ME 547 Project

Max Howell

December 4, 2025

1 System Overview and Dynamics

1.1 System Overview

The system considered in this project is a quadrotor which is restricted to fly in the 2-D plane. An overview of this system is shown in Figure 1. As can be seen in the figure, the quadrotor is restricted to the x, y plane and allowed to rotate from the x-axis with body angle θ. The external forces which act on the quadrotor are gravity and an aerodynamic drag force F_D. The control inputs are the left and right thrust from the rotors T_l and T_r, respectively. The state and control vectors are then defined as

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} T_l \\ T_r \end{bmatrix}, \tag{1}$$

where the state dimension is $n = 6$ and the control dimension is $m = 2$.

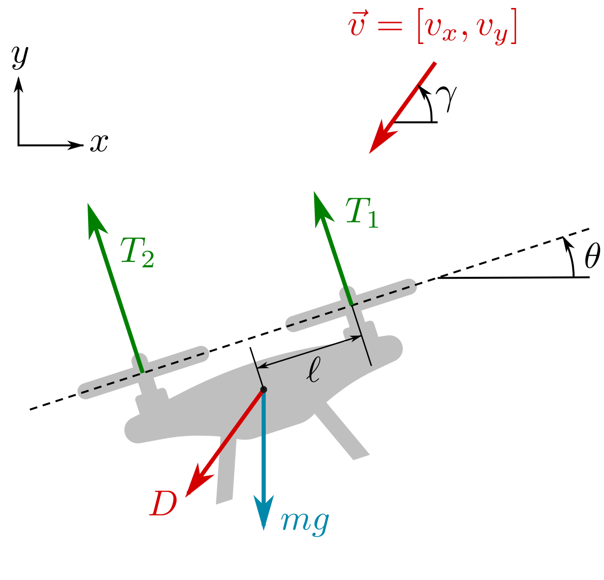

Figure 1: Planar Quadrotor Schematic

In order to derive the nonlinear equations of motion using a Newtonian approach, the total thrust force F_T and body drag force F_D as well as the x and y components are defined as

$$\begin{aligned} &F_{\mathrm{T}} = T_l + T_r, \qquad F_{\mathrm{Tx}} = -F_{\mathrm{T}}\sin(\theta), \qquad F_{\mathrm{Ty}} = F_{\mathrm{T}}\cos(\theta), \\ &F_{\mathrm{D}} = \beta(\dot{x}^2 + \dot{y}^2), \qquad F_{\mathrm{Dx}} = -F_{\mathrm{D}}\cos(\gamma), \qquad F_{\mathrm{Dy}} = -F_{\mathrm{D}}\sin(\gamma), \end{aligned} \tag{2}$$

where the constant β is a drag coefficient. The flight path angle is related to the velocity of the quadrotor by

$$cos(\gamma) = \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}}. \tag{3}$$

The physical parameters used for this example are shown in Table 1 and inspired by a drone presented in [1, Ch. 3.2].

Table 1: Parameters for the quadrotor problem.

Parameter	Value	Units
m	1.4	kg
I	0.0211	kg m^2
ℓ	0.159	m
β	0.1365	N/(m/s)^2

1.2 Derivation of Nonlinear Equations of Motion

Taking the summation of forces in the x and y directions and torques acting on the quadrotor about the central axis gives

$$\begin{aligned} \sum F_x &= m\ddot{x} = F_{\mathrm{Tx}} + F_{\mathrm{Dx}}, \\ \sum F_y &= m\ddot{y} = F_{\mathrm{Ty}} + F_{\mathrm{Dy}} - mg, \\ \sum \tau &= I\ddot{\theta} = T_r\ell - T_l\ell, \end{aligned} \tag{4}$$

where ℓ is the distance between the thrust acting points and the center of gravity and I is the moment of inertia. Using MATLAB's symbolic math toolbox to solve for the states and their derivatives gives

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \left(-(T_l + T_r)\sin\theta - \beta\dot{x}\sqrt{\dot{x}^2 + \dot{y}^2}\right)/m \\ \left((T_l + T_r)\cos\theta - \beta\dot{y}\sqrt{\dot{x}^2 + \dot{y}^2}\right)/m - g \\ (T_r - T_l)\ell/I \end{bmatrix}, \tag{5}$$

which defines the nonlinear quadrotor system.

1.3 Linearization

In order to design a linear control system, the nonlinear quadrotor system must be linearized about an appropiate operating point. For the states of this operating points we choose $\hat{\mathbf{x}} = \mathbf{0}$, where the quadrotor is hovering with zero body rotation. In order to make this operating point static, the reference control inputs must satisfy the nonlinear system $\dot{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \mathbf{0}$. The reference control input vector which satisfies this is

$$\hat{\mathbf{u}} = \begin{bmatrix} mg/2 \\ mg/2 \end{bmatrix}, \tag{6}$$

which completes the operating point $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$.

In order to linearize the quadrotor system the differential equations of motion can be written in implicit form as a function of the generalized coordinates, velocities, accelerations, and control inputs $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{u}) = \mathbf{0}$. The generalized coordinates are given as

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \tag{7}$$

and the generalized velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ are their respective time derivatives.

