Application of Linear Quadratic Regulator with Integral Action (LQRI) in Quadrotor Control: Simulating UAV Capture and Retrieval

Jared Morgan*, Wenpeng Wang†, Cooper Mann‡, Ryo Murakami§

December 2023

Abstract

The field of Unmanned Aerial Vehicles (UAVs) is being studied actively, but in some cases, its inherent physics such as nonlinearity makes it hard to design a controller. There are numerous types of controllers proposed and it is important to select an appropriate algorithm based on the task and the resources. For this project, we use a Linear Quadratic Regulator with Integral Action (LQRI) for controlling quadrotors in UAV capture and retrieval tasks. The controller's performance is evaluated through simulations that illustrate the quadrotor's behavior and response in capturing and retrieving an invading UAV. While effective in most scenarios, instances of instability under excessive external disturbances highlight the potential need for exploring nonlinear or hybrid control strategies for more complex missions. Our findings demonstrate the viability of LQRI in quadrotor control.

1 Introduction

The field of Unmanned Aerial Vehicles (UAVs), particularly quadrotors, represents a rapidly evolving area of hobby and research, driven by their increasing applications in various domains such as surveillance, delivery services, and environmental monitoring. The inherent complexity in the control and dynamics of quadrotors, owing to their nonlinear, coupled, and under-actuated nature, has spurred extensive research into effective control schemes [1]–[3].

A notable aspect of this research lies in the exploration of various control strategies to address the challenges posed by quadrotor dynamics. For instance, the work by Sabatino delves into mathematical modeling using Newton's and Euler's laws, comparing linear control strategies like the Linear Quadratic Regulator (LQR) with nonlinear feedback linearization techniques [1]. Ahmad *et al.* further emphasize the effectiveness of LQR-based control in managing the position and yaw of quadrotors [2]. Meanwhile, Sharma introduces a approach using a rotation matrix-based controller to accommodate uncertainties and external disturbances in quadrotor control [3].

Other researchers have proposed hybrid and advanced control methods. Wang *et al.* address trajectory tracking through a hybrid finite-time control solution [4], and Wong *et al.* focus on model linearization and H-infinity controller design for improved positional control [5]. Additionally, the backstepping control approach, as investigated by Madani and Benallegue, offers a robust solution for quadrotor control, utilizing the full-state backstepping technique grounded in Lyapunov stability theory [6]. These diverse methodologies underscore the active exploration and innovation in quadrotor control systems.

In this context, our project aligns with these efforts, particularly focusing on the utilization of LQR for controlling quadrotors in a simulated environment. Our approach leverages LQR's robustness in dealing with linear systems and its efficacy in managing the quadrotor's dynamic behaviors. This study aims to demonstrate the application of LQR in capturing an invading UAV and effectively bringing the object back to a designated location.

2 Methodology

System Linearization

In the context of controlling a quadrotor, which is inherently a nonlinear system, a crucial step involves the process of system linearization. This is necessary because many control strategies, including LQR, are primarily designed for linear systems. To transform nonlinear dynamics of a quadrotor into a linear approximation, we employ the Taylor series expansion. This mathematical technique approximates the nonlinear behavior of the quadrotor around a specific operating point, usually a steady state or equilibrium condition. This linear approximation allows for the application of linear control strategies, such as LQR, and we will briefly discuss the procedure related in obtaining the system's linear approximation.

The state space form of the system will be constructed

*jmmorgan4@wpi.edu
†wwang11@wpi.edu
‡cpmann@wpi.edu
§rmurakami@wpi.edu

here with state vector defined as $\mathbf{z} := [\mathbf{x}, \boldsymbol{\alpha}, \mathbf{v}, \boldsymbol{\omega}]^T$.

