment are shown in Table 1.

Table 1. Physical parameters of Parrot AR. 2.0 quadcopter.

| m (kg) | L (m) | I_x (kgm ²) | I_y (kgm ²) | I_z (kgm ²) |
|----------|---------|---------------------------|---------------------------|---------------------------|
| 0.46 | 0.127 | 2.24e-4 | 2.90e-4 | 5.30e-4 |

In the MATLAB/Simulink environment, the response of the quadcopter to the applied input to observe the response to the control inputs is shown in Figure 4.

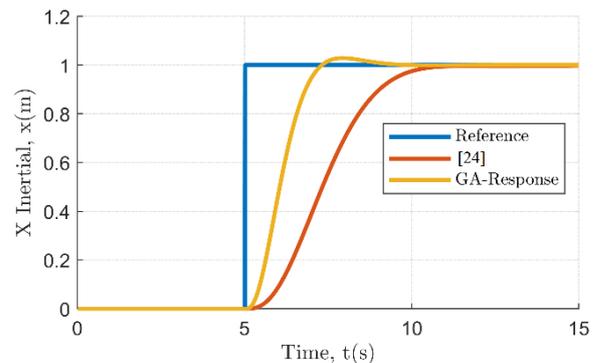


Figure 4. Reference response of the GA-LQR controller on the x -axis.

In this paper, the abbreviation GA-LQR is used for the proposed controller. The results obtained are shared in Table 4. Used in the tables: t_r represents the rise time (s) and M_p represents the overshoot (%). Accordingly, while the GA-LQR controller exhibited a faster rise performance to the 1-meter reference given on the x-axis, some overshoot was observed.

Table 4. Evaluation of the reference response in the x-axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 2.9677 | 0 |
| GA-LQR | 1.3629 | 2.8549 |

The behavior of the quadcopter to reach the reference on the y-axis is shown in Figure 5.

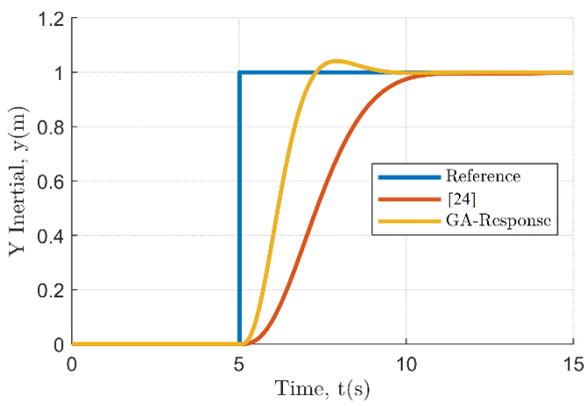


Figure 5. Reference response of the GA-LQR controller on the y-axis.

The developed GA-LQR controller exhibited similar behavior in the x and y axes. Obtained results are shown in Table 5.

Table 5. Evaluation of the reference response in the y-axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 2.9677 | 0 |
| GA-LQR | 1.3714 | 4.1294 |

As in the x-axis, the y-axis also reached the reference in a shorter time. In addition, some overshoot occurred. When evaluated in terms of rise times, it was observed that there was a two-fold difference.

Another evaluation signal for position control was applied for the z-axis. The responses obtained after giving the quadcopter a 1-meter elevation reference are shown in Figure 6. The GA-LQR controller reached the reference in the z-axis in a shorter time than the other position references. Similarly, it can be said to be advantageous in terms of settling time.

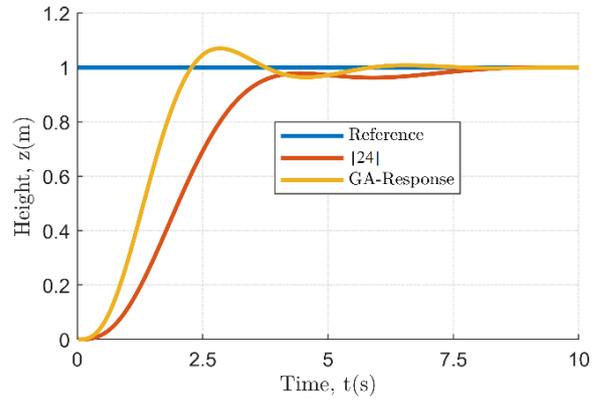


Figure 6. Reference response of the GA-LQR controller on the z-axis.

