# Design of LQR Controller for a Quadrator to Execute UAV Capture and Return

Ayesha Akhtar
*Robotics Engineering*
*Worcester Polytechnic Institute*
Worcester, MA, USA
aakhtar@wpi.edu

Chirag Patel
*Robotics Engineering*
*Worcester Polytechnic Institute*
Worcester, MA, USA
cpatel@wpi.edu

Dharmik Raval
*Robotics Engineering*
*Worcester Polytechnic Institute*
Worcester, MA, USA
draval@wpi.edu

Sumanth Pashuparthi
*Robotics Engineering*
*Worcester Polytechnic Institute*
Worcester, MA, USA
spashuparthi@wpi.edu

***Abstract*—This project simulates the control and stable flight of a quadrotor, focusing on achieving three flight maneuvers: stable hover, intercepting an unknown drone's trajectory using a Linear Quadratic Regulator (LQR) control system, and returning with the captured drone while maintaining stability under randomized disturbances. By successfully simulating these maneuvers, the project demonstrates the effectiveness of LQR control for robust quadrotor operations in object capture and return missions.**

***Index Terms*—Quadrotor Control, Linear Quadratic Regulator (LQR), Trajectory Tracking, Aerial Capture, Unmanned Aerial Vehicles (UAVs), Robust Control, Nonlinear Systems**

## I. Introduction

Quadrotors, unmanned aerial vehicles with four rotors, offer unique maneuverability but present challenges due to their complex, non-linear flight dynamics [1]. Precise control strategies are crucial for tasks demanding real-time responsiveness and stable execution. This project explores the use of a Linear Quadratic Regulator (LQR) control system to simulate a specific scenario: a quadrotor needs to track an unknown drone, capture it, and return safely while encountering disturbances.

To achieve the objective of robust quadrotor control for target intercept, capture, and return, this project adopts a comprehensive approach. First, a thorough understanding of the quadrotor's dynamics is established. Then, the non-linear dynamic model will be strategically linearized. Finally, an LQR controller will be designed and implemented, considering constraints for successful mission execution. Throughout this exploration, we will delve into the details of the control design approach, highlighting the considerations made to achieve a robust and effective solution.

## II. Literature Review

Quadrotors have emerged as a versatile platform for various applications due to their vertical take-off and landing (VTOL) capabilities and exceptional maneuverability. However, their inherent non-linear dynamics pose a significant challenge for achieving precise control [1]. Extensive research has been conducted to develop effective control strategies for quadrotor flight, with a particular focus on tasks requiring real-time responsiveness and stability.

Linear Quadratic Regulator (LQR) control is a well-established technique known for its effectiveness in regulating linear systems. It offers an optimal control solution by minimizing a quadratic cost function that incorporates system states and control inputs [2]. The application of LQR control to quadrotor flight control has been explored in numerous studies. For instance, [3] demonstrates the successful implementation of LQR control for trajectory tracking of a quadrotor, achieving good performance despite external disturbances. Similarly, [4] utilizes LQR control for a quadrotor system, emphasizing its ability to handle parameter uncertainties.

Beyond LQR control, other control strategies have also been investigated for quadrotor flight. Sliding Mode Control (SMC) is another popular approach known for its robustness to disturbances [5]. Studies such as [6] showcase the effectiveness of SMC in quadrotor trajectory tracking and disturbance rejection. The choice of control strategy depends on the specific requirements of the application. In this project, LQR control is chosen due to its focus on optimal control and its well-established theoretical framework, making it suitable for the simulated scenario of quadrotor intercept, capture, and return.

However, the non-linear nature of quadrotor dynamics necessitates linearization before applying LQR control. Linearization techniques, such as Taylor series expansion, are commonly employed to approximate the non-linear model around a specific operating point [7]. This linearization step is crucial for the successful implementation of LQR control in quadrotor systems.

The control of quadrotors presents a unique challenge due to their non-linear dynamics. LQR control offers a promising approach for achieving optimal control in linear systems, and its application to quadrotor flight control has been validated in various studies. This project leverages the established theory of LQR control while acknowledging the need for linearization of the quadrotor's non-linear model. By combining these elements, the project aims to design a robust control system for the simulated quadrotor mission of target intercept, capture, and return.

## III. Control design

### A. Approach

We will begin by generating the state space variables and defining the equations that define the state space. We will prove controllability of the system and choose a controller methodology. Using the controller, we will simulate hovering as well as moving to a location and hovering. Finally, we will introduce external forces and moments that would be experienced by simulating a target capture in the problem statement.

In order to simulate the quadrotor and its flight dynamics, the design will be coded into MATLAB, using an ODE solver to compute the state variables, with full state variables, $A$ and $B$ matrices, $K$ gain matrix, $K_{\text{capture}}$ gain matrix (incorporating $n$ and $r$ forces and torques from the target), as well as characteristics for graphing.

