

# Quadcopter Dynamics and Numerical Simulation

## MAE 542 Final Project

Matthew Coleman  
December 6, 2022

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Problem Setup</b>	<b>3</b>
2.1	Reference Frames . . . . .	3
2.2	Forces and Moments . . . . .	3
2.3	Rotation . . . . .	4
2.4	State . . . . .	5
<b>3</b>	<b>Dynamics</b>	<b>6</b>
3.1	Lagrangian . . . . .	6
3.2	Translations . . . . .	6
3.3	Rotations . . . . .	7
<b>4</b>	<b>Controls</b>	<b>9</b>
<b>5</b>	<b>Simulations</b>	<b>10</b>
5.1	Pure Dynamics . . . . .	10
5.2	Simulated Control . . . . .	12
<b>6</b>	<b>Discussion</b>	<b>14</b>
<b>7</b>	<b>Bibliography</b>	<b>15</b>
<b>8</b>	<b>Appendix</b>	<b>16</b>
8.1	Coriolis Matrix . . . . .	16

# 1 Introduction



Figure 1: DJI Mavic 3 Quadcopter

A quadcopter is a helicopter with four propellers arranged in a square formation. It can be controlled by rapidly changing the angular velocity of each rotor independently, such that a directional thrust and torque can be achieved about each rotor and likewise about the body of the drone. They have a wide variety of uses including aerial photography, search and rescue, agriculture, and surveillance [4].

Although the structure is simple to construct and models are widely accessible on consumer markets, the dynamics and control systems required to make any use of the physical system are complex and nonlinear. In addition, quadcopters can be expensive and difficult to learn to control for a human operator. Thus, the setting is ideal for rigorous dynamical analysis, and a comprehensive numerical simulation of the quadrotor dynamics would be particularly helpful for someone that wishes to learn how to pilot one without running the risk of destroying it accidentally or otherwise damaging the drone or the surroundings.

In this work, I will derive the equations of motion of a quadcopter using the Euler-Lagrange equations, develop a simple PID controller to stabilize the drone horizontally, and present numerical simulations testing the validity of the dynamics and controller. The code and configurations used to generate these results are available at this [public repository](#).

## 2 Problem Setup

### 2.1 Reference Frames

Figure 2 shows an inertial reference frame  $\mathcal{I}$  as well as the body frame  $\mathcal{B}$  of the quadcopter, which has its origin centered on the quadcopter's center of mass  $G$  and principle axes aligned with the quadcopter's rotors. The positions of the rotors  $\bar{r}_i$  are orthogonal to each other and have identical length  $l$ , and their orientations are fixed in the body frame, i.e., they cannot rotate independently of the drone structure. In the following derivations I will consider the drone structure as a rigid body with controllable point forces emitted upwards from the center of each rotor and moments which act about the center of each rotor. The effects of air resistance will be neglected.

