EECE660/MECH653 – Final Project Report

Quanser 3 DOF Hover

Final project proposal for course MECH 653 (System Analysis and Design) at the Department of Mechanical Engineering, American University of Beirut, Lebanon

Fall 2019-2020

Final Report

Group Members:

Daoud, Yves Georgy

Haddad, Ramzi

Maalouf, Guy

Dr. Naseem Daher

Date: 11/12/2019

Contents

Figures

Abstract

The aim of this project is to model a Quadrotor, evaluate different attitude controllers, and design two controllers to be tested on the Quanser 3-DOF Hover testbed. Neglecting the effects of drag, gyroscopic effects, bias voltage, saturation limit, disturbances, and nonlinearities can have serious implications on the system's actual performance. Thus, controllers designed for a linearized simplistic system have to be experimentally evaluated. The model and different controllers are tested first on simulations to compare their performance, and then deployed on the Quanser testbed to demonstrate their effectiveness and robustness.

Introduction and Literature Review

With the boom of unmanned aerial vehicles (UAVs), more work is being done going in two directions. On one hand, researchers and industries are using the UAVs as black boxes to accomplish various tasks and applications such as monitoring vegetation conditions in precision agriculture [1], detecting landmines autonomously using onboard radars [2], performing search and rescue missions in inaccessible and harsh locations [3], 3D surveying of monuments and archaeological sites [4], and delivering packages in a fast, structured and efficient way [5, 6]. The demand for UAVs keeps on increasing in popularity and its industry is growing fast. Additionally, more and more applications are surfacing that require extreme precision and high level of control to achieve these exclusive maneuverability advantages. Unfortunately, these present a complexity and instability in the UAV's dynamics, making it hard to control.

This is where the other researchers step in to develop more robust controllers that can handle various conditions and variabilities. The field of quadcopter control is an emerging area in linear and nonlinear control study among researchers. Of the many contributions on modeling and control of quadrotors, some of those can be summarized as follows: Salih et al. [7] developed a realistic model of a quadcopter and presented a PID control algorithm to control its attitude and altitude. Anastasia et al. [8] controlled the altitude via a normal PID and a fuzzy logic based PID controller, which can be easily extended to an attitude controller. Abdelkhalek et al. [9] compared two different designs used for attitude control, a classical PD and an angular acceleration plus PD controller. Kaan et al. [10] developed an LQR based position and attitude control algorithm, applied on a nonlinear dynamic model of a quadcopter, with testing in simulation. Stevanovic et al. [11] developed a robust tracking controller to compensate for the external disturbances caused by wind, considering feedback linearization control, sliding mode control, and sliding mode derivative filter control. Tamami et al. [12] used a proportional derivative active force controller (PDAFC), where the PD part is used to stabilize the quad, and the AFC is used to reject disturbance uncertainty (e.g. wind). Tamami et al. concluded that the PDAFC is more robust than the normal PD controller in terms of disturbance rejection.

Suresh et al. [13] compared three different common techniques to control a quadcopter that they modeled using the Newton Euler approach, based on fall time, percentage undershoot and computation time. They started with the simplest one and most widely used controller, the PID and then followed by the LQR and the feedback linearization to finally come up with the following conclusion. The PID performed the worse being the slowest with very large undershoots. The LQR on the other hand, was the fastest and had the least undershoots with very sharp responses, but due to a small delay in response was not chosen as the best control technique, instead this title was given to the FBL (Feedback linearization), having the best balance between speed, percentage undershoot, computation time. Unfortunately this paper was only tested through simulation and no real life experiment was made, therefore an implementation of such techniques in real life might reflect more accurately how good those methods are and how robust they are when affected by external disturbances like ground effect, external wind or external forces that might occur due to small imperfections in manufacturing leading to an unbalanced quadcopter with an offset on the center of mass.

Dynamic Model Overview

The Quanser 3 DOF Hover, like any other quadcopter, can rotate in 3 directions: about its pitch axis, about its roll axis, and about its pitch axis. The system will be divided into each angle rotation individually to perform analysis on the dynamics of the system.

Yaw Axis Dynamics

The yaw angle θ_y is assumed to be positive in the counterclockwise direction when viewing the quadcopter from the yaw axis view. The front and back propellers are chosen to be moving in the clockwise direction. Consequently, to counteract the motion of the front and back propellers and keep the quadcopter having a yaw angle , θ_y, of 0°, the left and right propellers must be rotating in the counterclockwise direction.

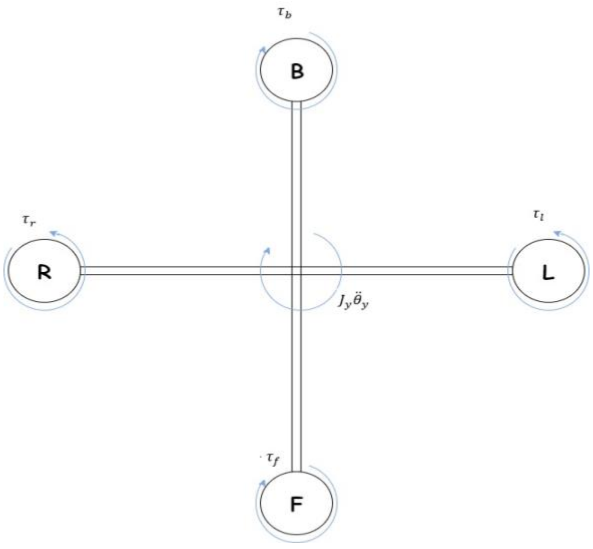


Figure 1: Free Body Diagram of Quadcopter for Yaw Axis

Using Newton's 2nd Law of motion for rotation and considering the coupling effect on the pitch and roll axes,

$\sum M_O = 0$	*Equation 1*
$J_y\ddot{\theta}_y + \tau_f + \tau_b = \tau_l + \tau_r + (J_r - J_p)\,\dot{\theta}_p\dot{\theta}_r$	*Equation 1*
$J_y\ddot{\theta}_y = (J_r - J_p)\,\dot{\theta}_p\dot{\theta}_r + \tau_l + \tau_r - \tau_f - \tau_b$	*Equation 1*

Knowing that the forces can be represented in terms of the input voltages by the relation $\tau = K_t V$,

$J_y\ddot{\theta}_y = (J_r - J_p)\,\dot{\theta}_p\dot{\theta}_r + K_t V_l + K_t V_r - K_t V_f - K_t V_b$	*Equation 1*

Pitch Axis Dynamics

Looking at the quadcopter from the pitch axis, the pitch angle θ_p is assumed to be positive in the counterclockwise direction.

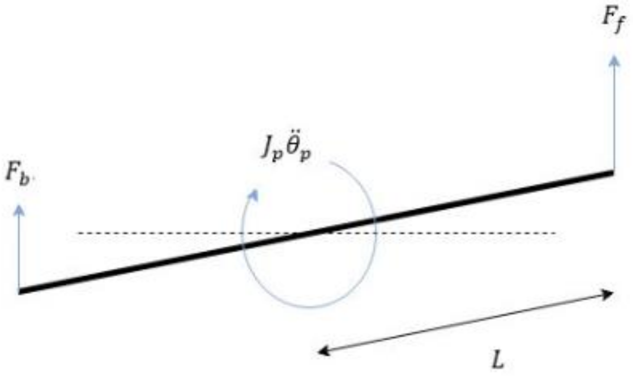


Figure 2: Free Body Diagram of Quadcopter for the Pitch Axis

Using Newton's 2nd Law of motion for rotation and taking coupling into consideration,

$\sum M_O = 0$	*Equation 2*
$J_p\ddot{\theta}_p + F_b L = F_f L + (J_y - J_r)\,\dot{\theta}_y\dot{\theta}_r$	*Equation 2*

Knowing that the forces can be represented in terms of the input voltages by the relation $F = K_f V$,

$J_p\ddot{\theta}_p = (J_y - J_r)\,\dot{\theta}_y\dot{\theta}_r + F_f L - F_b L = (J_y - J_r)\,\dot{\theta}_y\dot{\theta}_r + K_f V_f L - K_f V_b L$	*Equation 2*

Roll Axis Dynamics

Looking at the quadcopter from the roll axis, the roll angle θ_r is assumed to be positive in the counterclockwise direction.

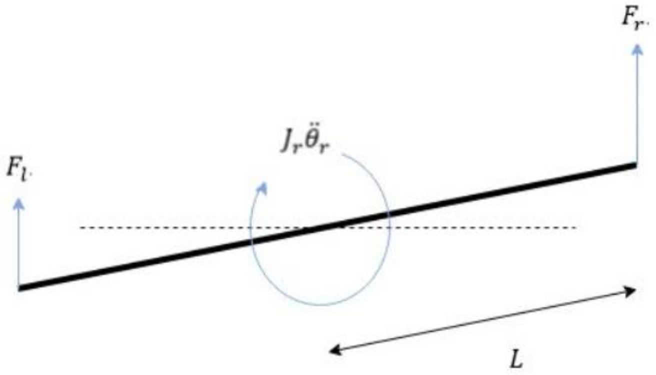


Figure 3: Free Body Diagram of Quadcopter for Roll Axis

Like the pitch dynamics,

$\sum M_O = 0$	*Equation 3*
$J_r\ddot{\theta}_r + F_l L = (J_p - J_y)\dot{\theta}_p\dot{\theta}_y + F_r L$	*Equation 3*

Knowing that the forces can be represented in terms of the input voltages by the relation $F = K_f V$,

$J_r\ddot{\theta}_r = (J_p - J_y)\dot{\theta}_p\dot{\theta}_y + F_r L - F_l L = (J_p - J_y)\dot{\theta}_p\dot{\theta}_y + K_f V_r L - K_f V_l L$	*Equation 3*

The state space representation of the quadcopter system is defined by:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Since the quadcopter has 3 degrees of freedom, there are 6 state variables. The state variables are chosen to be the pitch, raw, yaw angles and their respective rates.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \theta_y \\ \theta_p \\ \theta_r \\ \dot{\theta}_y \\ \dot{\theta}_p \\ \dot{\theta}_r \end{bmatrix}$$

The quadcopter has 4 propellers, so there are 4 inputs which are the voltages fed into each propeller.

$$u = \begin{bmatrix} V_f \\ V_b \\ V_r \\ V_l \end{bmatrix}$$

The outputs of the system are the 3 angles that define the system.

$$y = \begin{bmatrix} \theta_y \\ \theta_p \\ \theta_r \end{bmatrix}$$

The system will be linearized for simplicity by neglecting the coupling terms introduced.