### B. Methodology

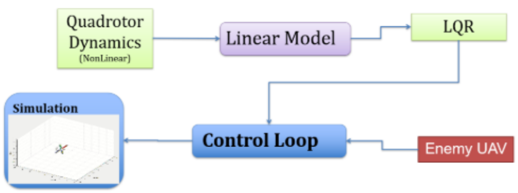


Fig. 1. Control Methodology

The methodology used in this project to achieve the mentioned task is to use Linear - Quadratic - Regulator (LQR) controller to control the quadrotor trajectory, track the enemy UAV, and bring it back to the nest position. To do so, we have to follow the control law of the LQR controller, in which the first step is to linearize the quadrotor dynamics using a linear model. Hence, the system is represented by linear differential equations:

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

where:

- $x(t)$ is the state vector.
- $u(t)$ is the control input vector.
- $A$ is the system matrix.
- $B$ is the input matrix.

### C. Frames and Constants

Coordinate frames are identified for the reference/inertial frame $E = [e1, e2, e3]$ at the center bottom of the airspace and the body frame $C = [c1, c2, c3]$. All rotors are equidistant from the center of mass and in the same $x - y$ plane as the body's center of mass. The external forces and moments on the system are represented by $r$ and $n$, where $r = r1c1 + r2c2 + r3c3$ and $n = n1c1 + n2c2 + n3c3$ are directly applied to the center of mass. We are assuming that the torque of the rotor is proportionally related to the input thrust via the constant $\sigma > 0$, for $\tau_i = \sigma u_i$. We will be utilizing $I$ as our diagonal inertial matrix where diagonal elements $I_x$, $I_y$, and $I_z$ represent the mass moments of inertia about $c1$, $c2$, and $c3$ respectively. For the purpose of our analysis, we will assume simplified aerodynamic effects and ignore drag, ground effects, translational lift, blade flapping, feathering and lagging, and other aerodynamic complexities.

### D. State Variable

Development of the quadrotor state equations are found in reference [8]. We'll establish the state variable, $z$, to represent the quadrotor's $xyz$ center-of-mass position in the earth frame $x = [x_1\, x_2\, x_3]^T$; the roll, pitch, yaw angles in the earth frame, $\alpha = [\phi\, \theta\, \psi]^T$; the $xyz$ linear velocities in the earth frame, $v = [v_1\, v_2\, v_3]^T$; and the angular accelerations $\omega = [\omega_1\, \omega_2\, \omega_3]^T$ in the earth roll, pitch, yaw directions. Thus, our state variable is $z = [x\, \alpha\, v\, \omega]^T$. We define $\dot{z}$ as $[\dot{x}\, \dot{\alpha}\, \dot{v}\, \dot{\omega}]^T$, with:

$$\dot{x} = v \tag{1}$$

$$\dot{\alpha} = T^{-1}\omega \tag{2}$$

$$\dot{v} = -ge_3 + \frac{R_{C/E}}{m}(u_1 + u_2 + u_3 + u_4)c_3 + \frac{R_{C/E^r}}{m} \tag{3}$$

$$\dot{\omega} = I^{-1}((u_2-u_4)lc_1+(u_3-u_1)lc_2+(u_1-u_2+u_3-u_4)\sigma c_3+n-\omega\times I\omega) \tag{4}$$

where u1, u2, u3 and u4 can be defined as control inputs and $R_{C/E}$ is defined in the problem statement as the Euler rotation matrix from quadrotor frame $C$ to earth frame $E$ and $T^{-1}$, also defined in the problem statement, relates the angular rate of change of the Euler angles based on the angular velocity of the quadrotor. Our system is nonlinear because it is underactuated; four actuators for six degrees of freedom. We can utilize linearization to simplify this model and approximate its behavior for easier control. To do this, we approximate our system as $\dot{z} = Az + Bu$. We derive the $A$ matrix as the Jacobian of our $\dot{z}$ by $z$, and our $B$ as the Jacobian of our $\dot{z}$ by $u$. For our approximation, we model the quadrotor in a stable hovering position, where a given state is $z = [x\, y\, z\, 0]^T$; no velocities, accelerations, and its orientation level. We assume a base output of $\frac{mg}{4}$, or each motor outputting the necessary force of a hover. We assign these values to our derived $A$ and $B$ matrices. We then confirm that our system is controllable with these matrices. We create a controllability matrix $C$, which we define as $C = [A^0B\, A^1B\, A^2B \ldots A^n - 3B\, A^n - 2B\, A^n - 1B]$, where $n$ is the size of our input, 12. We check the rank of the controllability matrix $C$ to assure that we have a rank equal to the number of states in our $z$ vector. the rank of our derived controllability matrix is 12.