The results obtained on the z-axis are presented in Table 6.

Table 6. Evaluation of the reference response in the z-axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 2.3437 | 0.0628 |
| GA-LQR | 1.3532 | 6.9580 |

The GA-LQR controller showed faster responses in terms of rise time in the x and y axes. However, it was found that a certain amount of overshoot occurred. After comparing the responses to position references, roll, pitch, and yaw angle reference responses were also analyzed. A roll angle reference of 0.2 radians was applied to the quadcopter, and the controller response is shown in Figure 7.

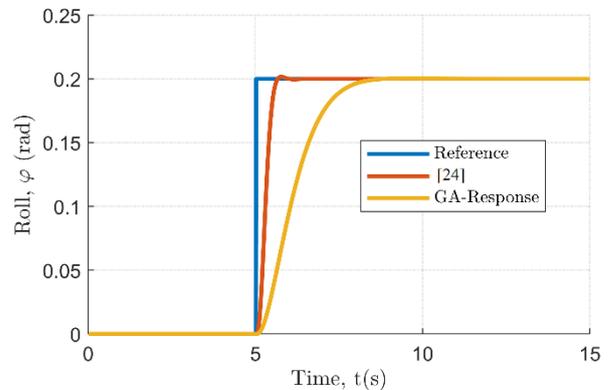


Figure 7. Reference response of GA-LQR controller for roll angle.

While the GA-LQR controller reached the roll reference angle later, no overshoot occurred this time. The results of the responses to the roll angle reference are presented in Table 7.

Table 7. Evaluation of the reference response in the roll axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 0.3529 | 0.8185 |
| GA-LQR | 1.7875 | 0.1639 |

Similarly, the controller response to a pitch angle reference of 0.2 radians is shown in Figure 8.

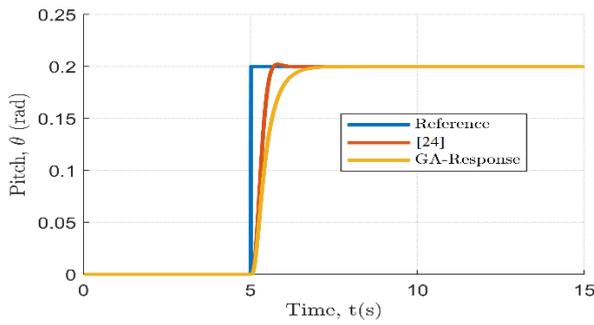


Figure 8. Reference response of GA-LQR controller for pitch angle.

It is observed that the response produced against the roll angle reference is produced in the same way for the pitch angle reference. The reactions occurring on the pitch axis are shared in Table 8.

Table 8. Evaluation of the reference response in the pitch axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 0.3706 | 1.0163 |
| GA-LQR | 0.7955 | 0 |

When the responses at the pitch angle reference are analyzed according to Table 8, it is observed that no overshoot occurred in the GA-LQR controller. The responses of the controllers to the yaw angle reference are shown in Figure 9.

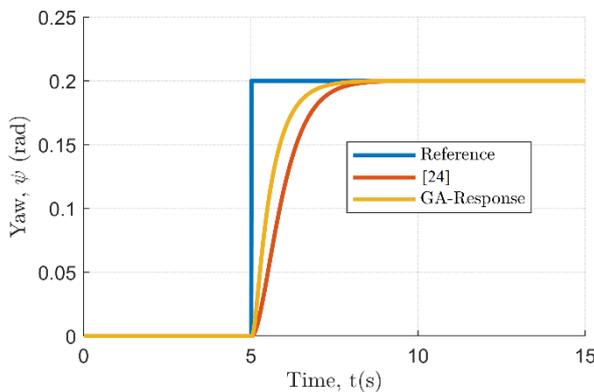


Figure 9. Reference response of the GA-LQR controller for yaw angle.

When the data obtained at the yaw angle were analyzed, it was determined that an overshoot did not occur in both controllers. The data obtained are shown in Table 9.

Table 9. Evaluation of the reference response in the yaw axis.

| Controller | t_r (s) | M_p (%) |
|------------|-----------|-----------|
| [24] | 1.6421 | 0 |
| GA-LQR | 1.2079 | 0 |

4.1. Stability Analysis

The developed controller is expected to achieve success in certain performance criteria. However, stability analyses of the controller should also be performed. The stability of a system expressed in state-space representation can be analyzed by looking at the eigenvalues of the A matrix. Quadcopter controllers are arranged in a close-loop fashion with the coefficients determined by GA-LQR. The eigenvalues of the obtained through MATLAB system are shown in Table 10.