To find the A and B matrices, each state variable is derived in terms of the states as follows,

$\dot{x}_1 = \dot{\theta}_y = x_4$	*Equation 4*
$\dot{x}_2 = \dot{\theta}_p = x_5$	*Equation 5*
$\dot{x}_3 = \dot{\theta}_r = x_6$	*Equation 6*
$\dot{x}_4 = \ddot{\theta}_y = -\frac{K_t}{J_y}V_f - \frac{K_t}{J_y}V_b + \frac{K_t}{J_y}V_l + \frac{K_t}{J_y}V_r$	*Equation 7*
$\dot{x}_5 = \ddot{\theta}_p = \frac{K_f L}{J_p}V_f - \frac{K_f L}{J_p}V_b$	*Equation 8*
$\dot{x}_6 = \ddot{\theta}_r = \frac{K_f L}{J_r}V_r - \frac{K_f L}{J_r}V_l$	*Equation 9*

Therefore, using equations 4-9 to represent the system in state space,

$$\dot{x} = Ax + Bu$$

Hence,

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{K_f L}{J_p} & -\frac{K_f L}{J_p} & 0 & 0 \\ 0 & 0 & \frac{K_f L}{J_r} & -\frac{K_f L}{J_r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.0222 & -0.0222 & 0.0222 & 0.0222 \\ 0.322 & -0.322 & 0 & 0 \\ 0 & 0 & 0.322 & -0.322 \end{bmatrix}$$

$$y = Cx + Du$$

Hence,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

System Identification

Thrust Characteristics

In order to test the thrust characteristics of the DC motors, a rope was connected to a very heavy mass placed on a balance below the propeller. Then a voltage was induced at the tested motor's terminals (colored in red) and increased by steps of 1 V while inducing simultaneously half this voltage at the terminals of the two adjacent motors (colored in green) in order to counteract the yaw effect produced by the motor.

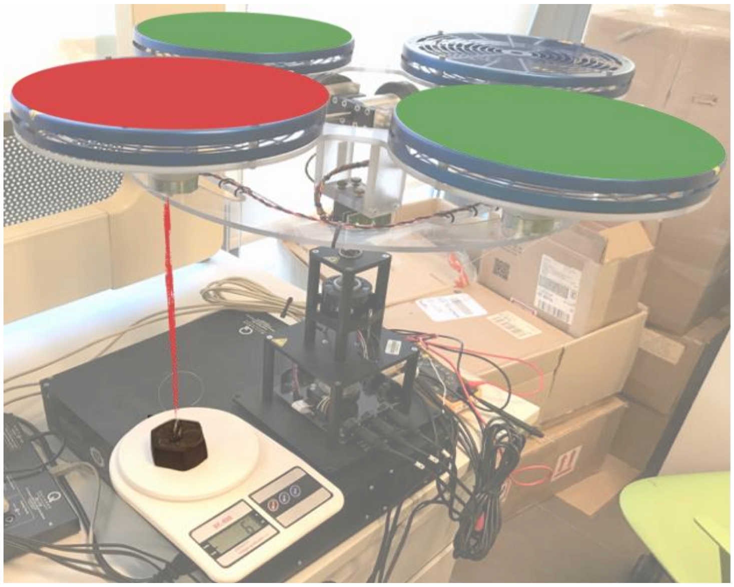


Figure 4: Experimental setup to find the thrust characteristics of the DC motors

After recording the different thrust values, a plot was generated. We can see that the motor has a dead-zone of approximately 2 V and then acts as a first-order system, thus thrust can be modeled as $K_f\ V$, with $K_f = 0.090217\ \mathrm{N/V}$ in comparison to the K_f given by Quanser which was $0.1188\ \mathrm{N/V}$.

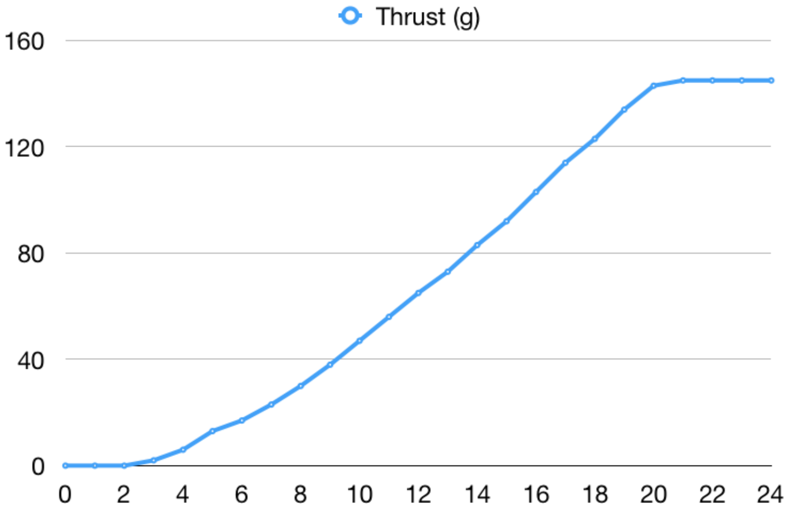


Figure 5: Thrust Characteristic Curve

Figure 6 shows a comparison of the response of an LQR controller with the two values of K_f, and it can be seen that the response with the custom K_f had a much smaller overshoot and slightly less steady-state error.

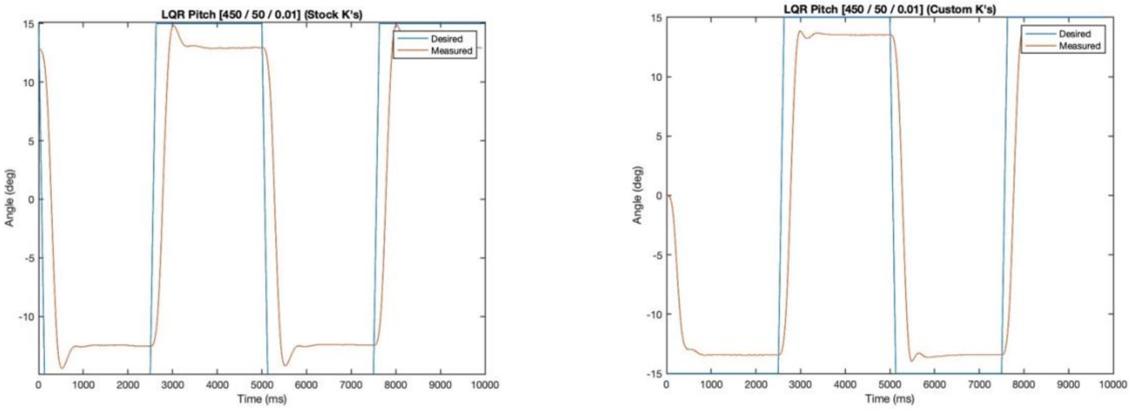


Figure 6: Response of LQR controller with Stock K_f (left) versus Custom K_f (Right)

Torque Characteristics

In order to test the torque characteristics of the DC motors, a rope was connected perpendicularly to a very heavy mass placed on a balance, as seen in figure 7. A voltage was induced at opposite motors' terminals (colored in red) and increased by steps of 1 V.

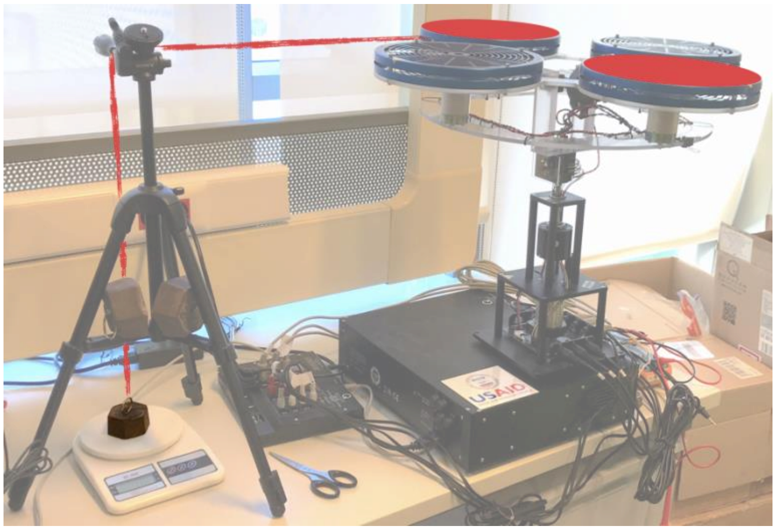


Figure 7: Experimental setup to find the torque characteristics of the DC motors

After recording the different torque values, a plot was generated, and as seen in figure 8, the motor has a dead-zone of approximately 5V and then acts as a first-order system, thus torque can be modeled as $K_t V$, with $K_t = 0.0024439 \text{ N m/V}$

in comparison to the K_t given by Quanser that was 0.0036 N m/V. It is important to note that the Value of K_t is half the value of the slope of the graph recorded since to generate the torque two motors were excited simultaneously.

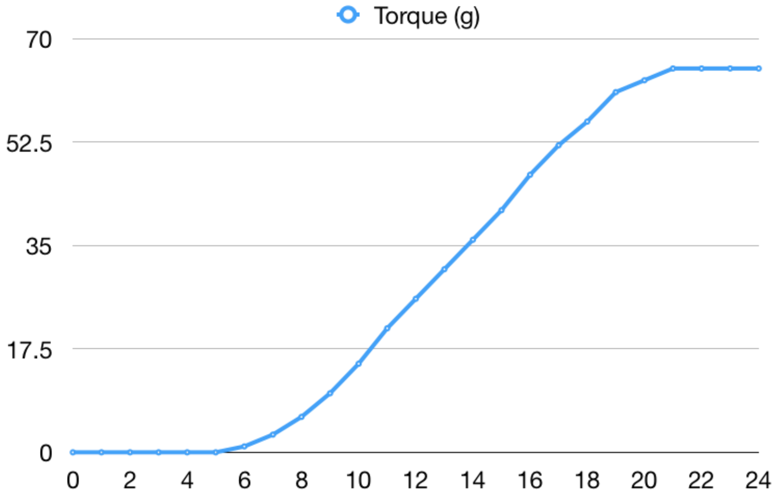


Figure 8: Torque Characteristic Curve

Figure 9 illustrates a comparison of the response of an LQR controller with the two values of K_t, and it can be seen that there is a very slight increase in overshoot and steady-state error while using the K_t provided by Quanser.