Expanding the implicit equations of motion with a Taylor series expansion gives

$$f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{u}) \approx \underbrace{\frac{\partial f}{\partial \ddot{\mathbf{q}}}\Big|_{\mathrm{OP}}}_{\mathbf{M}}(\ddot{\mathbf{q}} - \hat{\ddot{\mathbf{q}}}) + \underbrace{\frac{\partial f}{\partial \dot{\mathbf{q}}}\Big|_{\mathrm{OP}}}_{\mathbf{C}}(\dot{\mathbf{q}} - \hat{\dot{\mathbf{q}}}) + \underbrace{\frac{\partial f}{\partial \mathbf{q}}\Big|_{\mathrm{OP}}}_{\mathbf{K}}(\mathbf{q} - \hat{\mathbf{q}}) + \underbrace{\frac{\partial f}{\partial \mathbf{u}}\Big|_{\mathrm{OP}}}_{\mathbf{U}}(\mathbf{u} - \hat{\mathbf{u}}), \tag{8}$$

where the reference generalized coordinates and their derivatives are equal to zero. Taking the derivative matrices and forming the linear system by solving for the state derivative $\dot{\mathbf{x}} = [\dot{\mathbf{q}}, \ddot{\mathbf{q}}]^\top$ gives

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}^{-1}\mathbf{U} \end{bmatrix} (\mathbf{u} - \hat{\mathbf{u}}). \tag{9}$$

Again using MATLABs symbolic math toolbox, plugging the parameters from Table 1 into the system gives the full linearized state space system

$$\begin{aligned} \dot{\mathbf{x}} &= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -9.81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \mathbf{x} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.7143 & 0.7143 \\ -7.5355 & -7.5355 \end{bmatrix}}_{\mathbf{B}} (\mathbf{u} - \hat{\mathbf{u}}), \\ \mathbf{y} &= \begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x}, \end{aligned} \tag{10}$$

where the output vector is defined as $\mathbf{y} = \begin{bmatrix} x & y \end{bmatrix}^\top$ and the output dimension is $p = 2$.

1.4 Transfer Function Representation

In order to represent a multi-input multi-output (MIMO) system as a transfer function, multiple transfer functions must be used to map from each input to each output. For a system with p outputs and m inputs, the transfer functions can be represented in a (2×2) matrix which contains a transfer function as each entry. For the quadrotor system, this matrix is given as

$$\mathbf{Y}(s) = \begin{bmatrix} X(s) \\ Y(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} T_l(s) \\ T_r(s) \end{bmatrix}, \tag{11}$$

where each H_{ij} maps the j^{th} input to the i^{th} output in the Laplace domain.

Using the MATLAB functions ss() and tf(), each transfer function can be built. These transfer functions are given as

$$\begin{aligned} H_{11}(s) &= \frac{73.92s}{s^5}, \\ H_{12}(s) &= \frac{-73.92s}{s^5}, \\ H_{21}(s) &= \frac{0.7143s^3}{s^5}, \\ H_{22}(s) &= \frac{0.7143s^3}{s^5}. \end{aligned} \tag{12}$$

Note that all poles for these transfer functions are 0. This is consistent with the findings of the next section that the open loop system is Lyapunov stable.

2 Properties of the Linearized System

2.1 Stability, Controllability, and Observability

Before an observer and full state feedback control system can be designed, the properties of the linear system 10 must be assessed. To assess the controllability of the system, the eigenvalues of $\mathbf{A}$ can be given by MATLAB as

$$\begin{aligned} \lambda_1 &= 0 \\ \lambda_2 &= 0 \\ \lambda_3 &= 0 \\ \lambda_4 &= 0 \\ \lambda_5 &= 0 \\ \lambda_6 &= 0, \end{aligned} \tag{13}$$

which defines the state space system as being Lyapunov stable. Our later goal will be to design a full state feedback controller which stabilizes the system

In order to assess the controllability and observability of the system using the defined output vector $\mathbf{y}$, the controllability and observability test matrices can be formed in MATLAB as

$$
\begin{aligned}
\mathcal{P} &= \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} & \mathbf{A}^4\mathbf{B} & \mathbf{A}^5\mathbf{B} \end{bmatrix}, \\
\mathcal{O} &= \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \mathbf{CA}^4 \\ \mathbf{CA}^5 \end{bmatrix}.
\end{aligned} \tag{14}
$$

Taking the rank of both of these matrices in MATLAB shows that they are both full rank, which confirms that the state space system is both controllable and observable.

2.2 Response to Initial Conditions

In order to assess the response of the open-loop system system, both the nonlinear and linearized models were simulated in Simulink with control inputs of $mg/2$ (hovering). The initial conditions used are

$$
\mathbf{x}(0) = \begin{bmatrix} 0.1 \\ 0.1 \\ 0 \\ 0.1 \\ 0 \\ 0 \end{bmatrix}, \tag{15}
$$

which is equivalent to a hovering quadcopter at the x, y position $(0.1, 0.1)$ being perturbed in the x direction.