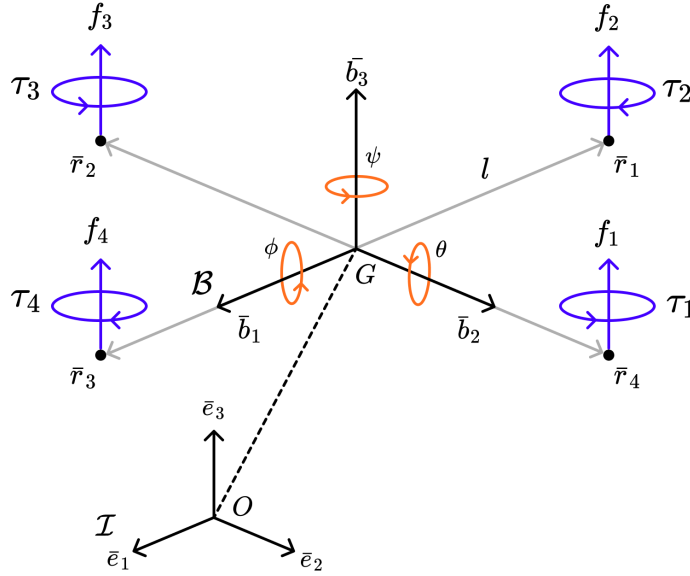


Figure 2: Quadcopter frames, forces and moments

With this setup laid out, the goal of this project is therefore to derive the equations of motion of the quadrotor using the Lagrangian formulation of dynamics and then to carry out numerical experiments using the results. From the figure, one can imagine that the evolution of the quadrotor state can easily become chaotic in flight, as it will be very easy for small changes in the roll or pitch of the body to produce oscillations or even invert the craft completely.

### 2.2 Forces and Moments

From Figure 2 it is clear that the net force on the drone will be the sum of the 4 individual propeller thrusts acting upwards along the body frame vertical axes. Each propeller generates a thrust that is proportional to a constant  $k$  multiplied by its angular velocity  $\omega_i$  squared [3]. Thus:

$$f_i = k\omega_i^2, \quad \mathbf{F} = k \begin{pmatrix} 0 & 0 & \sum_{i=1}^4 \omega_i^2 \end{pmatrix}^T, \quad \mathbf{f} = B\mathbf{F} \quad (1)$$

describes the force  $\mathbf{F}$  on the quadrotor center of mass in the body frame as well as the force  $\mathbf{f}$  on the quadrotor center of mass in the inertial frame. The rotation matrix  $B$  which maps vectors in the body frame to the inertial frame will be derived in the following section.

Furthermore, the moment produced by each rotor is proportional to a different constant  $b$  multiplied by its angular velocity  $\omega_i^2$  squared. Since each rotor is fixed and the moments produced by each blade are parallel, they can also be summed, i.e.,  $\boldsymbol{\tau}_\psi = \sum_i \boldsymbol{\tau}_i$ . For  $\boldsymbol{\tau}_\theta$  and  $\boldsymbol{\tau}_\phi$ , consider the forces produced by the pairs of rotors opposite each other about the origin. Also note, neighboring rotors spin in opposite directions. Thus:

$$\boldsymbol{\tau} = \begin{pmatrix} l(f_1 - f_3) \\ l(f_2 - f_4) \\ \sum_{i=1}^4 \tau_i \end{pmatrix} = \begin{pmatrix} kl(\omega_1^2 - \omega_3^2) \\ kl(\omega_2^2 - \omega_4^2) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{pmatrix} \quad (2)$$

## 2.3 Rotation

Across several sources [1, 5, 2], the most varied choice in deriving the dynamics of the quadcopter is the parametrization of rotation between the inertial frame  $\mathcal{I}$  and body frame  $\mathcal{B}$ . The most straightforward approach, however, is the Euler-angle representation, which yields the rotation matrix  $B$  through the composition of three rotations about a fixed principle axis.

Thus, the following generalized coordinates will be considered to parametrize the rotations:

$$\boldsymbol{\eta} = (\phi \quad \theta \quad \psi)^T, \quad \dot{\boldsymbol{\eta}} = (\dot{\phi} \quad \dot{\theta} \quad \dot{\psi})^T \quad (3)$$

Each component matrix of the Euler-angle representation along with the final representation itself are members of the special orthogonal group  $SO(3)$ . As we have seen in class, members of this group can be generated via matrix exponentials, i.e. (Using  $c_\theta$  to represent  $\cos \theta$  and  $s_\theta$  to represent  $\sin \theta$ , etc.):

$$e^{\phi \hat{e}_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix}, \quad e^{\theta \hat{e}_2} = \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix}, \quad e^{\psi \hat{e}_3} = \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