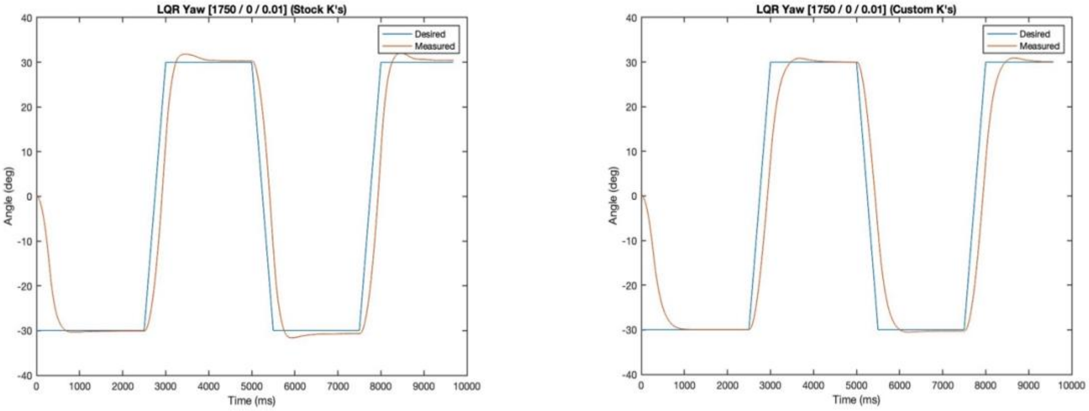


Figure 9: Response of LQR controller with stock K_t (left) versus custom K_t (right)

System Analysis

Phase Portrait Analysis

Phase portraits help in understanding the stability of the system. However, to reduce dimensions of the plot, it is required to decouple the system for each degree of freedom.

Yaw System

$$\begin{bmatrix}\dot{\theta}_y\\ \ddot{\theta}_y\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\theta_y\\ \dot{\theta}_y\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ -\dfrac{K_t}{J_y} & -\dfrac{K_t}{J_y} & \dfrac{K_t}{J_y} & \dfrac{K_t}{J_y}\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

$$\begin{bmatrix}\dot{\theta}_y\\ \ddot{\theta}_y\end{bmatrix}=\underbrace{\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}}_{}\begin{bmatrix}\theta_y\\ \dot{\theta}_y\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ -0.0222 & -0.0222 & 0.0222 & 0.0222\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

Pitch System

$$\begin{bmatrix}\dot{\theta}_p\\ \ddot{\theta}_p\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\theta_p\\ \dot{\theta}_p\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ \dfrac{K_f L}{J_p} & -\dfrac{K_f L}{J_p} & 0 & 0\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

$$\begin{bmatrix}\dot{\theta}_p\\ \ddot{\theta}_p\end{bmatrix}=\underbrace{\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}}_{}\begin{bmatrix}\theta_p\\ \dot{\theta}_p\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ 0.322 & -0.322 & 0 & 0\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

Roll System

$$\begin{bmatrix}\dot{\theta}_r\\ \ddot{\theta}_r\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\theta_r\\ \dot{\theta}_r\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & \dfrac{K_fL}{J_r} & -\dfrac{K_fL}{J_r}\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

$$\begin{bmatrix}\dot{\theta}_r\\ \ddot{\theta}_r\end{bmatrix}=\underbrace{\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}}_{}\begin{bmatrix}\theta_r\\ \dot{\theta}_r\end{bmatrix}+\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0.322 & -0.322\end{bmatrix}\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$$

Since all three systems have the same A matrix, they generate the same phase portrait. The plot is generated using MATLAB’s quiver command.

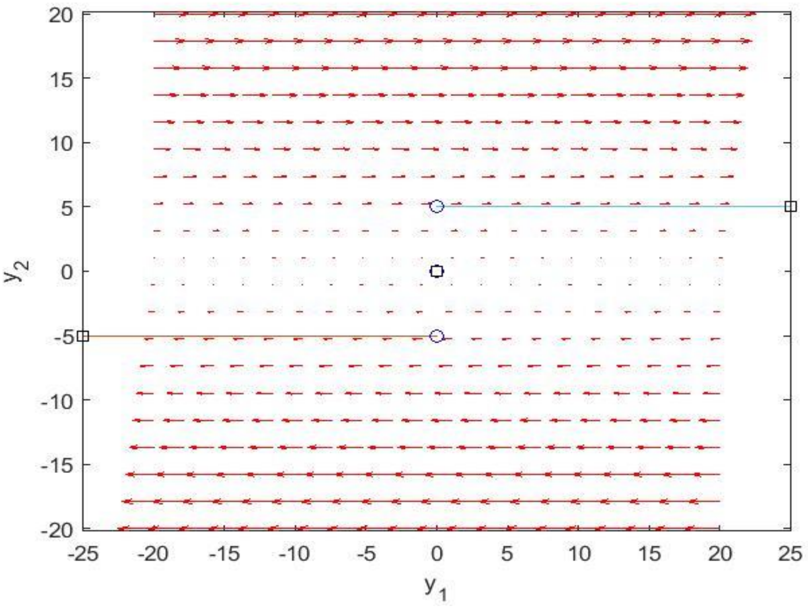


Figure 10: Phase portrait of all 3 decouples systems

As shown in the figure, the equilibrium point is at the origin, and it is an improper degenerate node. All three systems are highly unstable as they do not converge back to the origin, as shown by the points that start off at $y_2 = 5$ or 5.

Controllability and Observability

To find whether the quadcopter is fully state controllable or not, the controllability matrix W_C is calculated,

$$W_C = [B \mid AB \mid A^2B \mid A^3B \mid A^4B \mid A^5B]$$

$$= \left[\begin{array}{cccc|cccc|c} 0 & 0 & 0 & 0 & -0.0222 & -0.0222 & 0.0222 & 0.0222 & \\ 0 & 0 & 0 & 0 & 0.322 & -0.322 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.322 & -0.322 & 0 \mid 0 \mid 0 \mid 0 \\ -0.0222 & -0.0222 & 0.0222 & 0.0222 & 0 & 0 & 0 & 0 & \\ 0.322 & -0.322 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0.322 & -0.322 & 0 & 0 & 0 & 0 & \end{array}\right]$$

$$rank(W_C) = 6 = length(A)$$

The system is fully state controllable since W_C is full rank.

The observability of the system is also set to be found by calculating the observability matrix W_O,

$$W_O = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \end{bmatrix}$$

$$W_O = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$rank(W_O) = 6 = length(A)$$

The system is fully state observable since W_O is full rank.

Controller Design

1. Open Loop Analysis

In order to know whether a controller is needed, an open-loop test was conducted. The graphs below represent the response of the system for a 10V input on one of the motors. The system goes unstable, and that there is no form of self-stabilization. Therefore, comes the need for a control system.

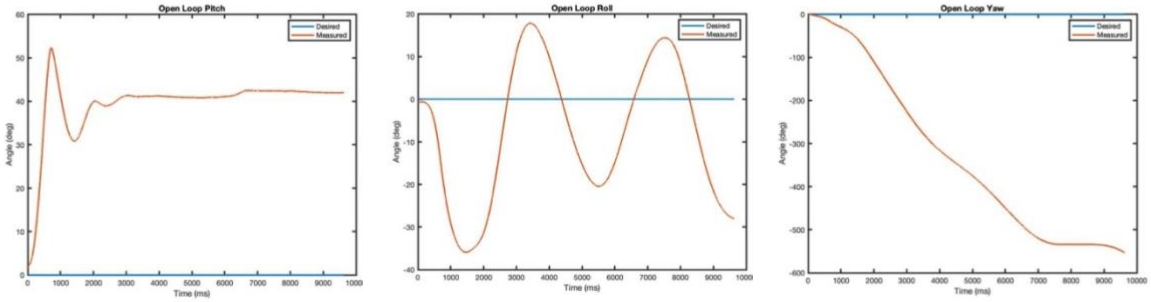
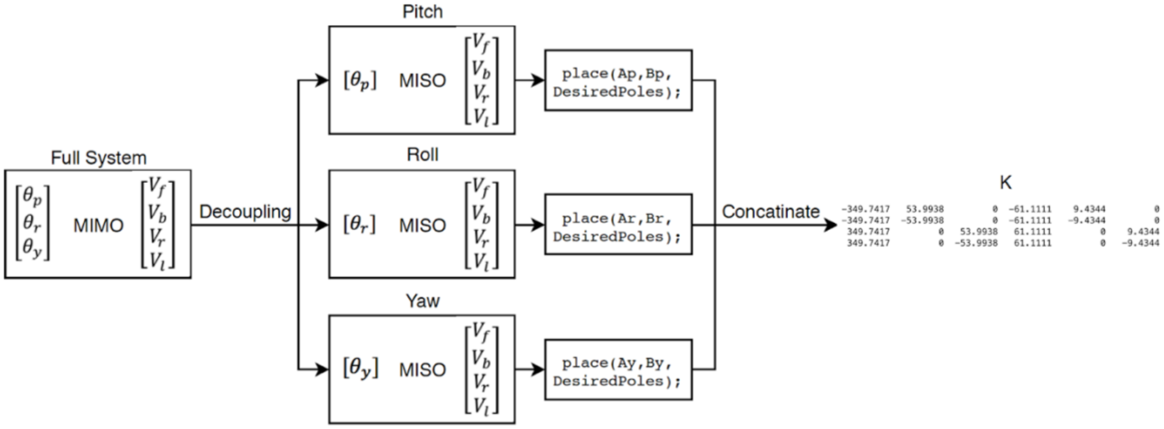


Figure 11: Open Loop Response of Quanser 3DOF Hover

2. Proportional Controller

Design

Before getting into complex controllers that optimize performance and effort, the proportional controller is to be tested and evaluated on this system. Since the techniques to design a proportional controller learned in class only apply to SISO systems, the current MIMO model had to be transformed into a SISO system. One way to accomplish the following is to decouple the system by assuming that no subsystem influences the others.