The resulting trajectory from this simulation is shown in Figure 2. As can be seen, the quadcopter continues to hover steadily at $y = 0.1$ while drifting in the positive x direction. Note that the linear system, in the time period of simulation, strays further from the initial condition than the nonlinear system. This is because the aerodynamic drag force which is present in the nonlinear system, is equal to zero at the operating point. This causes the linearized system to not have any drag effects which would reduce the velocity of the quadcopter. This is further shown in Figure 3, which shows the time evolution of all states for the simulation. It can be seen that the time derivative of the x position slowly decays to zero in the nonlinear case, but stays approximately equal to 0.1 in the linearized system.

3 Controller Design

3.1 Controller Requirements

The control task for this project is to design a full-state feedback controller that can track a reference trajectory $\mathbf{r}(t)$. The control law should take the form

$$
\mathbf{u}(t) - \hat{\mathbf{u}} = -\mathbf{K}\mathbf{x} + \mathbf{G}_1\mathbf{r} + \mathbf{G}_2\dot{\mathbf{r}}, \tag{16}
$$

where $\mathbf{K}$, $\mathbf{G}_1$, and $\mathbf{G}_2$ are linear controller gains.

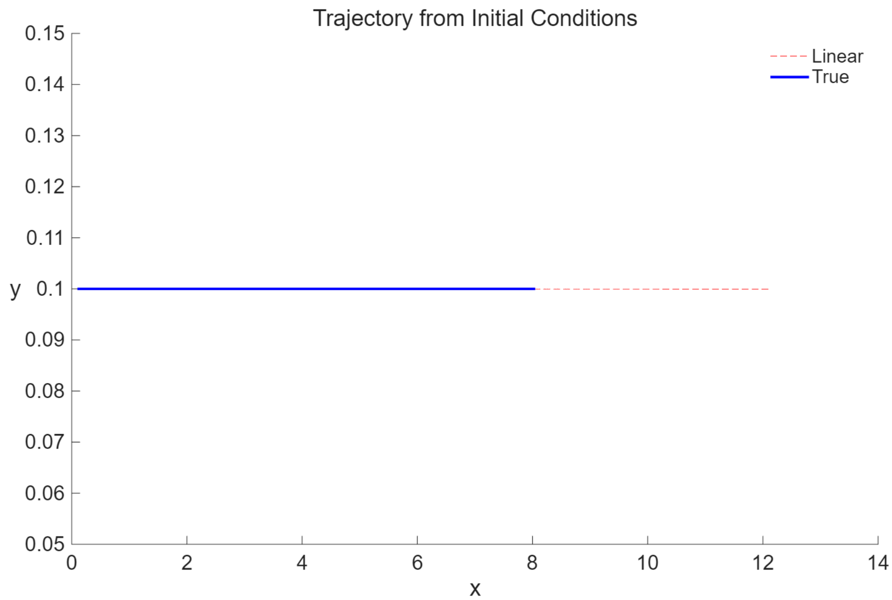

Figure 2: Simulated Trajectory response for the linear and nonlinear (True) systems due to initial conditions.

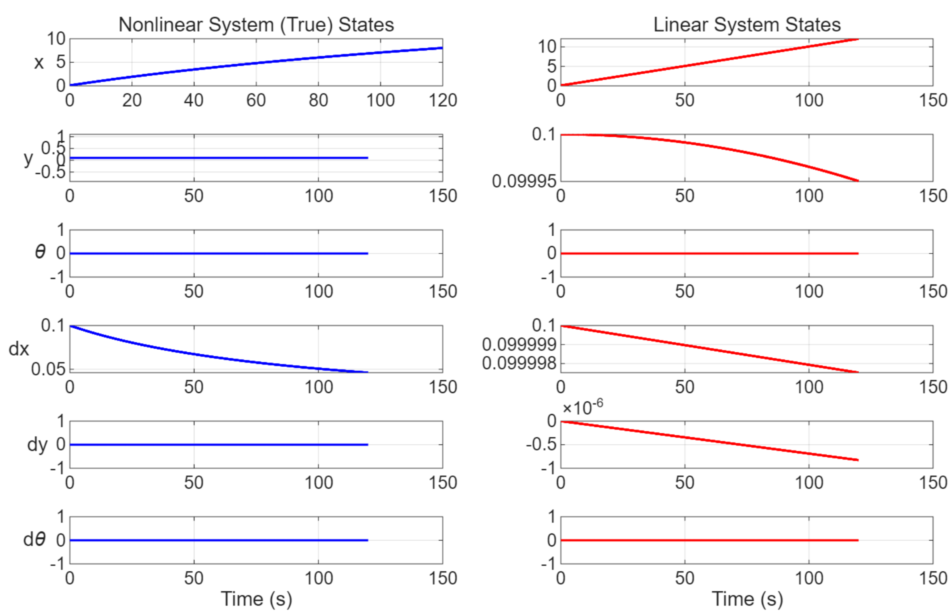

Figure 3: Time evolution of all states for the linear and nonlinear responses to initial conditions.