Other system parameters can be defined as:

$$\begin{aligned}
g &= 9.81\,\text{m/s}^2 \\
l &= 0.2\,\text{m} \\
m &= 0.5\,\text{kg} \\
I &= \begin{bmatrix}1.24 & 1.24 & 2.48\end{bmatrix}\,\text{kg m}^2 \\
\mu &= 3.0\,\text{N} \\
\sigma &= 0.01\,\text{m} \\
I_{11} &= 1.24\,\text{kg m}^2 \\
I_{22} &= 1.24\,\text{kg m}^2 \\
I_{33} &= 2.48\,\text{kg m}^2
\end{aligned}$$

We selected the Linear-Quadratic Regulator (LQR) approach as an optimal control technique for generating the gain matrix,

$K$. LQR operates on our linearized system, utilizing both $A$ and $B$ matrices (states and inputs, respectively). LQR also utilizes two additional matrices - $Q$ and $R$, which correspond to $A$ and $B$, respectively. $Q$'s and $R$'s purpose are to impose a weighted cost such that there will be a trade-off between particular elements of the state and inputs in the optimization. Since we are not asked to weigh the cost of using our actuators (as if we were conserving fuel), we have set the $4 \times 4$ $R$ matrix as an identity matrix. Our $12 \times 12$ $Q$ matrix starts as an identity matrix, but we found through experimentation that our desired performance required adjustments we discuss later. There is an additional matrix, $N$, which acts as a penalty to the interactions between state variables and inputs. We simplify by leaving $N$ as an identity matrix.

LQR acts on the linearized system of the form $\dot{x} = Ax + Bu$ and calculates a cost function based on optimization. This integrates the costs of both $Q$ and $R$ for comparison:

$$J = x_0^T F(0) x(t_0) + \int_{t_0}^{t_f} x^T Q x + u^T R u + 2x^T N u \, dt \quad (5)$$

$$K = R^{-1}(B^T P(t) + N^T) \quad (6)$$

where $P$ is a positive semi-definite matrix to be determined. The optimal control law is derived as follows:

$$u(t) = -Kx(t)$$

where $K$ is the optimal feedback gain, calculated as

$$K = R^{-1} B^T P$$

which is known as the control gain. Once we solve the Algebraic Riccati Equation to find $P$, we have to compute the optimal gain $K$ using $P$. Finally, implement the control law $u(t) = -Kx(t)$ in the system.

$$A^T P(t) + P(t)A - (P(t)B + N) R^{-1} (B^T P(t) + N^T) + Q = \dot{P}(t) \quad (7)$$

## IV. Implementation

- The foundation of simulating the controller lies in solving the system of differential equations governing the dynamics of our original system while incorporating the additional influence from our Linear Quadratic Regulator (LQR) controller. We encapsulated the changes in the environment within a separate function to serve as the Ordinary Differential Equation (ODE) function. This allows us to accommodate time-varying adjustments to the target path, as well as changes in the torques and forces (represented by $n$ and $r$) after capture.
- To showcase successful control of our quadrotor, we established an initial state where the altitude $z$ is at ground level ($z = 0$) and a desired state $\mathbf{zd} = [5, 5, 5, 0]^T$ to demonstrate movement and hovering. By inputting these states into our ODE solver, we achieve smooth movement and hovering at the specified altitude $z_d$.
- The control logic is designed in such a way the quadrotor continues gets the target trajectory values from the time iteration loop and gets the altitude update.Boundary Conditions are set in such a way that the UAV gets captured if it gets into the l/2 neighbourhood of the quadrotor and bring it back to the nest positioned at origin.If the quadrotor is not able to capture the UAV in the stipulated time and escapes,the quadrotor is bound to return to base.
- To model post-capture disturbance forces and torques, we induce changes in $n$ and $r$ within the ODE function code. This prompts a recalculation of the control term $u$ to adapt to the altered dynamics. Mission parameters dictate that the magnitude of the moment $||n||$ must not exceed 1, and the magnitude of the force $||r||$ must not exceed 2. Initially, we test the system's response to constant moments and forces in each direction before considering more complex scenarios, as the system must reliably handle constant disturbances before tackling time-varying or randomized forces.

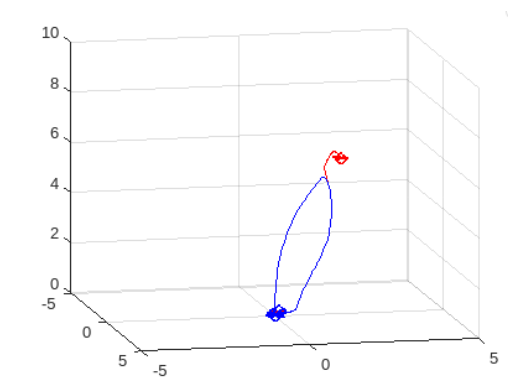


Fig. 2. Plot showing the capture of UAV and return to base

- As the disturbance forces tend to deviate the return trajectory,a new Set of QR gains are defined and the resultant gain is input to the system to nullify these.
- Based on the simulation results,QR gains are tuned accordingly on trial and error basis to modify the behaviour of the system.