Table 10. Closed-loop eigenvalue analysis with optimized coefficients.

| Axis Controller | Close-loop eigenvalues |
|-----------------|-------------------------------------|
| x | $\{-1.3106 \pm 1.1906i, -4.8039\}$ |
| y | $\{-1.1924 \pm 1.3667i, -2.4882\}$ |
| z | $\{-1.3393 \pm 1.2512i, -2.0662\}$ |
| roll | $\{-1.3840 \pm 0.7178i, -55.7223\}$ |
| pitch | $\{-2.9947, -8.6620, -29.2386\}$ |
| yaw | $\{-1.8593, -11.2946, -100.9108\}$ |

As seen in Table 10, all eigenvalues have negative values. It has been determined that the control system is stable because the real parts of the closed loop poles are negative.

After observing the success of the developed GA-LQR controller in tracking only one reference, it was also tested to track multiple references simultaneously. The GA-LQR controller is designed to track x, y and z position references, roll, pitch, and yaw angle references as stated in section 3. In this context, the ability to follow the trajectory given as a reference for x, y and z position controllers is shown in Figure 10. The trajectory tracking capability of the GA-LQR controller at x, y and z positions is demonstrated by the tests performed.

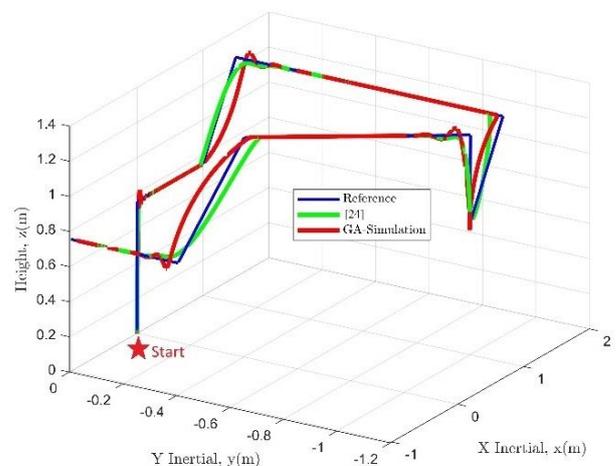


Figure 10. Control response of GA-LQR controller for trajectory tracking

The responses in the x-axis following the reference trajectory shown in Figure 11 are presented in Figure 12.

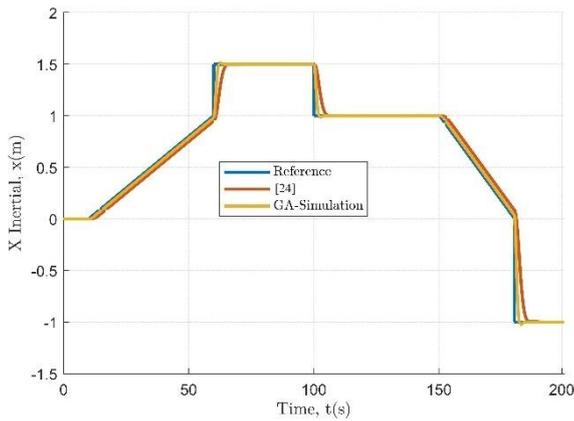


Figure 11. Control response of the GA-LQR controller in the x-axis for trajectory tracking

According to the data shown in Figure 11, it is observed that the GA-LQR controller provides superiority in terms of rise time in position references. However, some overshoots were also observed. The controller responses on the y-axis during the tracking of the reference trajectory are shown in Figure 12.

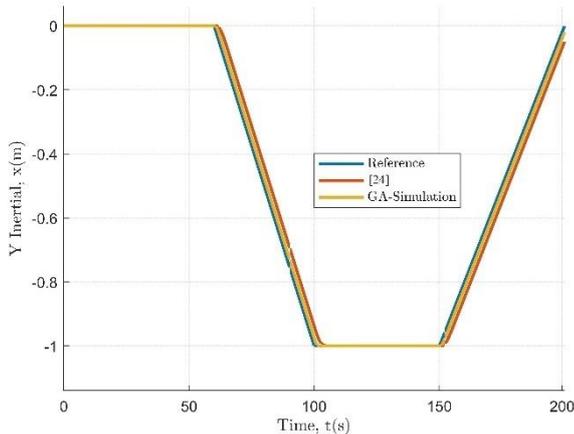


Figure 12. Control response of the GA-LQR controller in the y-axis for trajectory tracking.