The reference signal is given as a circle in the (x, y) plane with period $T_r = 30$s, frequency $\omega = 2\pi/T_r$ and radius of $r = 0.2$m. The equations for the reference signal and its derivatives can be given as

$$\begin{aligned}
\mathbf{r}_x(t) &= r\text{cos}(\omega t), \\
\mathbf{r}_y(t) &= r\text{sin}(\omega t), \\
\dot{\mathbf{r}}_x(t) &= -r\omega\text{sin}(\omega t), \\
\dot{\mathbf{r}}_y(t) &= r\omega\text{cos}(\omega t).
\end{aligned} \tag{17}$$

The controlled quadrotor system should track the reference trajectory as closely as possible while using less than 8 Newtons of thrust per rotor, which is the maximum control output of the system. The control input should also not be less than zero at any point in time as this violates the physics of the system.

3.2 Controller Design

The chosen full state feedback controller for this project is a linear quadratic regulator (LQR) controller. This controller minimizes the LQR cost

$$J = \frac{1}{2}\int_0^\infty \mathbf{x}(t)^\top \mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^\top \mathbf{R}\mathbf{u}(t)dt, \tag{18}$$

where $\mathbf{Q} \succeq \mathbf{0}$ and $\mathbf{R} \succ \mathbf{0}$ are state and control weighting matrices, respectively. In order to design the matrices $\mathbf{Q}$ and $\mathbf{R}$ an iterative process was taken to design the diagonal entries to the matrices by selecting entries (starting and $\mathbf{Q} = \mathbf{I}_n$, $\mathbf{R} = \mathbf{I}_m$) and simulating the closed loop system to evaluate the tracking performance and control input magnitude. The final design for the state and control weighting matrices that best tracked the reference trajectory without exceeding 8 Newtons of thrust per rotor are $\mathbf{Q} = \mathbf{I}_n$ and $\mathbf{R} = 0.2\mathbf{I}_m$. Using MATLAB, the feedback gain matrix $\mathbf{K}$ and controller poles $\mathbf{p}_c$ can be given as $[\mathbf{K}, \mathbf{p}_c] = \text{lqr}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R})$.

The tracking gains $\mathbf{G}_1$ and $\mathbf{G}_2$ can be found from

$$\begin{aligned}
\mathbf{G}_1 &= [\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}]^{-1}, \\
\mathbf{G}_2 &= \frac{\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-2}\mathbf{B}}{(\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B})^2},
\end{aligned} \tag{19}$$

which completes the full-state feedback tracking controller of form 16.

3.3 Simulation with Full-State Feedback Controller

In order to iteratively design the full state feedback controller independently of the observer and assess the controllers effectiveness, the closed loop system was simulated assuming all states could be sensed. This simulation implemented the control law in eq. (16) to track the reference signal. The quad-rotor's initial states was hovering perfectly with zero body rotation at $(x.y) = (1, 0.4)$.

The resulting trajectory of this simulation for both the linear and nonlinear systems are shown in Figure 4. As can be seen, the controller design effectively tracks reference signal in the (x, y) plane for both the linear and nonlinear systems. The small amount of error between the quadrotor and reference trajectory is caused by the constantly changing reference signal. Using the above formulation for $\mathbf{G}_1$ and $\mathbf{G}_2$, the quadrotor is only guaranteed to track the steady state values of the reference signals and its first derivative. While this error could be eliminated by considering more derivatives or increasing the period of the reference signal, this would require a larger control input which is not desired.

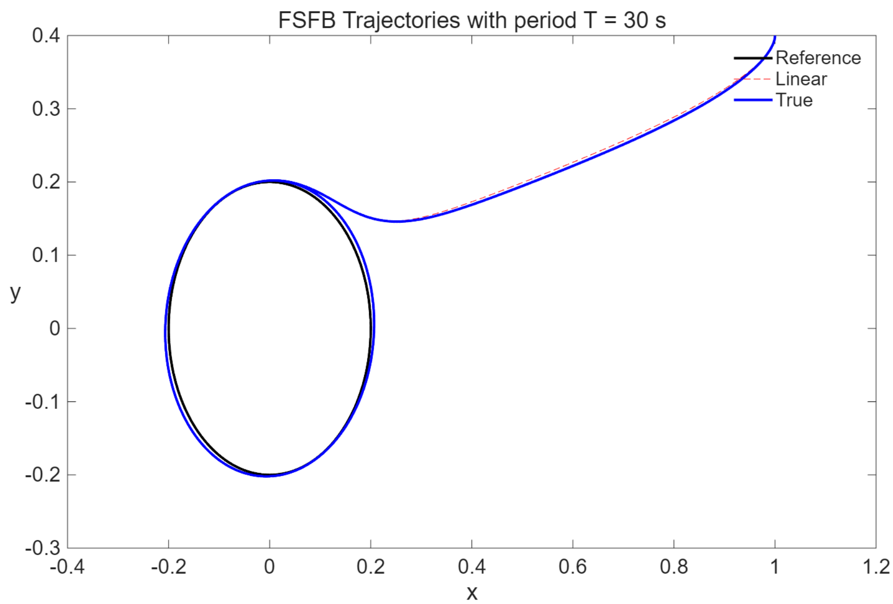

Figure 4: The full state feedback controller tracks the reference trajectory with only a small error for both the linear and nonlinear (True) systems.