There are numerous conventions of the rotation matrix, but in this case I will use the Tait-Bryan convention representing a sequence of yaw, pitch, and roll angles, which is the most common in the aerospace setting. Thus, the final representation of the rotation through  $\psi$ ,  $\theta$ , and  $\phi$  successively is given by the following:

$$B(\phi, \theta, \psi) = e^{\psi \hat{e}_3} e^{\theta \hat{e}_2} e^{\phi \hat{e}_1} \quad (4)$$

$$= \begin{pmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix} \quad (5)$$

$$= \begin{pmatrix} c_\psi c_\theta & s_\phi s_\theta c_\psi - s_\psi c_\phi & s_\phi s_\psi + s_\theta c_\phi c_\psi \\ s_\psi c_\theta & s_\phi s_\psi s_\theta + c_\phi c_\psi & -s_\phi c_\psi + s_\psi s_\theta c_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \quad (6)$$

Equation 6 provides a clear formulation of the rotation matrix  $B$  as a function of  $\boldsymbol{\eta}$ , however, it will also be useful to derive an expression for the angular velocity as a function of  $\dot{\boldsymbol{\eta}}$ . To do this, we can consider each dimension of the angular velocity as each coordinate of  $\dot{\boldsymbol{\eta}}$  is transformed into the body-fixed frame  $\mathcal{B}$ .

$$\Omega_\phi = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \dot{\phi} \quad (7)$$

$$\Omega_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \dot{\theta} = \begin{pmatrix} 0 \\ c_\phi \\ s_\phi \end{pmatrix} \dot{\theta} \quad (8)$$

$$\Omega_\psi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{pmatrix} \begin{pmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{\psi} = \begin{pmatrix} s_\theta \\ -c_\theta s_\phi \\ c_\theta c_\phi \end{pmatrix} \dot{\psi} \quad (9)$$

Thus, the matrix  $W_\eta$  can be derived, which transforms the time-derivative of the rotation state  $\dot{\boldsymbol{\eta}}$  into the angular velocity vector  $\boldsymbol{\Omega}$ :

$$\begin{pmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & s_\theta \\ 0 & c_\phi & -c_\theta s_\phi \\ 0 & s_\phi & c_\theta c_\phi \end{pmatrix}}_{\equiv W_\eta} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (10)$$

$$\boldsymbol{\Omega} = W_\eta \dot{\boldsymbol{\eta}} \quad (11)$$

## 2.4 State

Along with the Euler-angle representation, I will consider the Cartesian coordinates of the center of mass of the drone as generalized coordinates to represent translations of the body frame:

$$\boldsymbol{\xi} = (x \ y \ z)^T, \quad \dot{\boldsymbol{\xi}} = (\dot{x} \ \dot{y} \ \dot{z})^T \quad (12)$$

Thus, the following generalized coordinates together define a 6-degree-of-freedom dynamical system with the state  $\mathbf{q}$ :

$$\mathbf{q} = \begin{pmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \end{pmatrix} \quad (13)$$

### 3 Dynamics

#### 3.1 Lagrangian

The Lagrangian  $L$  is defined as:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T - V \quad (14)$$

This system is nearly identical to rigid body free rotation; however the Lagrangian has nonzero potential energy since the quadcopter is subject to gravity. As we have shown in class, the kinetic energy in this case is the same as in free rigid body rotation:

$$T = \underbrace{\frac{1}{2}\dot{\boldsymbol{\eta}}^T I \dot{\boldsymbol{\eta}}}_{\text{Rotational kinetic energy}} + \underbrace{\frac{1}{2}m\dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}}}_{\text{Translational kinetic energy}} \quad (15)$$

$$= \frac{1}{2}\dot{\boldsymbol{\eta}}^T \underbrace{W_{\boldsymbol{\eta}}^T I W_{\boldsymbol{\eta}}}_{\equiv J(\boldsymbol{\eta})} \dot{\boldsymbol{\eta}} + \frac{1}{2}m\dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} \quad (16)$$

$$= \frac{1}{2}\dot{\boldsymbol{\eta}}^T J(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} + \frac{1}{2}m\dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} \quad (17)$$