Full System
$\begin{bmatrix}\theta_p\\ \theta_r\\ \theta_y\end{bmatrix}$
MIMO
$\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$
Decoupling
Pitch
$[\theta_p]$
MISO
$\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$
place(Ap,Bp,
DesiredPoles);
Roll
$[\theta_r]$
MISO
$\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$
place(Ar,Br,
DesiredPoles);
Yaw
$[\theta_y]$
MISO
$\begin{bmatrix}V_f\\ V_b\\ V_r\\ V_l\end{bmatrix}$
place(Ay,By,
DesiredPoles);
Concatinate
K

```
-349.7417   53.9938         0  -61.1111    9.4344         0
-349.7417  -53.9938         0  -61.1111   -9.4344         0
 349.7417         0   53.9938   61.1111         0    9.4344
 349.7417         0  -53.9938   61.1111         0   -9.4344
```

Figure 12: Decoupling for the MIMO system

According to Figure 12, the full MIMO system is decoupled into its three degrees of freedom (DOF), the pitch, roll, and yaw angles. For each DOF a Multiple Input Single Output system is derived, and pole placement is applied to arrive at the desired poles determined by the required dynamic characteristics. Finally, the three gain matrices are concatenated to achieve the full system's gain matrix.

The desired poles are chosen at a maximum overshoot of 20% and a settling time of 1 second. The maximum overshoot equation and the 2% settling time equation , are used to obtain the location of the desired poles.

$$PO = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \; ; \; T_s = \frac{4}{\zeta\omega_n}$$

The following complex conjugate poles are obtained:

$$SP = -4 \pm 7.8079i$$

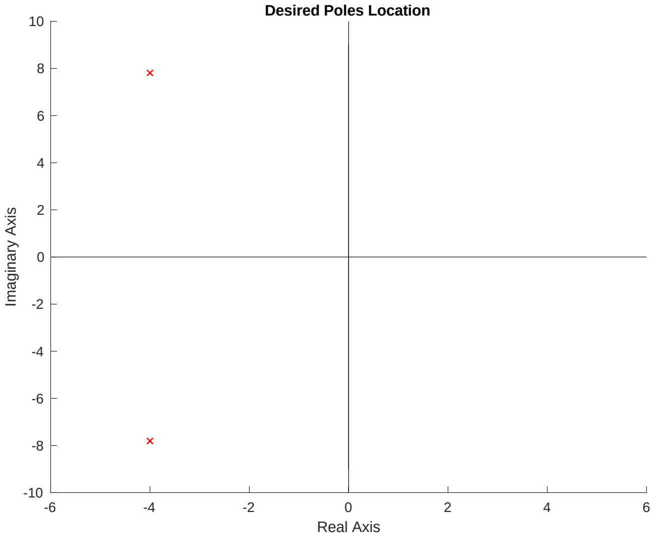


Figure 13: Desired poles location

For the pitch angle, the states, outputs, and inputs are chosen as:

$$x_p = \begin{bmatrix} \theta_p \\ \dot{\theta}_p \end{bmatrix} \; ; \; y_p = [\theta_p] \; ; \; u_p = \begin{bmatrix} V_f \\ V_b \\ V_r \\ V_l \end{bmatrix}$$

State space matrices A, B, C, and D are then obtained as:

$$A_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} ; \; B_p = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{K_f L}{J_p} & -\dfrac{K_f L}{J_p} & 0 & 0 \end{bmatrix} ; \; C_p = [1 \quad 0] ; \; D_p = [0]$$

Similarly, for the roll angle:

$$x_r = \begin{bmatrix} \theta_r \\ \dot{\theta}_r \end{bmatrix} ; \; y_r = [\theta_r] ; \; u_r = \begin{bmatrix} V_f \\ V_b \\ V_r \\ V_l \end{bmatrix}$$

$$A_r = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} ; \; B_r = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{K_f L}{J_r} & -\dfrac{K_f L}{J_r} \end{bmatrix} ; \; C_r = [1 \quad 0] \; ; \; D_r = [0]$$

Finally, the yaw angle:

$$x_y = \begin{bmatrix} \theta_y \\ \dot{\theta}_y \end{bmatrix} ; \; y_y = [\theta_y] ; \; u_y = \begin{bmatrix} V_f \\ V_b \\ V_r \\ V_l \end{bmatrix}$$

$$A_y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} ; \; B_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\dfrac{K_t}{J_y} & -\dfrac{K_t}{J_y} & \dfrac{K_t}{J_y} & \dfrac{K_t}{J_y} \end{bmatrix} ; \; C_y = [1 \quad 0] \; ; \; D_y = [0]$$

For each DOF, a gain matrix of 4x2 is obtained by applying the MATLAB command place (A, B, DesiredPoles). The following K's are obtained:

$$K_p = \begin{bmatrix} 53.99 & 9.43 \\ -53.99 & -9.43 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} ; \; K_r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 53.99 & 9.43 \\ -53.99 & -9.43 \end{bmatrix} ; \; K_y = \begin{bmatrix} -349.74 & -61.11 \\ -349.74 & -61.11 \\ 349.74 & 61.11 \\ 349.74 & 61.11 \end{bmatrix}$$

The next step is to construct the full 4x6 gain matrix K by concatenating the columns of K_p, K_r, and K_y by having the first columns of each gain matrix successively, then the second column that has an effect on the angular speed.

The gain matrix is:

$$K = \begin{bmatrix} -349.74 & 53.99 & 0 & -61.11 & 9.43 & 0 \\ -349.74 & -53.99 & 0 & -61.11 & -9.43 & 0 \\ 349.74 & 0 & 53.99 & 61.11 & 0 & 9.43 \\ 349.74 & 0 & -53.99 & 61.11 & 0 & -9.43 \end{bmatrix} \text{with } u = -Kx$$

Simulation results

Gain matrix K is saved in the Simulink simulation program and a square wave specifying the reference angles is applied to the pitch, roll, and yaw. The following results are recorded for the pitch and yaw angle, assuming that the roll and pitch behave similarly in reality:

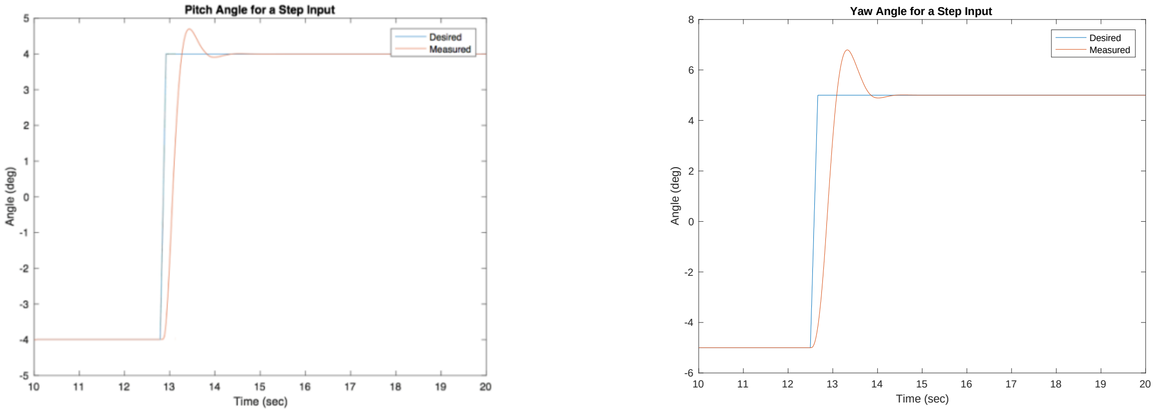


Figure 14: Simulation response for the proportional controller

As it can be seen, the proportional controller's performance fits the required specifications of 20% maximum overshoot and 1s settling time. Next, it will be tested on the actual setup.

Experimental results

For the same gain matrix K as above, the quadcopter was tested with a square wave changing from -4 to 4 degrees for the pitch and roll, and -5 to 5 degrees for the yaw at a frequency of 0.04Hz. The following results are obtained:

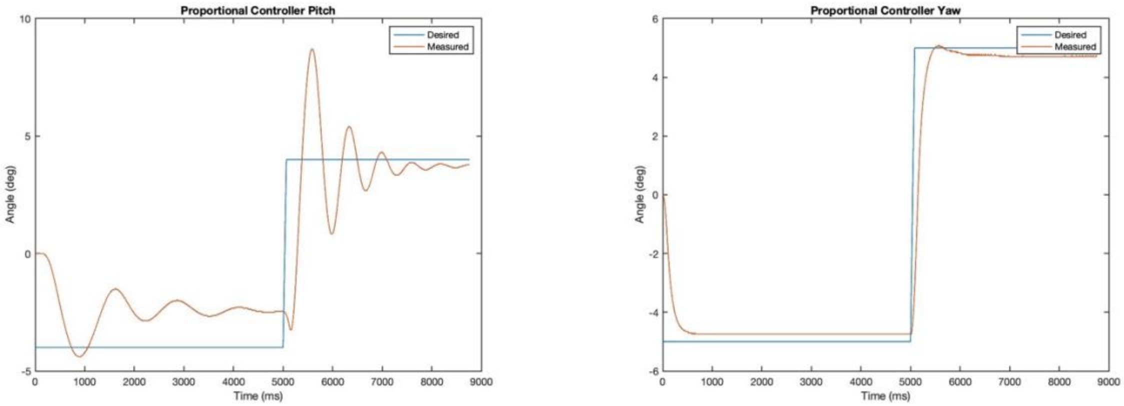


Figure 15: Experimental response for the proportional controller

From the experimental results, it can be observed that the pitch axis exhibits a large overshoot and oscillations, ending with a steady state error. For the yaw angle, the overshoot is minimal, and the settling time is small, but a steady state error is clear.

Thus, the pitch and roll can benefit from a dampening effect to smooth out the oscillations, while all of the DOFs can benefit from an integral action to remove the steady state error. Therefore, an LQR controller will be designed in the next section to optimize the tradeoff between performance and effort of the controller, while adding a derivative action for the pitch and roll.

3. Linear Quadratic Regulator (LQR)

Design

As seen in the previous section, the proportional controller leads the system to having a response with many damped oscillations. To remove these oscillations, a derivative action must be added.

The linear quadratic regulator acts as Proportional-Derivative controller.

To design the LQR, the cost function

$$V = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)\, dt$$

Should be minimized where Q is a positive semidefinite matrix that represents the cost on the states and R is positive definite matrix that represents the cost on the inputs.

Initially, Q and R are chosen as follows,

$$Q = C^T C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

However, the Q is tuned by increasing 5 diagonal entries thereby penalizing bad performance on all states except the yaw velocity,

$$Q = \begin{bmatrix} 1750 & 0 & 0 & 0 & 0 & 0 \\ 0 & 450 & 0 & 0 & 0 & 0 \\ 0 & 0 & 450 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 50 \end{bmatrix}$$

To obtain the gain K of the LQR, the P matrix should be solved for using the Algebraic Ricatti Equation (ARE):

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

Where P is symmetric and positive definite.