Shown in Figures 5 and 6 are the state time history and the control time history for the linear and nonlinear simulations. As can be seen, the maximum control input used is approximately 7.6 Newtons, which is less than the maximum control input of 8 Newtons. It can also be seen that the state time history converges to a steady orbit while tracking the reference trajectory over multiple periods. This simulation confirms that our full state feedback controller is designed well, and we can now implement an observer for the more realistic case where the full state $\mathbf{x}$ is not known.

4 Observer Design

In the case that it is only possible to measure the x and y location of the quadrotor position, it is desired to implement a Luenberger observer to reconstruct the full state of the system and process any sensor noise in the output vector $\mathbf{y}$ and as well as any uncertainty in the model. In order to design this Luenberger observer, the Kalman-Bucy filter approach is used. This approach requires defining covariance matrices $\mathbf{Q}_o = \mathbf{E}(\mathbf{w}\mathbf{w}^\top)$ and $\mathbf{R}_0 = \mathbf{E}(\mathbf{v}\mathbf{v}^\top)$, where $\mathbf{w}$ is white noise in the system model and $\mathbf{v}$ is white noise in the sensors.

Since the process noise in the model is minimal, the weighting matrix $\mathbf{Q}_0$ was chosen as $\mathbf{Q}_0 = 0.01\mathbf{I}_n$. In simulation of the system, a white noise with power 1×10^{-6} was added to the output matrix $\mathbf{y}$. With this noise, the matrix $\mathbf{R}_0$ was chosen as $\mathbf{R}_0 = 0.01\mathbf{I}_p$.

The Luenberger observer model is then given as

$$\begin{aligned} \dot{\bar{\mathbf{x}}} &= \mathbf{A}\bar{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \bar{\mathbf{y}}), \\ \bar{\mathbf{y}} &= \mathbf{C}\bar{\mathbf{x}}, \end{aligned} \tag{20}$$

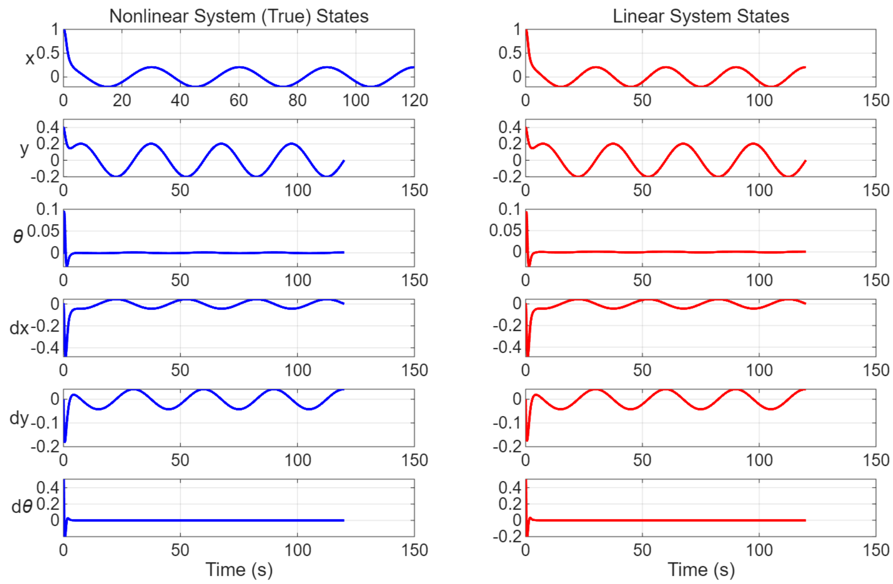

Figure 5: The state time evolution of the linear and nonlinear systems with a full state feedback controller.

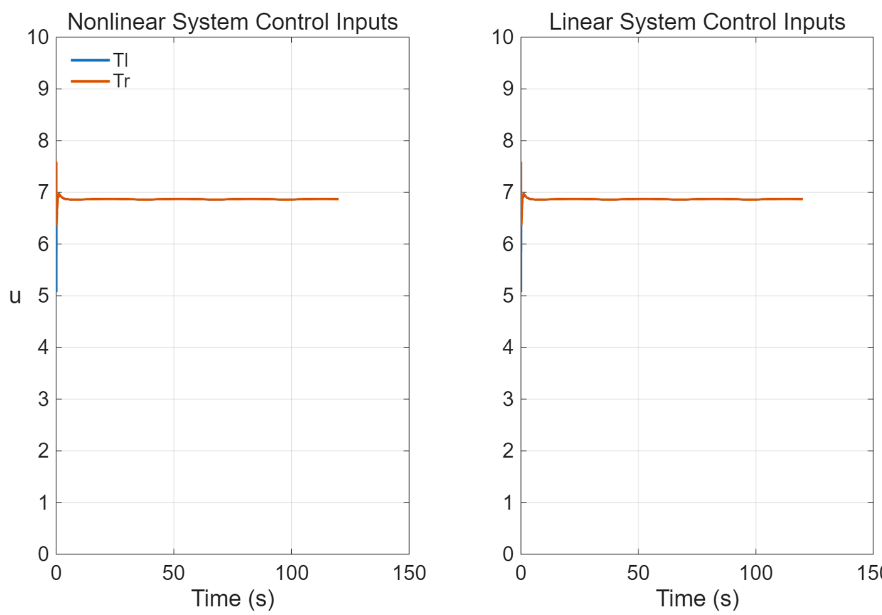

Figure 6: The control inputs over time for the full-state feedback controlled systems.