It is again shown that the GA-LQR controller is successful in terms of rise time in the y-axis as well as in the x-axis. The responses in the z-axis during the tracking of the reference trajectory are shown in Figure 13.

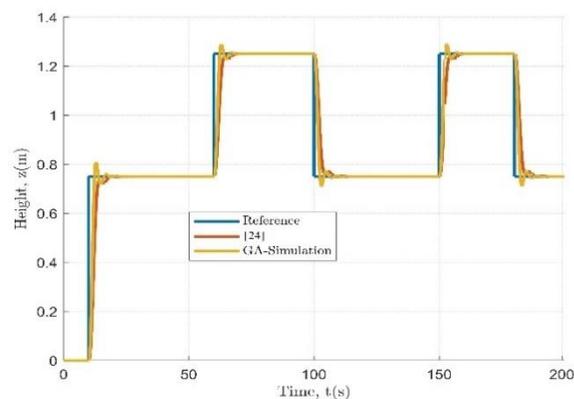


Figure 13. Control response of the GA-LQR controller in the z-axis for trajectory tracking.

When the responses occurring in the z-axis are analyzed, it is determined that the GA-LQR controller is more advantageous in terms of response time while overshoot occurs. The results of the evaluation according to the root-mean-square error (RMSE) criterion in the trajectory tracking scenario shown in Figure 10 in the Simulink environment are shown in Table 11.

According to the results shared in Table 11, the LQR controller with integrative action and genetic tuning follows the reference better than the inner-outer loop structure proposed in this study according to the root-mean-square error value.

Table 11. Results were obtained according to the root-mean-square error criterion

| Controller | x (m) | y (m) | z (m) | Total |
|---------------|---------------|---------------|---------------|---------------|
| [24] | 0.0865 | 0.0714 | 0.0556 | 0.2135 |
| GA-LQR | 0.0756 | 0.0143 | 0.0926 | 0.1825 |

As a result of the experiments carried out in MATLAB/Simulink environment for both tracking the reference in one axis and tracking multiple references, the reference tracking capability of the GA-LQR controller has been increased. It has been shown that not only more successful but also significant advantages, such as automatic detection of the controller coefficients, have been achieved.

According to the experimental results obtained from optimizing the LQR controller with GA, it was more successful. At this point, if the rise time, settling time or overshoot is essential for the area of use, the performance requirements will be met by adjusting the objective function of the GA according to this situation. Since such a situation was unimportant in the experiments, organizing the objective function according to the RMSE criterion.

In addition, the files of the quadrotor modelled in MATLAB/Simulink environment are shared at <https://github.com/atahirkarasahin/GA-LQR.git>.

5. CONCLUSION

In this paper, a genetically tuned LQR controller with integrative action operating according to the inner-outer loop structure is proposed. First, an integrator is added to the LQR controller and the system model in state-space form is adapted accordingly. With the addition of the integrator, perturbations and steady-state errors occurring in the LQR controller are eliminated. The developed GA-LQR controller is compared with a study in the literature in MATLAB/Simulink environment. With the proposed GA-LQR controller, an improvement in trajectory tracking is achieved with respect to the RMSE value. The genetically tuned LQR controller has achieved successful results both in terms of controller design time and according to the specified performance criteria. As a result of the tests performed, if the overshoot amount is an important criterion for the designer, different configurations can be realized by adjusting the

parameters in the GA to increase its success in this performance criterion.

Author contributions: Ali Tahir KARAŞAHİN: Conceptualization, Methodology, Software, Data curation, Writing- Reviewing and Editing.

Conflict of Interest: There is no conflict of interest.

Financial Disclosure: The study was funded by Sakarya University of Applied Sciences BAP with project number 078-2022.