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

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

$$\dot{\mathbf{v}} = -g\mathbf{e_3} + \frac{1}{m}R_{C/E}(u_1 + u_2 + u_3 + u_4)\mathbf{c_3} + \frac{1}{m}R_{C/E}\mathbf{r} \tag{1c}$$

$$\dot{\boldsymbol{\omega}} = I^{-1}((u_2 - u_4)l\mathbf{c_1} + (u_3 - u_1)l\mathbf{c_2} + (u_1 - u_2 + u_3 - u_4)\sigma\mathbf{c_3} + \mathbf{n} - \boldsymbol{\omega}\boldsymbol{\omega}) \tag{1d}$$

From the state equations, we could tell for any arbitrary $\mathbf{x}^* \in \mathbb{R}^3$ and $\mathbf{u}^* = [1,1,1,1]^T \frac{mg}{4}$, the system is in equilibrium since $\dot{\mathbf{z}}^* = [\dot{\mathbf{x}}^*, \dot{\boldsymbol{\alpha}}^*, \dot{\mathbf{v}}^*, \dot{\boldsymbol{\omega}}^*] = \mathbf{0}$ by substituting $\mathbf{v}^* = \mathbf{0}$ to (1a), substituting $\boldsymbol{\omega}^* = \mathbf{0}$ to (1b), deriving $\dot{\mathbf{v}} = \mathbf{0}$ for (1c), and lastly substituting $\boldsymbol{\omega}^* = \mathbf{0}$ in (1d). Then a linear approximation could be construct at the equilibrium points. Based on Taylor's series, we can write the linear approximation of the state transition function as

$$\mathbf{f}(\mathbf{z},\mathbf{u}) \approx \mathbf{f}(\mathbf{z}^*,\mathbf{u}^*) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{(\mathbf{z},\mathbf{u})=(\mathbf{z}^*,\mathbf{u}^*)} (\mathbf{z}-\mathbf{z}^*) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{(\mathbf{z},\mathbf{u})=(\mathbf{z}^*,\mathbf{u}^*)} (\mathbf{u}-\mathbf{u}^*) \tag{2}$$

Based on the (2), we could formulate the system as linear into

$$\dot{\bar{\mathbf{z}}} = A\bar{\mathbf{z}} + B\bar{\mathbf{u}} \tag{3}$$

The corresponding matrices are defined as

$$A := \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{(\mathbf{z},\mathbf{u})=(\mathbf{z}^*,\mathbf{u}^*)}, \text{and } B := \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{(\mathbf{z},\mathbf{u})=(\mathbf{z}^*,\mathbf{u}^*)} \tag{4}$$

The result of A and B matrix are:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1/m & 1/m & 1/m & 1/m \\ 0 & l/I_{11} & 0 & -l/I_{11} \\ -l/I_{22} & 0 & l/I_{22} & 0 \\ \sigma/I_{33} & -\sigma/I_{33} & \sigma/I_{33} & -\sigma/I_{33} \end{bmatrix} \tag{6}$$

where, $\mathbf{0} \in R^{8\times 1}$.

To check if the linear approximation of the system about $(\mathbf{z}^*, \mathbf{u}^*)$ is controllable, we need to check the rank of controllability matrix

$$\mathcal{C} = \begin{bmatrix} A & AB & A^2B & ... & A^{11}B \end{bmatrix} \tag{7}$$

After plugging in the result from (5, 6), the controllability matrix has full rank, where its columns are linearly independent. Thus, the system's linear approximation about its equilibrium point is controllable, and we can proceed with controller design next.

LQRI Controller Design

For the design of the LQRI controller, MATLAB was utilized to derive the optimal gain. First, the known values for the system are substituted into the two-state matrices A and B. Next, Bryson's rule was utilized to determine initial values for the state-cost and input-cost matrices. The following outlines the tuned values for these matrices.

$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$R = I \in R^{4\times 4} \tag{9}$$

For the weights, z distance and velocity were weighted heavier than x and y distances and velocities. This is because we want the drone to reach the intended target in the z direction quicker. After applying LQR control, the following optimal gain matrix was obtained.

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

This gain matrix is then applied to the system input. The integral component is utilized to reduce the steady-state error still present in the system. It is applied by integrating the difference between the current and desired states of the quadrotor [7]. Notably, the integral gain was not applied to the angle because attempting to force the drone upright may cause it to not apply enough force to resist the UAV. As discussed later in the report, this addition drastically improved the stability of returning the UAV to the start position.

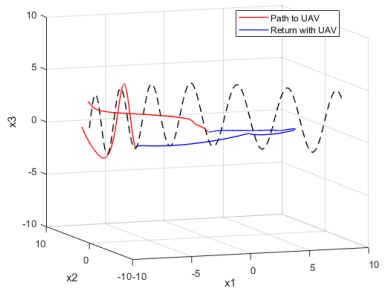

Figure 1: Simulated Airspace of a Stable Example.