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are the reconstructed state and output estimated by the observer. Using the Kalman-Bucy filter approach, the optimal observer gain $\mathbf{L}$ as well as the observer poles can be given in MATLAB as $[\mathbf{L}^\top, \mathbf{p}_o] = \text{lqr}(\mathbf{A}^\top, \mathbf{C}^\top, \mathbf{Q}_0, \mathbf{R}_o)$. Both the observer and controller poles for the final control system design are shown in Figure 7. As can be seen, the all but one of the controller poles are at approximately -1 to -2.5 on the real axis with one very fast pole at approximately -24 on the real axis. The observer poles are placed with two complex conjugate pairs at approximately -1 on the real axis and one repeated pole at around -3 on

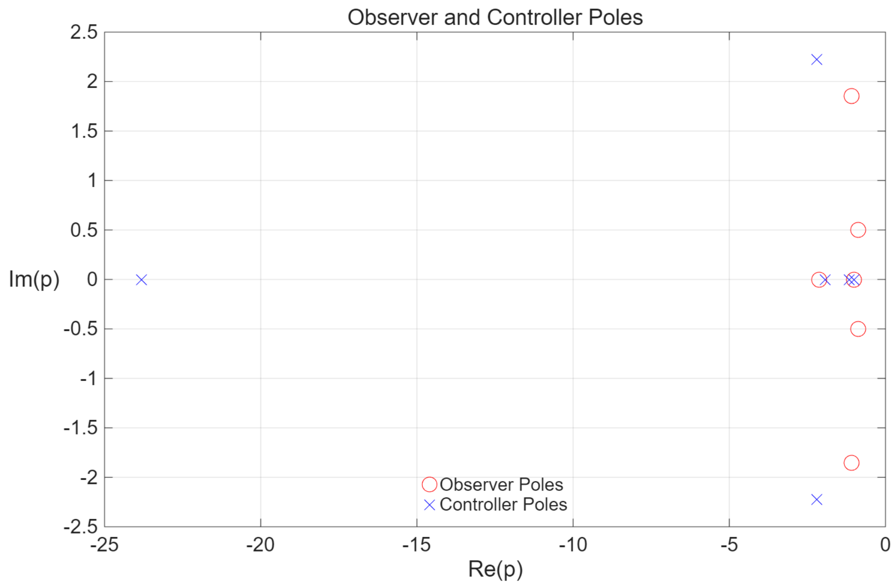

Figure 7: Complex poles for the final observer and controller design.

the real axis.

Using the combined observer and full state feedback to control the quadrotor, the updated control law can be given as

$$\mathbf{u}(t) - \hat{\mathbf{u}} = -\mathbf{K}\bar{\mathbf{x}} + \mathbf{G}_1\mathbf{r} + \mathbf{G}_2\dot{\mathbf{r}}, \tag{21}$$

where the reconstructed state from the observer is used rather than the true (unknown) state of the system.

5 Closed Loop Simulation

To assess the entire closed loop system with the observer and full state feedback controller, the system was simulated with Simulink. The initial conditions for this simulation are the same as those in section 3.3, where the quadcopter is initially at a steady hover with zero body rotation at $(x, y) = (1, 0.4)$.

The resulting trajectory for both the nonlinear and linear systems are shown in Figure 8. As can be seen, this simulation has a few key differences from the simulation with only the full-state feedback controller implemented. First, the white noise applied to the sensors is very apparent in the simulation as the tracking is not smooth. Also, there is a noticeable difference in the trajectories of the linear and nonlinear systems, which was not present in the previous simulation. This is most likely caused by the fact that the observer is balancing the error in the sensors and the error between the linear and nonlinear systems. While the nonlinear simulation uses the true nonlinear equations of motion, the observer still simulates the linear system. This, combined with the added noise, potentially causes a incorrect reconstructed state leading to differences between the linear and nonlinear systems.

The state time history for this simulation is shown in Figure 9. The small amount of white noise added to the sensors can be seen in the resulting state evolution. While the states do converge to a orbit while tracking the reference signal, it is not perfectly smooth. A small