Using MATLAB's icare command,

$$P = \begin{bmatrix} 574.0281 & 0 & 0 & 94.1452 & 0 & 0 \\ 0 & 163.3797 & 0 & 0 & 4.6588 & 0 \\ 0 & 0 & 163.3797 & 0 & 0 & 4.6588 \\ 94.1452 & 0 & 0 & 30.8811 & 0 & 0 \\ 0 & 4.6588 & 0 & 0 & 1.6915 & 0 \\ 0 & 0 & 4.6588 & 0 & 0 & 1.6915 \end{bmatrix}$$

Therefore, the gain K of the LQR can be found,

$$K = R^{-1}B^T P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -0.0222 & 0.322 & 0 \\ 0 & 0 & 0 & -0.0222 & -0.322 & 0 \\ 0 & 0 & 0 & 0.0222 & 0 & 0.322 \\ 0 & 0 & 0 & 0.0222 & 0 & -0.322 \end{bmatrix} P$$

$$= \begin{bmatrix} -209.1650 & 150 & 0 & -68.6095 & 54.4599 & 0 \\ -209.1650 & -150 & 0 & -68.6095 & -54.4599 & 0 \\ 209.1650 & 0 & 150 & 68.6095 & 0 & 54.4599 \\ 209.1650 & 0 & -150 & 68.6095 & 0 & -54.4599 \end{bmatrix}$$

Simulation results

Applying a square wave as a reference for the system to follow, it can be seen that derivative action successfully most of the damped oscillations for both yaw and pitch angles.

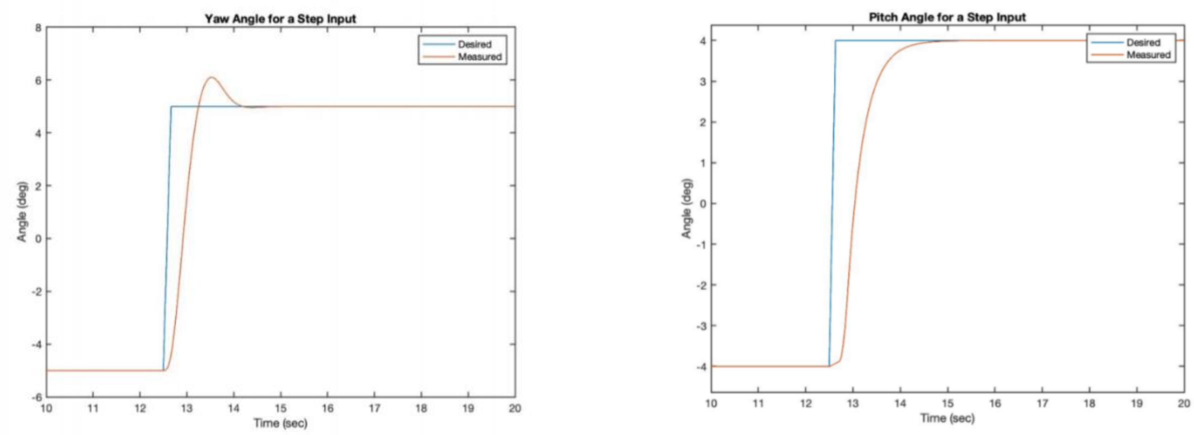


Figure 16: Simulation response for the LQR

Experimental results

The experimental results confirm what has been seen in the simulation. The oscillations have disappeared from the response of the both the yaw and pitch.

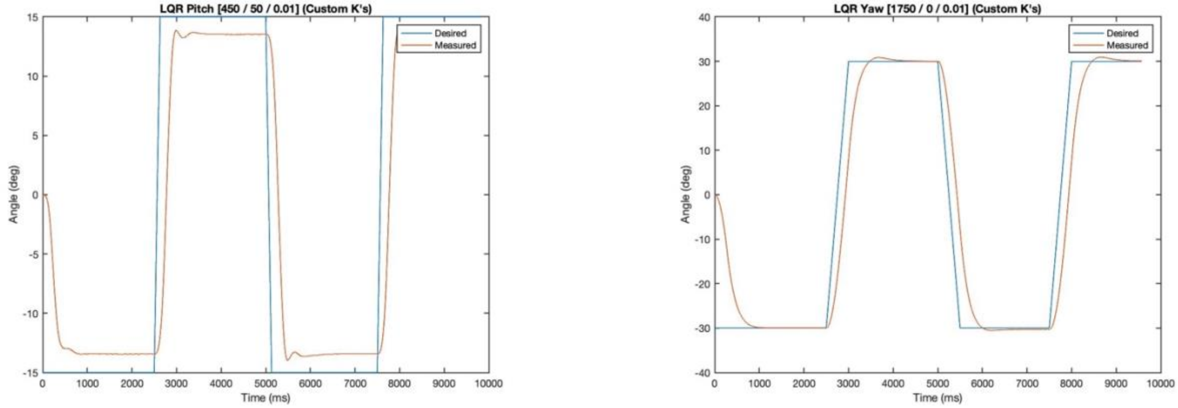


Figure 17: Experimental response for the LQR

As seen in the pitch response, the steady state error is significant. Therefore, an integral action through a Linear Quadratic Integrator (LQI) is required to reduce the error to 0.

4. Linear Quadratic Integrator (LQI)

Design

To minimize the steady state error, three new states, representing the integral of the errors, are introduced:

$$\hat{x} = \begin{bmatrix} \int (\theta_y - \theta_{y_d})\, dt \\ \int (\theta_p - \theta_{p_d})\, dt \\ \int (\theta_r - \theta_{r_d})\, dt \end{bmatrix}$$

These new states will be forced to go to zero.

Since states are added, the whole system has been augmented with a total of 9 states, represented by $\tilde{x}$

$$\tilde{x} = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

Therefore, keeping the same inputs and outputs, the defining matrices for the augmented system are:

$$\tilde{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.0222 & -0.0222 & 0.0222 & 0.0222 \\ 0.322 & -0.322 & 0 & 0 \\ 0 & 0 & 0.322 & -0.322 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} C & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\widetilde{D} = D$$

To design the LQI, LQR design is performed on the augmented system. Q and R are initially chosen to be:

$$Q = \tilde{C}^T \tilde{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

After penalizing bad performance on all the states except the yaw velocity,

$$Q = \begin{bmatrix} 1200 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 450 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 450 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 35 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 35 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 13 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 200 \end{bmatrix}$$

Using the lqr command on MATLAB, the gain is obtained.

$$K = [K_m \; K_i]$$

$$= \left[\underbrace{\begin{matrix} -179.7023 & 179.7023 & 0 & -63.594 & 48.0261 & 0 \\ -179.7023 & -179.7023 & 0 & -63.594 & -48.0261 & 0 \\ 179.7023 & 0 & 179.7023 & 63.594 & 0 & 48.0261 \\ 179.7023 & 0 & -179.7023 & 63.594 & 0 & -48.0261 \end{matrix}}_{K_m} \middle| \underbrace{\begin{matrix} -18.0278 & 100 & 0 \\ -18.0278 & -100 & 0 \\ 18.0278 & 0 & 100 \\ 18.0278 & 0 & -100 \end{matrix}}_{K_i} \right]$$

The gain K is split such that its last three columns serve as an integral gain on only the last 3 states, $\hat{x}$, while the first 6 columns serve as a type of LQR gain on the original states x.

Simulation results

Taking a square wave as a reference for the system to follow, it can be seen that integral action removed the steady state error from the response of the yaw and pitch.

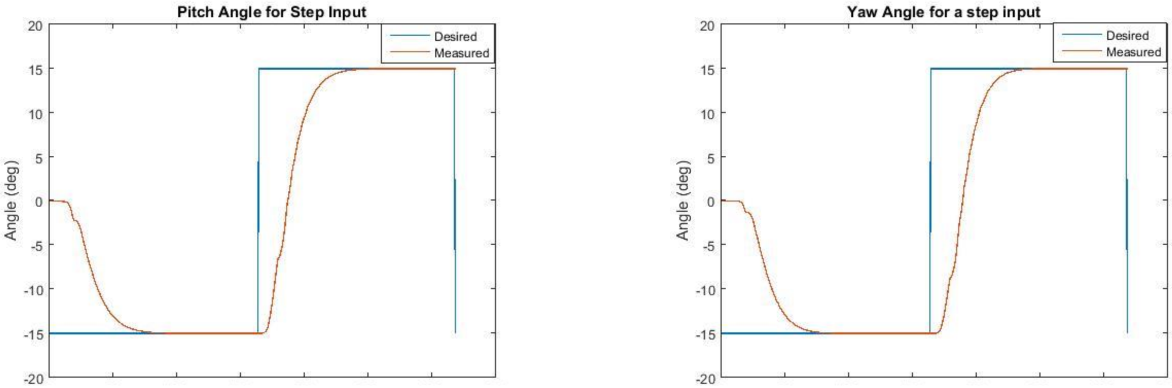


Figure 18: Simulation response for the LQI

Experimental results

As shown in the pitch response, although the settling time is high, the steady state is being reduced.

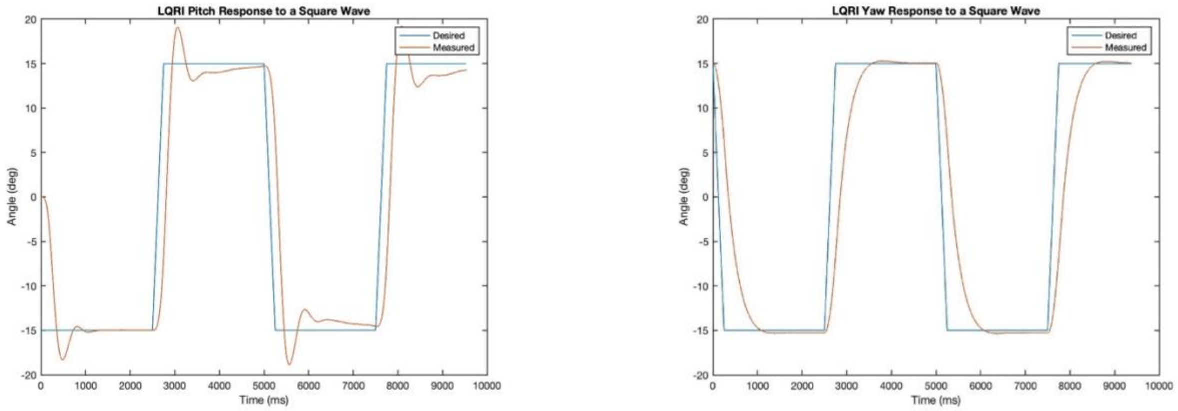


Figure 19: Experimental response for the LQI

System Response to Disturbances

Increased Moment of Inertia

The first test conducted was to increase the moment of inertia of the drone by attaching additional weights on its four motors while maintaining the balance of the drone, as seen in figure 20.

