A logo on a black background

Description automatically generated

Final Project Report

RBE 502

Jessica Hart

December 12, 2024

## Introduction

Blah blah blah intro shit

Quadcopter control is a complex topic, with many different approaches. For this project, the goal was to design two controllers. The first was a proportional derivative controller, and the second an optimal controller. For the optimal controller, there were two options, either a linear quadratic regulator or a model predictive controller. For the optimal controller, the linear quadratic regulator was selected.

## Methodology

I picked controllers and blah blah

#### Control Inputs

For the CrazyFile 2.0 quadcopter, there are two inputs. The first is the total thrust of the propellers, expressed as the sum of each of the four individual propellers (Equation 1). The second is the vector of moments experienced by the quadcopter about the , , and axes.

The given parameters were as follows:

|  |  |  |
| --- | --- | --- |
| Variable | Symbol | Value |
| Mass | m | 0.03kg |
| Gravity | g |  |
| Mass moment of inertia about x |  |  |
| Mass moment of inertia about y |  |  |
| Mass moment of inertia about z |  |  |
| Center of mass to center of rotor distance | l | 0.046m |
|  |  |  |

#### Proportional Derivative Controller

The first step for designing the proportional derivative controller was to determine the gain variables. For this simulation, there were 12 gain values, a proportional and derivative component for each axis of positional and angular control. Table 1 shows the positional gains, and Table 2 shows the angular gains.

Table 1: Proportional and derivative gains for position in the , , and () axes.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | 4.15 | 10.25 |
|  | 4.15 | 10.25 |
|  | 3.00 | 3.5 |

Table 2: Proportional and derivative gains for roll, pitch, and yaw ()

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | 0.5 | 3.0 |
|  | 0.5 | 3.0 |
|  | 1.2 | 1.8 |

After establishing the gains, the desired accelerations, roll, and pitch were calculated with the following equations:

After calculating the above values, the control inputs could be determined with the following equations:

#### Linear Quadratic Regulator

To implement the Linear Quadratic Regulator, the first step was to identify the linearized state equations to generate the A and B matrices. The sate variables and state space representation were identified as:

Using the linearized equations from the project statement, combined with near hover assumptions with small roll and pitch angles such that,

Using these expressions, the state space equations can be represented as:

Reformatting to produces these A and B matrices:

## Results

It did shit

## Discussion

Pd sucks LQR better but slow as fuck