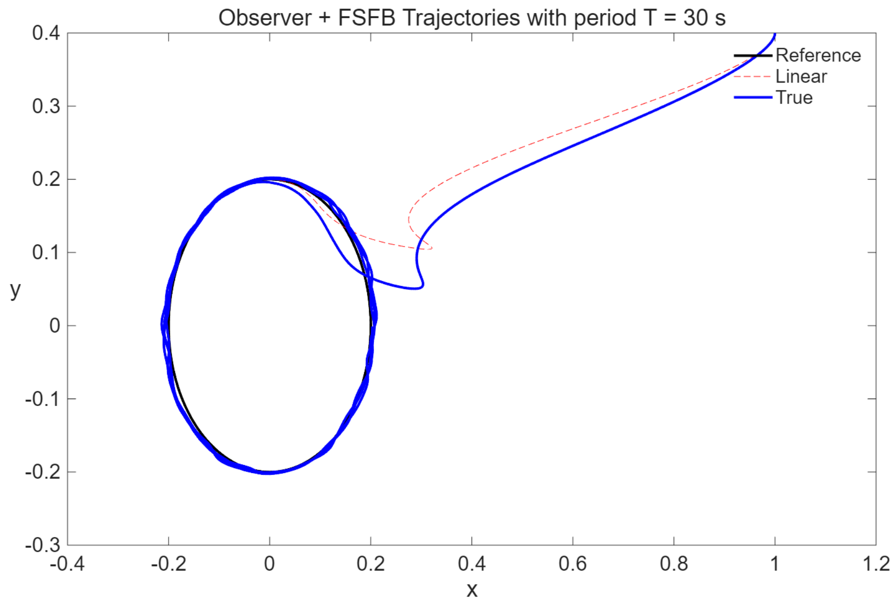

Figure 8: Trajectories of the linear and nonlinear (True) systems when simulated using both the full-state feedback controller and observer. Both systems are subjected to the same white noise in the sensor.

amount of noise can be seen in the states, specifically in the velocities of the positions and body rotation. This noise arises from the error between the reconstructed observer state and the true (unknown) state vector. A plot of the L^2 norm of the error between these two vectors is shown in Figure 10. As can be seen, even though the error $||\mathbf{x}(t) - \bar{\mathbf{x}}(t)||_{L^2}$ is very small (about 0.1), the white noise is very apparent. This white noise leads to the noisy states as well as the noisy trajectory tracking.

Shown in Figure 11 is the control effort used in the linear and nonlinear systems. The control effort for this case is very similar to the full-state feedback only system. This is because, even with the observer and the added noise, the linear control gains are still the exact same as before. Also, the white noise and observer does not significantly change the state $\bar{\mathbf{x}}$ which is used to determine the control input (shown in Figure 10). Therefore, the control effort still does not exceed the maximum value of 8 Newtons.

6 Conclusion

In this project, the planar quadcopter system with body drag was controlled with a full-state feedback (LQR) controller to track a reference signal. A Kalman filter was also used to observe the quadcopter system in the situation where the x and y position were sensed with imperfect (noisy) sensors. This process involved first deriving the full nonlinear equations of motion with a Newtonian approach, then linearizing about an appropriate operating point. Once the linear state space model was constructed, the systems properties were assessed and it was found that the system was Lyapunov stable as well as both observable and controllable.

A full-state feedback was then designed for the system using the LQR approach which stabilized the system by moving all the poles into the left half plane. A tracking control law was also implemented to make the quadrotor track a circular reference trajectory, taking

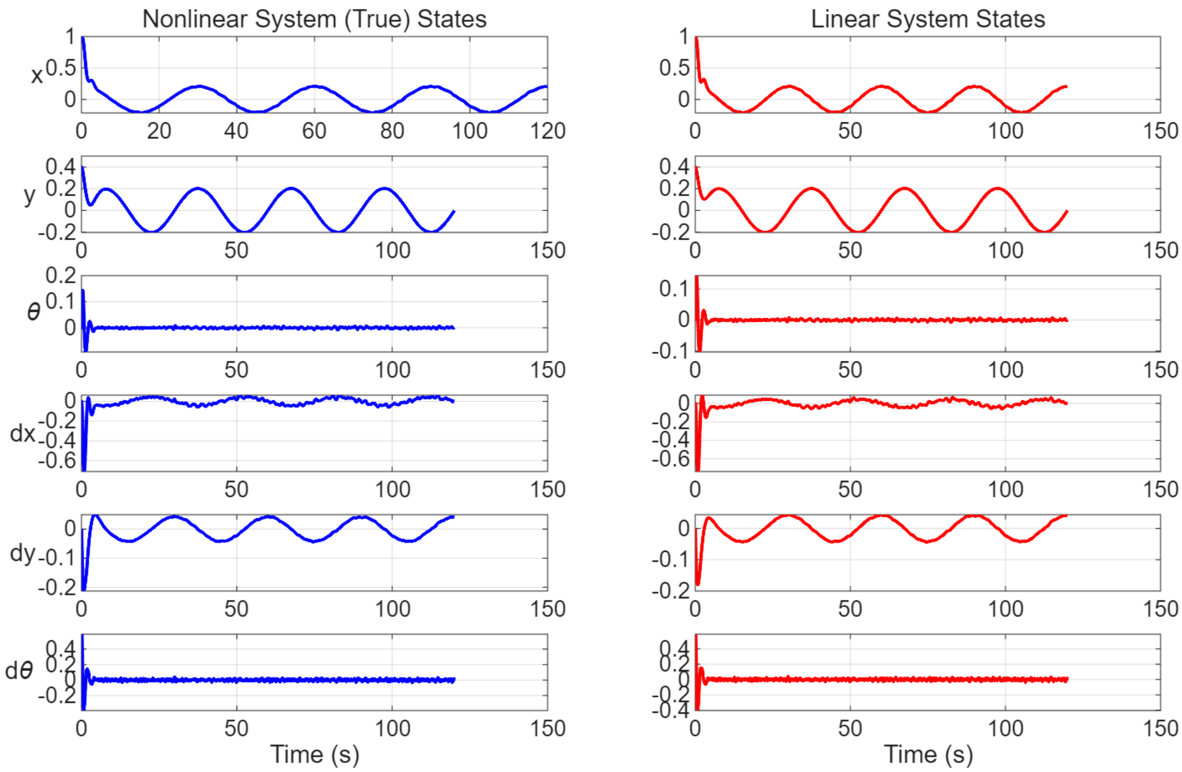

Figure 9: State time history for the simulations with the observer and full-state feedback controller.