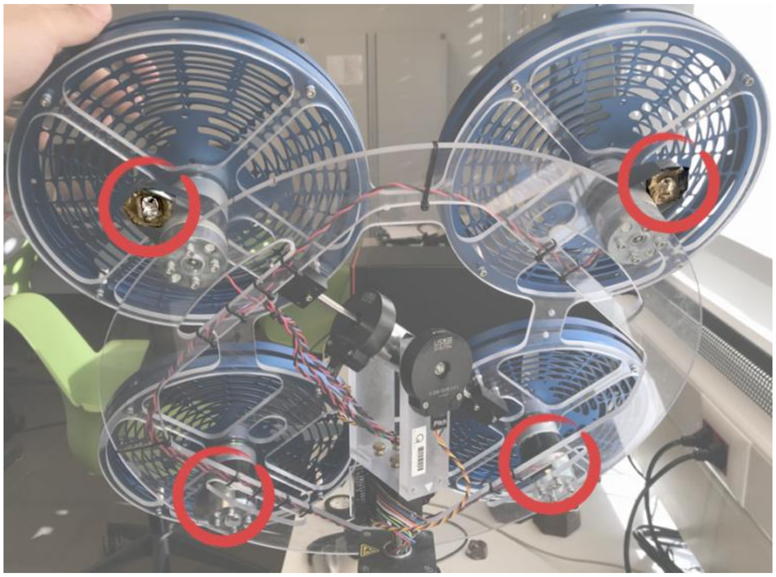


Figure 20: Increased Moment of Inertia

LQR Controller Response

The results obtained with the LQR controller show a significant increase in overshoot on the Pitch and Roll and a slight increase in steady-state error on the Pitch, Roll and Yaw

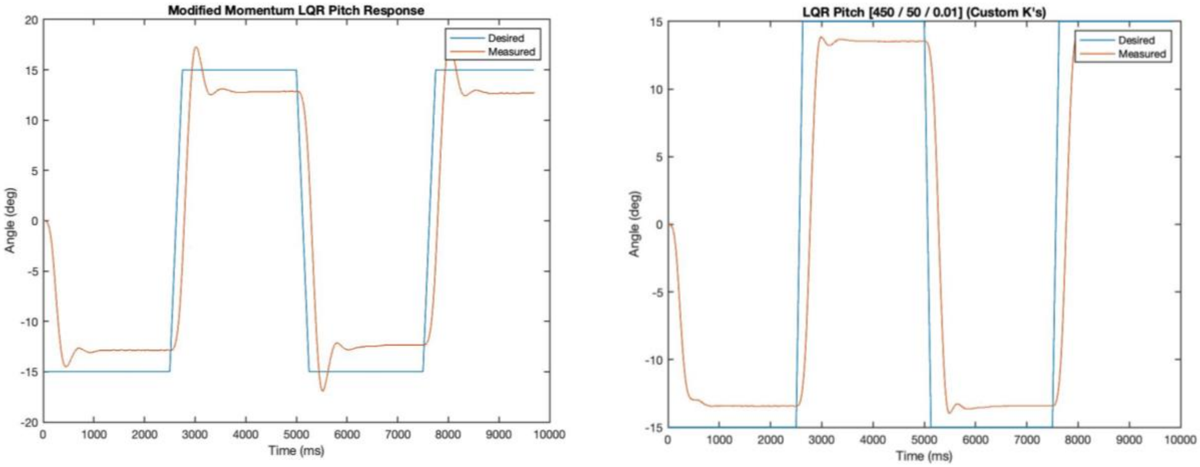


Figure 21: Comparison of LQR Pitch response to a square wave with (Left), and without (right) increased moment of inertia

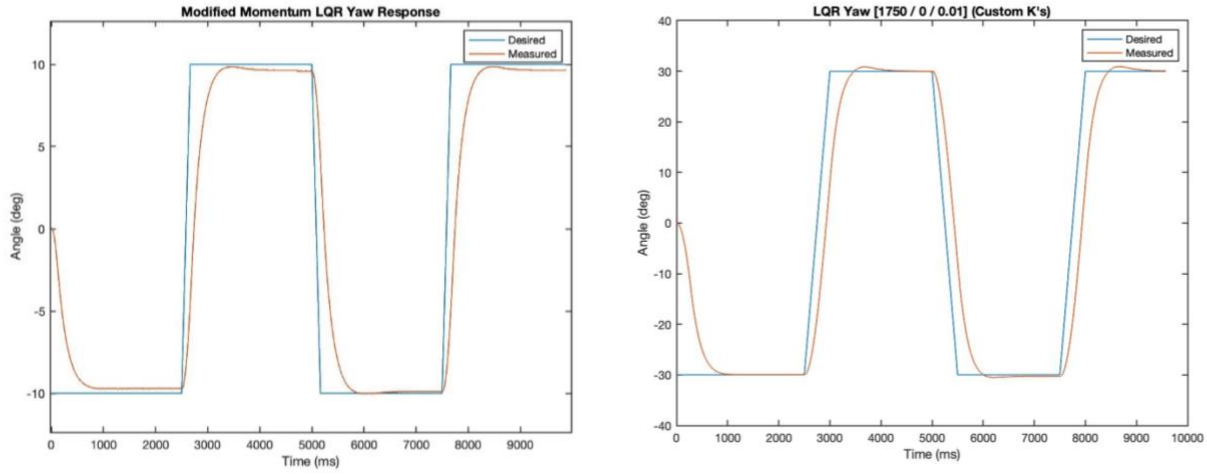


Figure 22: Comparison of LQR Yaw response to a square wave with (Left), and without (right) increased moment of inertia

LQI Controller Response

On the other hand, the LQI turned out to be much more robust against the unbalance, and no significant change in the behavior was noticed.

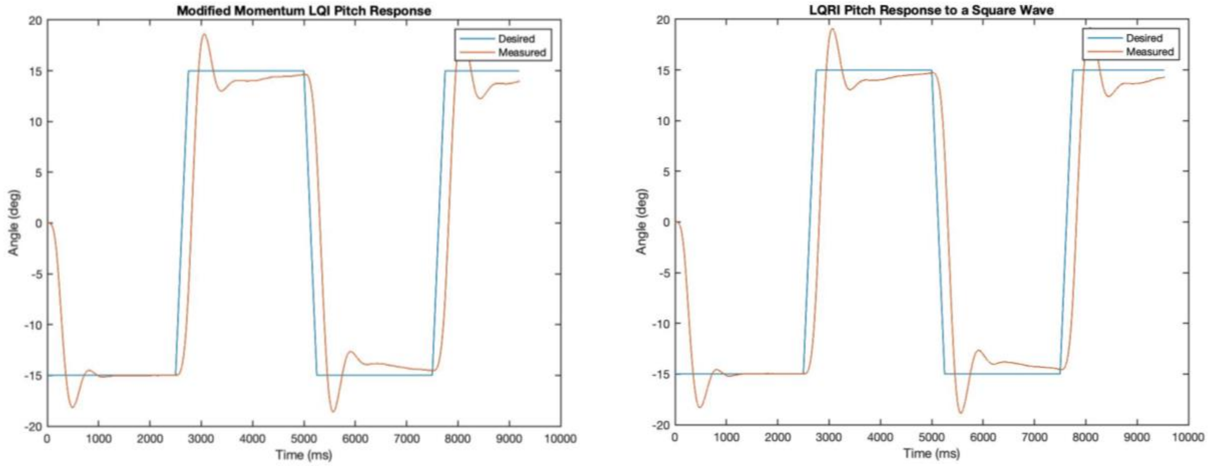


Figure 23: Comparison of LQI Pitch response to a square wave with (Left), and without (right) increased moment of inertia

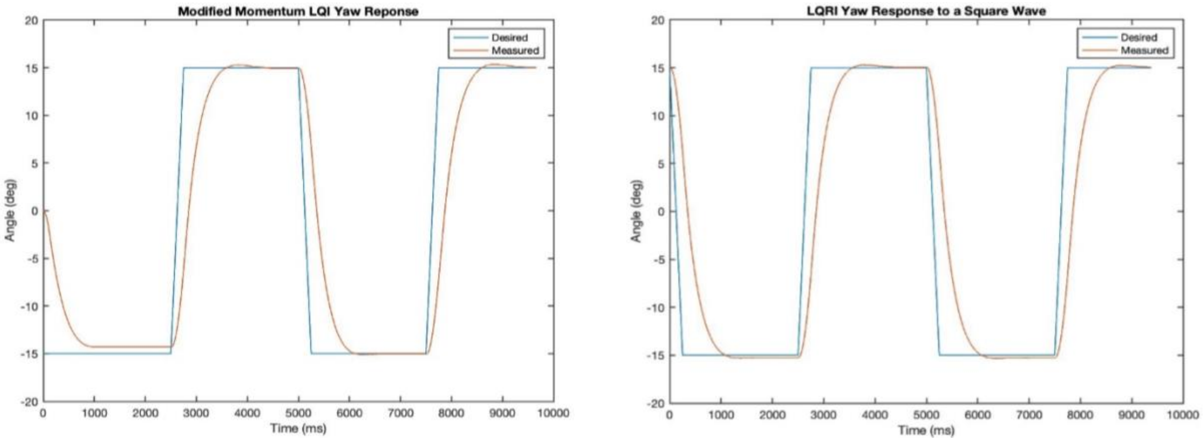


Figure 24: Comparison of LQI Yaw response to a square wave with (Left), and without (right) increased moment of inertia

Unbalanced Drone

The next test that was conducted was to attach a 190g mass between two motors using a short rope, to prevent swinging of the mass, in order to create an unbalance on the drone. Then the drone should just try to hover in place, after reaching a steady state, the rope was be cut and the response of the drone to this sudden change was be recorded.