And the potential energy can be determined from these coordinates in the standard way using the gravitational potential:

$$V = mgz = mg(\boldsymbol{\xi} \cdot \bar{\mathbf{e}}_3) \quad (18)$$

Thus, the entire Lagrangian is given by:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\boldsymbol{\eta}}^T J(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} + \frac{1}{2}m\dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} - mg(\boldsymbol{\xi} \cdot \bar{\mathbf{e}}_3) \quad (19)$$

#### 3.2 Translations

Since the equation is linear, we can split the Lagrangian into a sum of two functions, i.e.  $L(\mathbf{q}, \dot{\mathbf{q}}) = L(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) + L(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$  and consider them separately. First, the derivatives of  $L(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$ , which are relatively straightforward to derive:

$$\frac{\partial L}{\partial \boldsymbol{\xi}} = -mg\bar{\mathbf{e}}_3 = \begin{pmatrix} 0 & 0 & -mg \end{pmatrix}^T \quad (20)$$

$$\frac{\partial L}{\partial \dot{\boldsymbol{\xi}}} = m\dot{\boldsymbol{\xi}}, \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{\xi}}} = m\ddot{\boldsymbol{\xi}} \quad (21)$$

These results lead to the following equation of motion for the translation dynamics of the quadcopter using the Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{\xi}}} - \frac{\partial L}{\partial \boldsymbol{\xi}} = \mathbf{f} \quad (22)$$

$$m\ddot{\boldsymbol{\xi}} + mg\bar{\mathbf{e}}_3 = \mathbf{f} \quad (23)$$

$$\ddot{\boldsymbol{\xi}} = \frac{1}{m} (\mathbf{f} - mg\bar{\mathbf{e}}_3) \quad (24)$$

### 3.3 Rotations

The derivatives of  $L(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ , however, are more difficult to determine as they are higher-dimensional:

$$\frac{\partial L}{\partial \boldsymbol{\eta}} = \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\eta}} (\dot{\boldsymbol{\eta}}^T J) \dot{\boldsymbol{\eta}} \quad (25)$$

$$\begin{aligned} \frac{\partial L}{\partial \dot{\boldsymbol{\eta}}} &= \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T J \dot{\boldsymbol{\eta}}) = \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (J \dot{\boldsymbol{\eta}} \cdot \dot{\boldsymbol{\eta}}) \\ &= \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} \sum_i (J_{i*} \eta_i) \eta_i = \frac{1}{2} \sum_i J_{i*} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} \eta_i^2 \\ &= \sum_i J_{i*} \eta_i = J \dot{\boldsymbol{\eta}} \end{aligned} \quad (26)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{\eta}}} = \dot{J} \dot{\boldsymbol{\eta}} + J \ddot{\boldsymbol{\eta}} \quad (27)$$

Equation 25 is the most complicated expression; it is a 3x3 Jacobian matrix, the derivative of a vector  $\dot{\boldsymbol{\eta}}^T J$  with respect to a vector  $\boldsymbol{\eta}$ . What makes this expression difficult to compute, however, are the large amount of terms that come out of the derivative given that  $J = W_{\boldsymbol{\eta}}^T I W_{\boldsymbol{\eta}}$ . A full definition of this variable is given in the appendix section 8.1. Equation 26, however, is more like a normal derivative; the product rule can be used to split the derivative into a sum of dot products and then computed using a summation, eventually leading to the definition of matrix multiplication which yields  $J \dot{\boldsymbol{\eta}}$ .

The equation of rotational motion can also be determined using the Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{\eta}}} - \frac{\partial L}{\partial \boldsymbol{\eta}} = \boldsymbol{\tau} \quad (28)$$

$$\dot{J} \dot{\boldsymbol{\eta}} + J \ddot{\boldsymbol{\eta}} + \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T J) \dot{\boldsymbol{\eta}} = \boldsymbol{\tau} \quad (29)$$

$$J \ddot{\boldsymbol{\eta}} + \underbrace{\left( \dot{J} + \frac{1}{2} \frac{\partial}{\partial \dot{\boldsymbol{\eta}}} (\dot{\boldsymbol{\eta}}^T J) \right)}_{\equiv C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})} \dot{\boldsymbol{\eta}} = \boldsymbol{\tau} \quad (30)$$