REFERENCES

- [1] C.-C. Chang, J.-L. Wang, C.-Y. Chang, M.-C. Liang, and M.-R. Lin, "Development of a multicopter-carried whole air sampling apparatus and its applications in environmental studies," *Chemosphere*, vol. 144, pp. 484–492, 2016.
- [2] J. A. Paredes, J. González, C. Saito, and A. Flores, "Multispectral imaging system with UAV integration capabilities for crop analysis," in *2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE)*, 2017, pp. 1–4.
- [3] S. Anweiler and D. Piwowarski, "Multicopter platform prototype for environmental monitoring," *J Clean Prod*, vol. 155, pp. 204–211, 2017.
- [4] B. E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 244–249.
- [5] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 998–1007.
- [6] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *Int J Control Autom Syst*, vol. 7, pp. 419–428, 2009.
- [7] J. Farrell, M. Sharma, and M. Polycarpou, "Backstepping-based flight control with adaptive function approximation," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1089–1102, 2005.
- [8] Z. Zuo and C. Wang, "Adaptive trajectory tracking control of output constrained multi-rotors systems," *IET Control Theory & Applications*, vol. 8, no. 13, pp. 1163–1174, 2014.
- [9] J. Spencer, J. Lee, J. A. Paredes, A. Goel, and D. Bernstein, "An adaptive pid autotuner for multicopters with experimental results," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7846–7853.
- [10] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, vol. 21, no. 4, pp. 1400–1406, 2012.
- [11] E. A. Niit and W. J. Smit, "Integration of model reference adaptive control (MRAC) with PX4 firmware for quadcopters," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2017, pp. 1–6.
- [12] A. R. Dooraki and D.-J. Lee, "Reinforcement learning based flight controller capable of controlling a quadcopter with four, three and two working motors," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, 2020, pp. 161–166.
- [13] C. Guzay and T. Kumbasar, "Aggressive maneuvering of a quadcopter via differential flatness-based fuzzy controllers: From tuning to experiments," *Appl Soft Comput*, vol. 126, p. 109223, 2022.
- [14] G. Unal, "Integrated design of fault-tolerant control for flight control systems using observer and fuzzy logic," *Aircraft Engineering and Aerospace Technology*, vol. 93, no. 4, pp. 723–732, 2021.
- [15] E. Yazid, M. Garrat, F. S.-2018 I. Conference, and undefined 2018, "Optimal PD tracking control of a quadcopter drone using adaptive PSO algorithm," in *ieeexplore.ieee.org*, 2018, pp. 146–151.
- [16] M. S. Can and H. Ercan, "Real-time tuning of PID controller based on optimization algorithms for a quadrotor," *Aircraft Engineering and Aerospace Technology*, vol. 94, no. 3, pp. 418–430, 2021.
- [17] Ş. YILDIRIM, N. ÇABUK, and V. BAKIRCIOĞLU, "Optimal Pid Controller Design for Trajectory Tracking of a Dodecarotor Uav Based On Grey Wolf Optimizer," *Konya Journal of Engineering Sciences*, vol. 11, no. 1, pp. 10–20, 2023.
- [18] I. Siti, M. Mjahed, H. Ayad, and A. El Kari, "New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods," *Applied Sciences*, vol. 9, no. 9, p. 1780, 2019.
- [19] M. J. Mahmoodabadi and N. R. Babak, "Robust fuzzy linear quadratic regulator control optimized by multi-objective high exploration particle swarm optimization for a 4 degree-of-freedom quadrotor," *Aerosp Sci Technol*, vol. 97, p. 105598, 2020.
- [20] M. N. Shauqee, P. Rajendran, and N. M. Suhadis, "Proportional double derivative linear quadratic regulator controller using improvised grey wolf optimization technique to control quadcopter," *Applied Sciences*, vol. 11, no. 6, p. 2699, 2021.
- [21] L. Meier, ... D. H.-2015 I. international, and undefined 2015, "PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms," *ieeexplore.ieee.org*.
- [22] "ArduPilot Documentation — ArduPilot documentation." Accessed: Jun. 13, 2023. [Online]. Available: <https://ardupilot.org/ardupilot/>

- [23] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot Autom Mag*, vol. 19, no. 3, pp. 20–32, 2012.
- [24] L. Martins, "Linear and nonlinear control of uavs: design and experimental validation," Master's Thesis, Instituto Superior Técnico, Lisbon, Portugal, 2019.
- [25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics," 2009, doi: 10.1007/978-1-84628-642-1.
- [26] L. Martins, C. Carneira, and P. Oliveira, "Linear quadratic regulator for trajectory tracking of a quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019.
- [27] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.