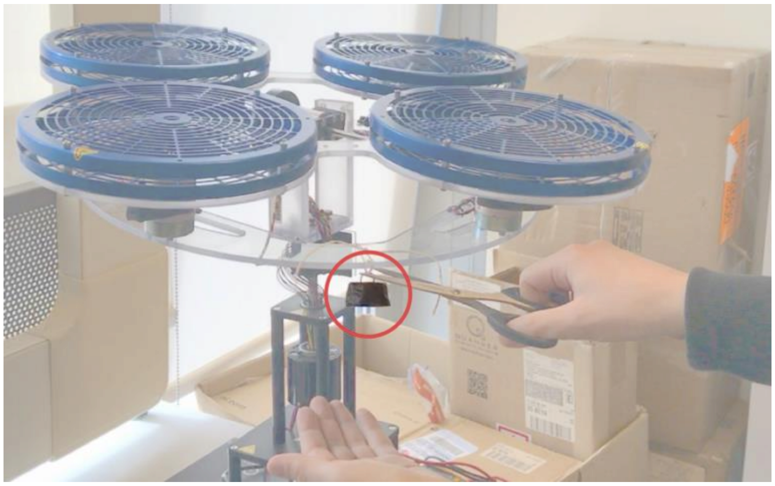


Figure 25: Unbalanced drone test

LQR Controller Response

Using the LQR controller, it is clear that with the additional mass, the drone is unable to stay level, therefore it oscillates with a magnitude of 1 degree and at an offset of approximately 4 degrees. Moreover, when the rope is cut, the drone goes back to steady state with very little oscillations.

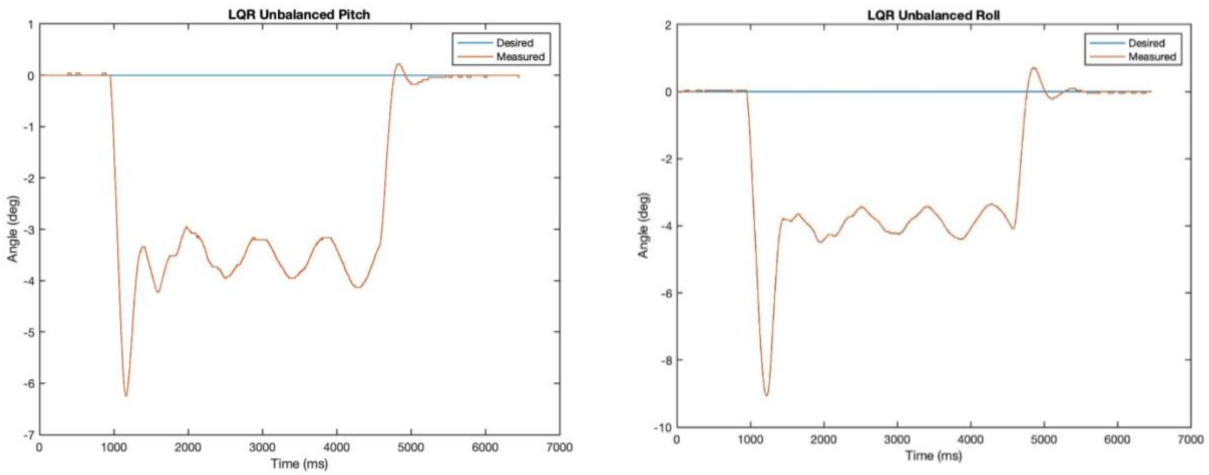


Figure 26: Response of the LQR controller to an unbalance, and sudden removal of this unbalance

LQI Controller Response

The LQI, on the other hand, is much more aggressive and oscillates with a magnitude of 3 degrees, but with only very little offset which tends to 0 because of the integral factor. Moreover, when the unbalance is removed, the controller causes an extremely high overshoot of 15 degrees and then takes approximately 2 seconds to go back to steady-state.

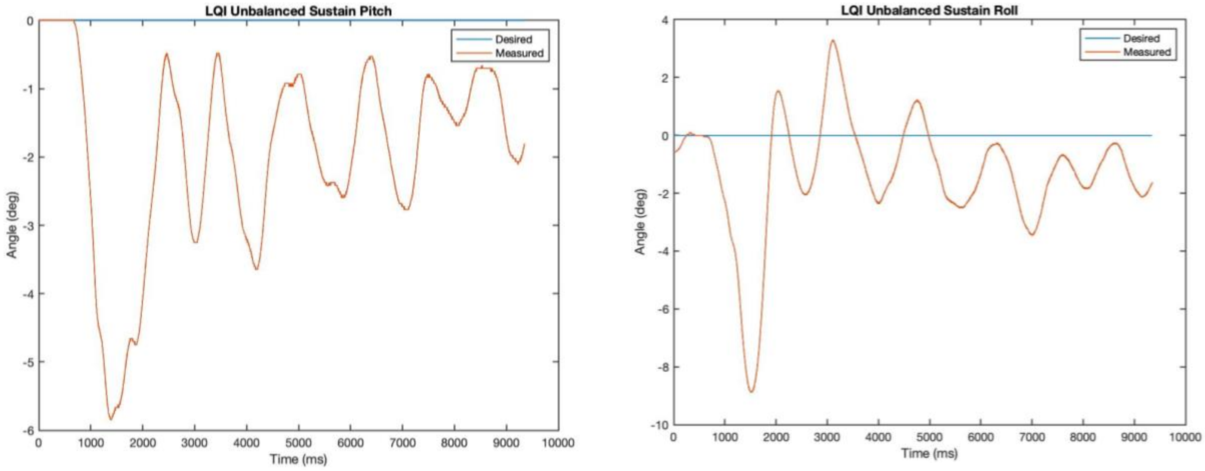


Figure 27: Steady-state Response of the LQI controller to an unbalance

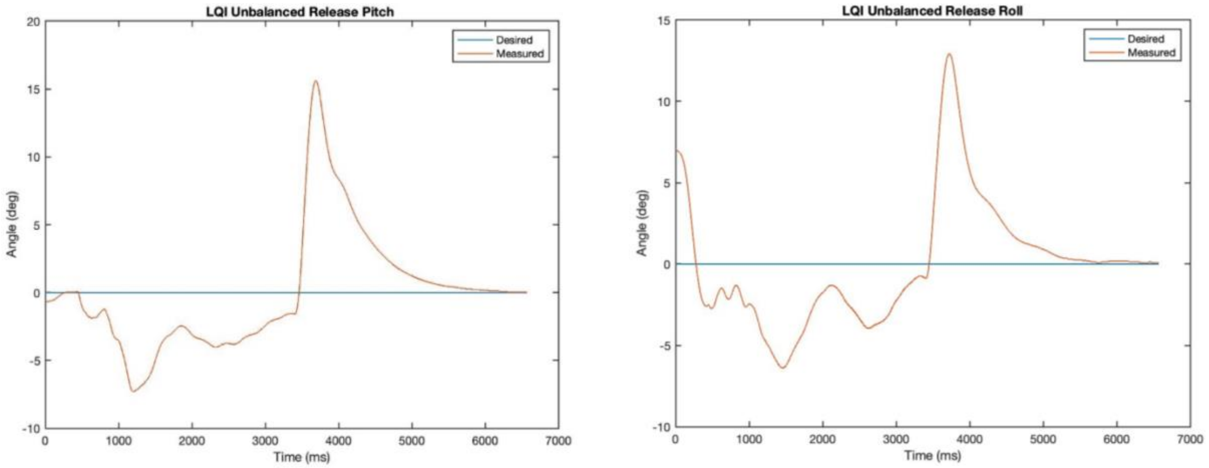


Figure 28: Response of the LQI controller to the sudden removal of the unbalance

Ground Effect

The next test conducted studied the ground effect on the quadrotor. The ground effect happens when propellers are close to a planar surface, where the air pushed down by the propeller is trapped between the propeller and the plane, increasing the pressure below the propeller, and thus increasing the lift created. In some cases, this increased lift will cause instability since it affects the overall response of the system. Figure 29 shows our setup, and highlights (in green) the high-pressure zone created due to the ground effect.

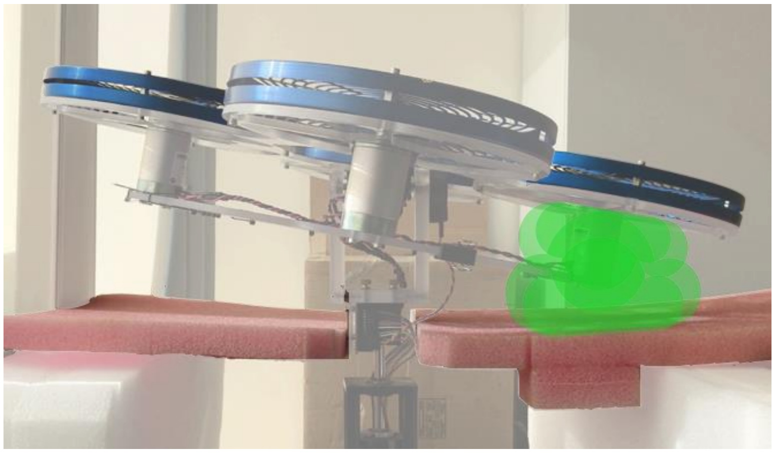


Figure 29: Ground Effect Setup

LQR Controller Response

With the ground effect, the LQR controller showed less overshoot, but more undershoot and steady-state error. By looking closely at the two graphs in figure 30, it is clear that both graphs have the same response shape, but the only difference is that the graph representing the ground effect is shifted towards the center of the graph, and this is due to the increased pressure caused by the ground effect pushing the drone away of the plane.