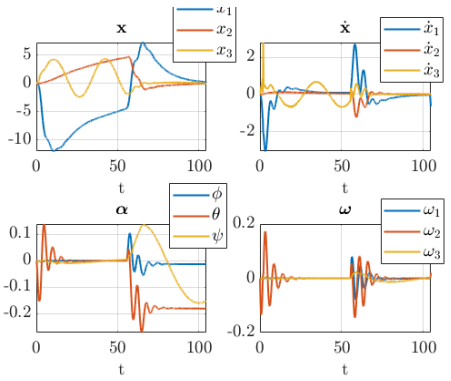

Figure 2: State Responses of a Stable Example.

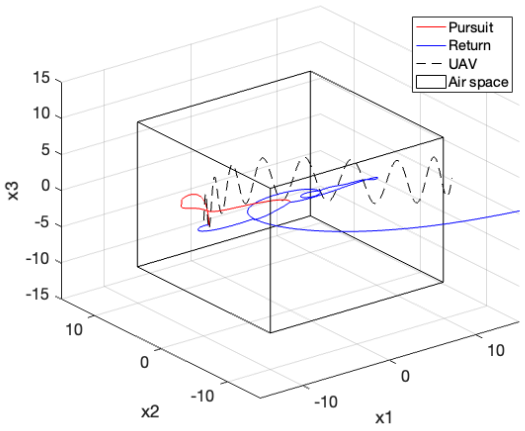

Figure 3: Simulated Airspace of an Unstable Example.

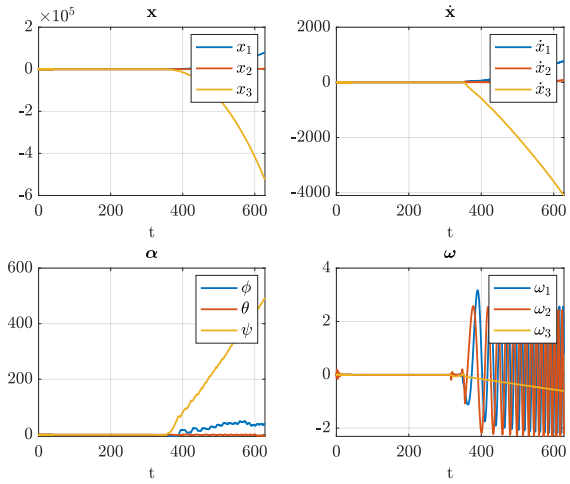

Figure 4: State Responses of an Unstable Example.

3 Simulation Results

In this section, we will present the results of our simulation using the proposed controller. For each set of results, we will provide two key components: a visualization of the quadrotor's behavior in a simulated airspace and a detailed state responses. The simulated airspace view will give a clear picture of the quadrotor's movement patterns, including its trajectory and interactions with any external elements, which is an invading UAV. The state response analysis will delve into the technical aspects, such as position, velocity, and pose changes over time, offering insight into the effectiveness and precision of the controller in real-time scenarios. This comprehensive presentation aims to illustrate both the practical and theoretical efficacy of our control approach in a simulated environment.

The above Figures 1 and 2 shows the quadrotor converges to the nest location and stops there after capturing the invading UAV. One should notice that, since the disturbance is randomly generated by the captured UAV to the quadrotor, in some cases the system could yield some unstable behavior. The Figures 3 and 4 are intended to show the unstable behavior mentioned. The cause of the unstable behavior is because of the acceptable force and torque in a given direction is dependant on the drone's angle. If the drone were, for instance, already angled to push back against the UAV, a sudden applied torque in the direction the UAV is pushing could cause the UAV to turn upside down or any other manner of destabilize.

The drone is capable of resisting some of the forces and torques specified in the problem statement. The drone can adapt to applied forces up to $r = [2, 0.2, 2]^T$ Newtons and applied torques up to $n = [0.1, 0.2, 0.01]^T$ Newton-meters.

Further simulations reveal the controller is more affective depending on the UAV's motion. If the UAV is traveling in a straight line, the drone can capture it at speeds up to 60 m/s. If the UAV is traveling in a spiral, the drone cannot capture it at speeds greater than 6m/s.