## V. Results

In the presented results, we observe the X, Y, and Z position trajectories of both a quadrotor and an Unmanned Aerial Vehicle (UAV) across a randomly generated flight paths executed by the UAV. These trajectories are depicted in a series of plots, where the dashed lines illustrate the quadrotor's paths over a simulation duration of 10 seconds. In contrast, the solid lines represent the corresponding trajectories of the UAV.

A notable aspect of these trajectories is that, after an initial period, there is a visible convergence between the paths of the quadrotor and the UAV. This convergence becomes more pronounced with time, eventually leading to a complete alignment of their trajectories.

The significance of this convergence lies in its representation of the quadrotor successfully "catching" or intercepting

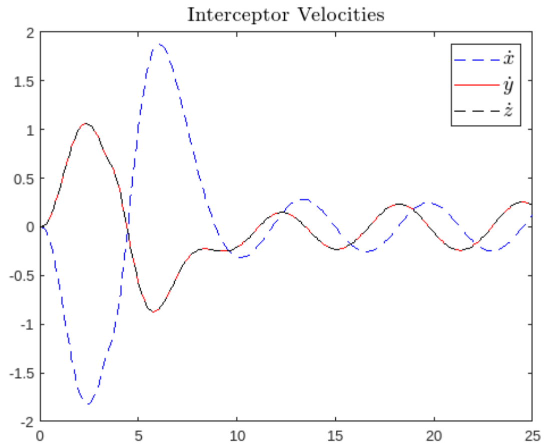


Fig. 3. Quadrotor and UAV Velocities

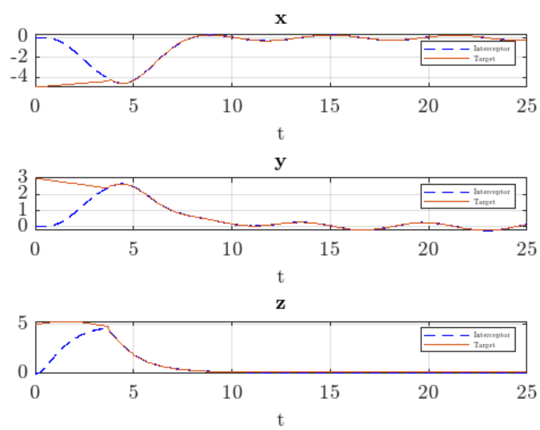


Fig. 4. Random Trajectory Plots

the UAV. Following this interception, the alignment of their paths symbolizes the joint ”return” of both the quadrotor and the UAV, indicating a successful completion of the simulated mission. This behavior, captured in the plots, is important in understanding the effectiveness of the control algorithm used in guiding the quadrotor to intercept and subsequently bring back the UAV to the nest position.

## VI. Conclusion

The approach presented in this report effectively addresses the basic objectives of the assigned mission, such as moving to and hovering at a desired location, as well as intercepting an intruding target UAV. However, it falls short in handling the resultant forces exerted by the target UAV when resisting capture. This indicates a limitation stemming from our reliance on linearization around a simplified hovering system.

Although the Linear Quadratic Regulator (LQR) framework facilitated the specification of multiple pairs of weighting matrices ($Q$ and $R$) to tailor the controller's behavior in different states, it proved insufficient in preventing the quadrotor from reaching a catastrophic failure state as some UAV trajectories are not able to be properly intercepted by Quadrotor.

## References

[1] Lee, D., Kim, S., Kang, J. (2016). Nonlinear flight control of a quadrotor using feedback linearization. International Journal of Control, Automation, and Systems, 14(8), 3708-3716.
[2] Bryson, A. E., Ho, Y. C. (1971). Applied optimal control. Hemisphere Publishing Corporation.
[3] Bouabdallah, S., Foufou, S., Siegwart, R. (2007). Design and control of a quad-rotor helicopter for autonomous flying. In 2007 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3992-3997). IEEE.
[4] Aguiar, A. P., Hespanha, J. P. (2007). Trajectory-tracking and stabilization for underactuated vehicles with time-varying uncertainties. IEEE Transactions on Automatic Control, 52(8), 1493-1503.
[5] Utkin, V. I. (1992). Sliding mode in control and optimization. Springer-Verlag.
[6] Li, Y., Yang, Z., He, Y. (2013). Trajectory tracking control of a quadrotor UAV using composite sliding mode control. Mathematical Problems in Engineering, 2013.
[7] Slotine, J. J., Li, W. (1991). Applied nonlinear control. Prentice Hall.
[8] Faal, Siamak. Notes on Control of Quadrotors. Class notes for RBE 502.