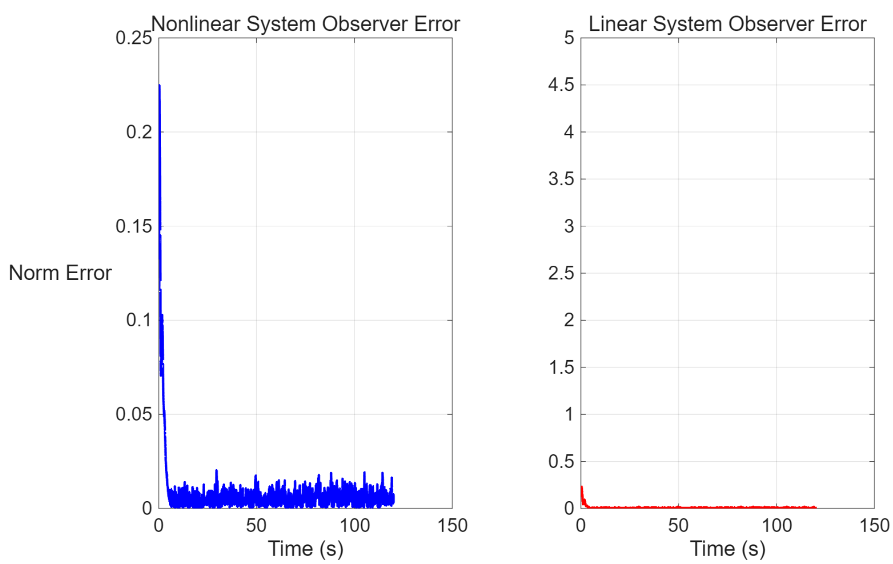

Figure 10: L^2 norm of the error between the reconstructed observer states $\bar{\mathbf{x}}$ and the true state $\mathbf{x}$. While the error is small, the white noise is very apparent.

into account both the trajectory as well as its first time derivative in the control law. It was

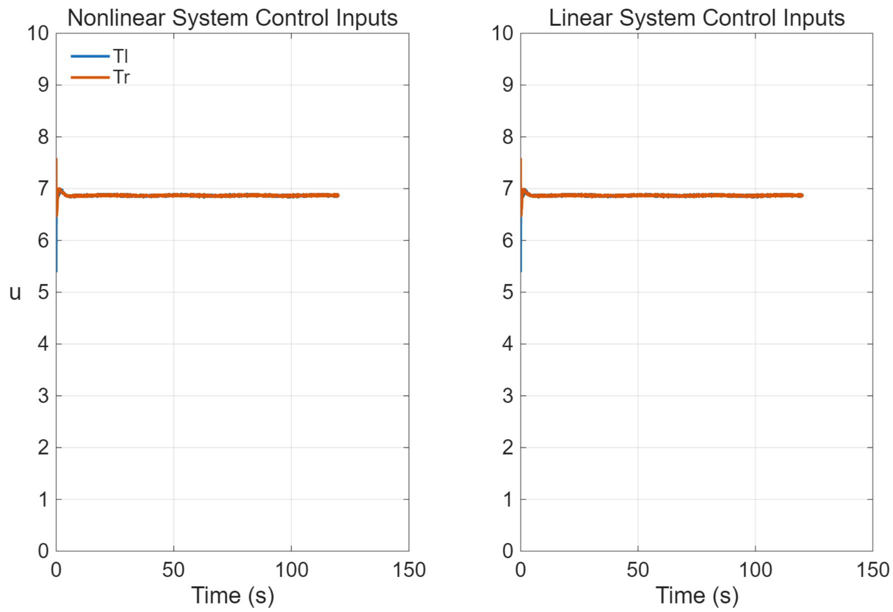

Figure 11: Control effort for the linear and nonlinear simulations of the full-state feedback system with a Kalman filter.

then shown that this full-state feedback tracking controller adequately tracked the reference trajectory without exceeding the thrust limitations of the physical quadcopter system.

In order to make a more realistic system where only the x and y positions of the quadrotor could be sensed with noisy sensors, a Luenberger observer was created using the Kalman filter approach. This observer was integrated with the full-state feedback controller in Simulink and it was shown that even with the noisy sensors, the observer was able to reconstruct the full state with minimal error. The fully implemented LQR controller with reference signal tracking and a Kalman filter observer was able to stabile the quadcopter system and track the circular reference signal in the (x, y) plane.

References

[1] Quan, Q., Dai, X., and Wang, S. (2020). *Multicopter Design and Control Practice: A Series Experiments based on MATLAB and Pixhawk*. Springer Singapore.