4 Discussions

As the team researched potential controllers to implement for the quadrotor, linear, nonlinear, and hybrid controllers were considered in the design. Linear controllers are simple, yet can still be applied to the full dynamic model of

a quadrotor. Nonlinear controllers such as H-infinity and backstepping control can be extremely efficient, yet also computationally intensive with a complex design. Hybrid controllers are a mix of two controller implementations [8]. The group decided to implement a linear controller due to a simpler implementation for the system. The linear approach aligned well with the system model and the requirements prior to the UAV's introduction and force application.

After the UAV was introduced, the failures of the LQRI system became more apparent. The most apparent inadequacy is the failure to accept applied forces greater than 0.2 Newtons in the x_2 direction. This is because the drone's linearization assumes small angles. The small angles assumption is no longer valid under an applied force because the drone rotates itself to resist the force. The x_2 direction is most affected by this because the first derivative of cosine at 0 is a worse approximation than the first derivative of sine at 0. The small angles assumption is also why the drone is unable to recover from an inversion and therefore unable to resist the fully expected torque of up to 1 Newton-meter.

The quality of the LQRI system was also inadequate when pursuing the UAV under circumstances where the integral term cannot properly increase. When the drone is traveling in a climbing spiral, the linear integral term can only increase in one direction meaning the drone is unable to reach the UAV in the other two directions. Instead, the drone follows behind the UAV in this configuration, failing to reach it at a given moment.

To improve the system's performance, the LQRI controller can be modified to account for more edge cases. For instance, the controller could consider its angle nonabsolutely and then turn its motors off while inverted and wait for its natural motion to right itself. This would solve some issues, but ultimately the only solution to this problem that satisfies all conditions is an nonlinear controller.

5 Conclusions

In this project, we aimed to design the controller for the quadrotor which pursues the invading UAV and takes it back to the nest by managing the UAV-oriented disturbance. Given the problem statement and pros and cons of each candidate control algorithm, LQRI was selected to be implemented. The results show that the designed controller is capable of completing the entire mission except for the cases where the required input exceeds the limit. If we want to manage the input limitation more rigorously or the mission becomes more complex, other control techniques including nonlinear control would need to be considered.

References

[1] F. Sabatino, "Quadrotor control: Modeling, nonlinearcontrol design, and simulation," M.S. thesis, KTH Royal Institute of Technology, 2015.

[2] F. Ahmad, P. Kumar, A. Bhandari, and P. P. Patil, "Simulation of the Quadcopter Dynamics with LQR based Control," en, *Materials Today: Proceedings*, vol. 24, pp. 326–332, 2020, ISSN: 22147853. DOI: 10.1016/j.matpr.2020.04.282.

[3] M. Sharma, "Geometric Control of uncertain quadrotor with external disturbances," in *2022 IEEE 19th India Council International Conference (INDICON)*, Kochi, India: IEEE, Nov. 2022, pp. 1–5, ISBN: 978-1-66547-350-7. DOI: 10.1109/INDICON56171.2022.10039780.

[4] N. Wang, Q. Deng, G. Xie, and X. Pan, "Hybrid finite-time trajectory tracking control of a quadrotor," en, *ISA Transactions*, vol. 90, pp. 278–286, Jul. 2019, ISSN: 00190578. DOI: 10.1016/j.isatra.2018.12.042.

[5] T. L. Wong, R. R. Khan, and D. Lee, "Model linearization and H∞, controller design for a quadrotor unmanned air vehicle: Simulation study," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore: IEEE, Dec. 2014, pp. 1490–1495, ISBN: 978-1-4799-5199-4. DOI: 10.1109/ICARCV.2014.7064536.

[6] T. Madani and A. Benallegue, "Backstepping Control for a Quadrotor Helicopter," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China: IEEE, Oct. 2006, pp. 3255–3260, ISBN: 978-1-4244-0258-8 978-1-4244-0259-5. DOI: 10.1109/IROS.2006.282433.

[7] H. G. Malkapure and M. Chidambaram, "Comparison of Two Methods of Incorporating an Integral Action in Linear Quadratic Regulator," en, *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 55–61, 2014, ISSN: 14746670. DOI: 10.3182/20140313-3-IN-3024.00105.

[8] S. Kumar and L. Dewan, "Different Control Scheme for the Quadcopter: A Brief Tour," in *2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA)*, Kurukshetra, India: IEEE, Jun. 2020, pp. 1–6, ISBN: 978-1-72813-069-9. DOI: 10.1109/ICMICA48462.2020.9242886.