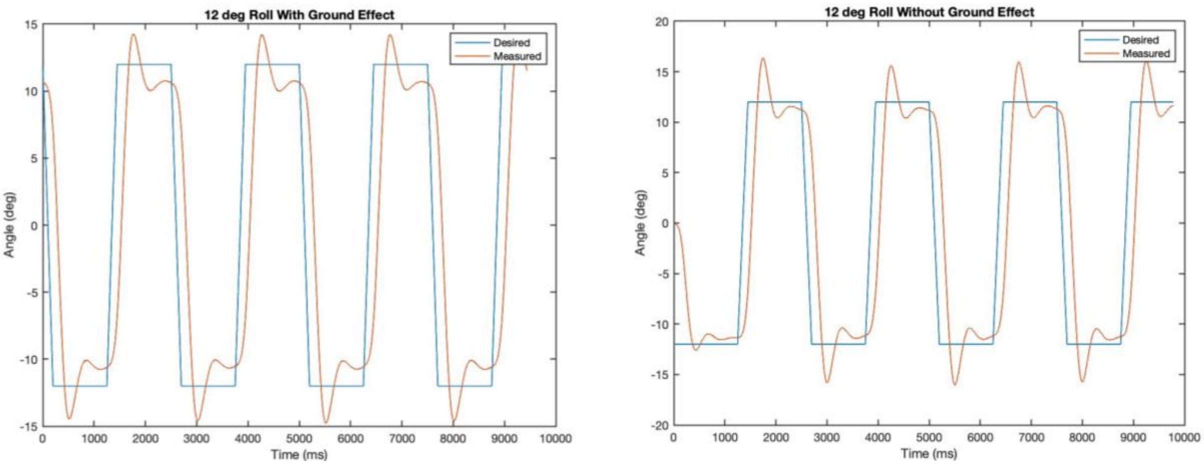


Figure 30: Response of the LQR controller to a square wave with (left), and without (right) ground effect

Dancing Drone

The last test conducted was aimed at finding whether or not having the controllers control the pitch, roll, and yaw all-together would have an effect on the drone. The desired behavior was to have the drone constantly yaw while being tilted at a certain angle. In order to do this, the controller was given a ramp input for the yaw, and the sine and cosine of the value of this ramp would be fed to the pitch and roll. A demonstration video can be found here.

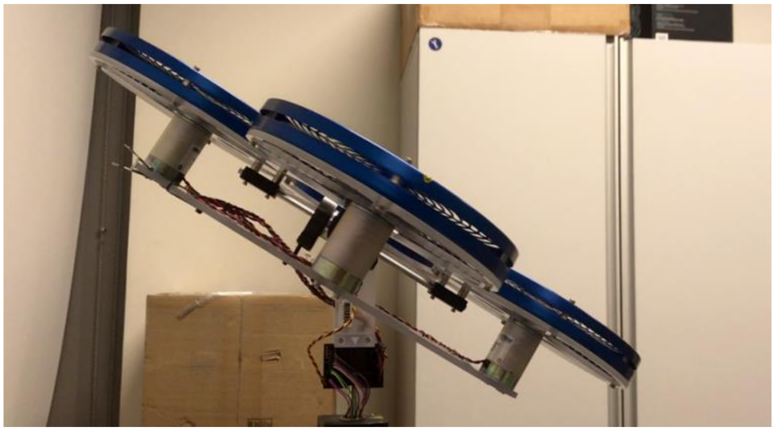


Figure 31: Dancing Drone test

LQI Controller Response

While conducting this test, a weird behavior was found to occur on the Roll of the quadrotor as seen in figure 32. After applying only a sinusoidal wave on the roll, it turned out that as the drone got closer to level, both propellers would be shut down for a certain delay, and this is due to delay of switching, therefore in order to solve this issue, the bias voltage that would account for the dead zone of the motor was increased. Following this increase, the problem was solved as seen in figure 33. Moreover, it can be seen that the drone tracks the desired angle with very little steady-state error, thus there is no negative effect of simultaneously controlling the pitch, roll, and yaw states.

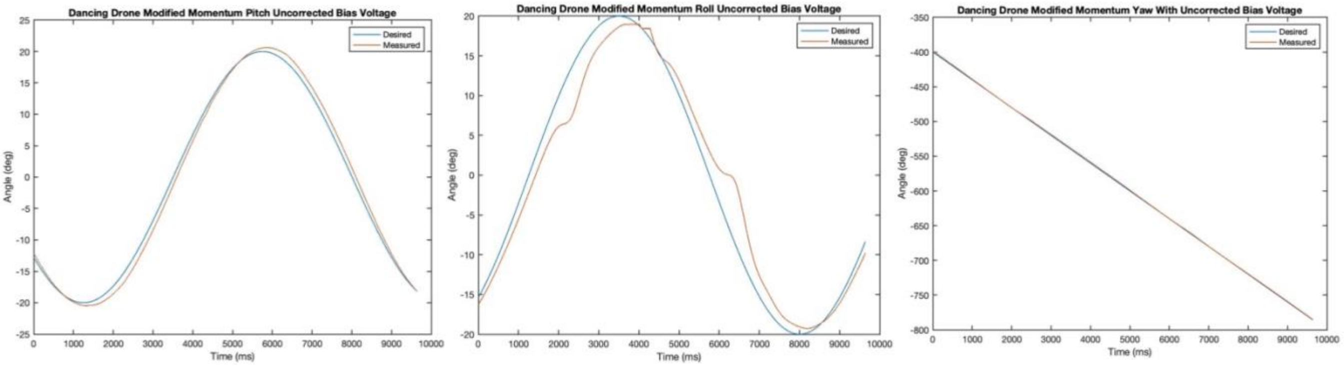


Figure 32: Response of LQI controller to Dancing Drone input before the increase of Bias Voltage

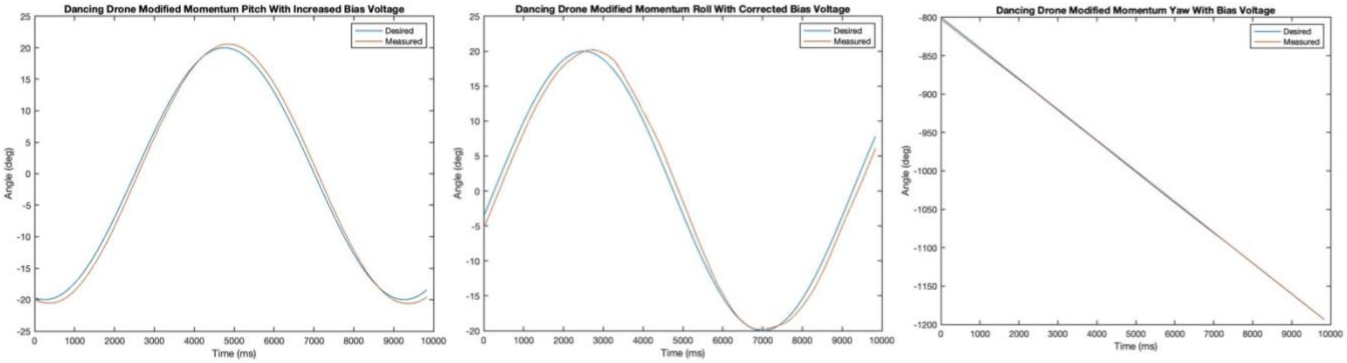


Figure 33: Response of LQI controller to Dancing Drone input after the increase of Bias Voltage

Conclusion

In this study, different controllers have been developed and deployed on the Quanser 3DoF Hover testbed. The proportional controller exhibited a large overshoot, settling time, and oscillations, while the LQR controller placed a weight on performance resulting in a more aggressive response in terms of rise and settling times. However, both controllers had a steady state error and weren't effective in rejecting disturbances. An LQI controller was designed, simulated, and experimentally proven to remove the steady state error and successfully reject various disturbances including a change in the moment of inertia, an unbalanced system, and the ground effect. Also, the controller exhibited a tracking behavior to a sine wave on the pitch and roll, and a ramp on the yaw with minimal error. The current controller relied on its integral component to reject the disturbances inflicted on the system. It would be interesting to model in a future work the disturbances inside the system and estimate their value. This would be useful in quadcopter applications involving a variable weight and having to employ an adaptive controller capable of adapting to these different situations.

Glossary

$K_f = \text{the propeller thrust} - \text{force constant} = 0.090217\ \text{N/V}$

$K_t = \text{the propeller thrust} - \text{torque constant} = 0.0024439\ \text{N m/V}$

$L = \text{Distance between pivot to each motor} = 0.00197\ m$

$J_p = \text{Equivalent moment of inertia about pitch axis} = 0.0552\ \text{kg}\ m^2$

$J_r = \text{Equivalent moment of inertia about roll axis} = 0.0552\ \text{kg}\ m^2$

$J_y = \text{Equivalent moment of inertia about yaw axis} = 0.11\ \text{kg}\ m^2$

References

[1] J. Primicerio et al, "A flexible unmanned aerial vehicle for precision agriculture," Precision Agriculture, vol. 13, (4), pp. 517-523, 2012.

[2] M. Garcia Fernandez et al, "Synthetic Aperture Radar Imaging System for Landmine Detection Using a Ground Penetrating Radar on Board an Unmanned Aerial Vehicle," IEEE Access, vol. 6, pp. 45100-45112, 2018.

[3] T. Niedzielski et al, "A real-time field experiment on search and rescue operations assisted by unmanned aerial vehicles," Journal of Field Robotics, vol. 35, (6), pp. 906-920, 2018.

[4] S. Campana, "Drones in Archaeology. State-of-the-art and Future Perspectives," Archaeological Prospection, vol. 24, (4), pp. 275-296, 2017.

[5] I. Sung and P. Nielsen, "Zoning a Service Area of Unmanned Aerial Vehicles for Package Delivery Services," Journal of Intelligent & Robotic Systems, 2019.

[6] A. Goodchild and J. Toy, "Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry," Transportation Research Part D, vol. 61, pp. 58, 2018.

[7] A. L. Salih et al, "Flight PID controller design for a UAV quadrotor," Scientific Research and Essays, vol. 5, (23), pp. 3660-3667, 2010.

[8] A. Razinkova et al, “Constant Altitude Flight Control for Quadrotor UAVs with Dynamic Feedforward Compensation,” International Journal of Fuzzy Logic and Intelligent Systems, vol. 14, pp. 26-33, 2014.

[9] M. Abdelkhalek et al, “Attitude Stability of Quadcopter Using Classic Control with Angular Acceleration,” International Journal of Computer Science, Information Technology, & Security, vol. 5, pp. 325-331, 2015.

[10] O. Kaan et al, “Dynamic Model and Control of a New Quadrotor Unmanned Aerial Vehicle with Tilt-Wing Mechanism,” World Academy of Science, Engineering and Technology, vol. 45, pp. 58-63, 2008.

[11] S. Stevanović et al, "Robust tracking control of a quadrotor helicopter without velocity measurement," In Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, pp. 595-600, 2012.

[12] N. Tamami et al, “Proportional Derivative Active Force Control for “X” Configuration Quadcopter,” Journal of Mechatronics, Electrical Power, and Vehicular Technology, vol. 5, pp. 67-74, 2014.

[13] H. Suresh, A. Sulficar and V. Desai, "Hovering control of a quadcopter using linear and nonlinear techniques," International Journal of Mechatronics and Automation, vol. 6, pp. 120-129, 2018.