$$J \ddot{\boldsymbol{\eta}} + C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \dot{\boldsymbol{\eta}} = \boldsymbol{\tau} \quad (31)$$

$$\ddot{\boldsymbol{\eta}} = J^{-1} (\boldsymbol{\tau} - C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \dot{\boldsymbol{\eta}}) \quad (32)$$

$$(33)$$

Here, the Jacobian matrix previously discussed as well as the time-derivative of  $J$  are combined into the Coriolis matrix  $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$ . The complete equations of motion of the quadrotor are thereby given by the following system:

$$\begin{cases} \ddot{\boldsymbol{\xi}} &= \frac{1}{m} (\mathbf{f} - mg\bar{\mathbf{e}}_3) \\ \ddot{\boldsymbol{\eta}} &= J^{-1} (\boldsymbol{\tau} - C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}}) \end{cases} \quad (34)$$



## 4 Controls

In order to better test the simulation of the dynamics, it will be useful to first derive a simple PID controller to stabilize the drone horizontally, i.e.,  $\boldsymbol{\eta}, \dot{\boldsymbol{\eta}} \rightarrow 0$ . To do this, I will consider a controller  $\mathbf{u}$  which takes  $\dot{\boldsymbol{\eta}}$  as an input and outputs the angular velocity of each propeller  $\omega_i$ . The controller is defined as follows [3]:

$$\mathbf{e} = \boldsymbol{\eta}^c - \boldsymbol{\eta} = \boldsymbol{\eta} \quad (35)$$

$$\mathbf{u} = k_p \mathbf{e} + k_i \int_0^t \mathbf{e} dt + k_d \frac{d}{dt} \mathbf{e} \quad (36)$$

$$= k_p \boldsymbol{\eta} + k_i \int_0^t \boldsymbol{\eta} dt + k_d \dot{\boldsymbol{\eta}} \quad (37)$$

The controller generates a torque about the center of mass of the quadcopter, which is described in Equation 2 (Let  $\gamma_i = \omega_i^2$ ). If we ignore the Coriolis matrix (which is not an unreasonable approximation; it is composed of nearly all higher-order terms) then we have the following equation for the controller:

$$\begin{pmatrix} kl(\gamma_1 - \gamma_3) \\ kl(\gamma_2 - \gamma_4) \\ b(\gamma_1 - \gamma_2 + \gamma_3 - \gamma_4) \end{pmatrix} = J\mathbf{u} \quad (38)$$

However, this is a system of 4 unknowns and only 3 equations. If we further constrain the sum of generated forces to keep the quadrotor hovering, i.e.,  $\mathbf{f}_3 = mg\bar{\mathbf{e}}_3$ , then we have an additional constraint. Note, in order to generate this force in the inertial frame, the drone must generate a force  $T = \frac{mg}{c_\theta c_\phi}$  in the body frame. This leads to the following system of equations (Let  $J\mathbf{u} = \mathbf{j}$ ,  $\mathbf{j}_i = j_i$ ):

$$\begin{pmatrix} kl(\gamma_1 - \gamma_3) \\ kl(\gamma_2 - \gamma_4) \\ b(\gamma_1 - \gamma_2 + \gamma_3 - \gamma_4) \\ k(\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4) \end{pmatrix} = \begin{pmatrix} j_1 \\ j_2 \\ j_3 \\ T \end{pmatrix} \quad (39)$$

This system has the following solution:

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \frac{T}{4k} + \frac{1}{2kl} \begin{pmatrix} -j_1 \\ -j_2 \\ j_2 \\ j_1 \end{pmatrix} + \frac{1}{4b} \begin{pmatrix} -j_3 \\ j_3 \\ -j_3 \\ j_3 \end{pmatrix} \quad (40)$$

## 5 Simulations

### 5.1 Pure Dynamics

In the absence of a physical drone to carry out experiments, I will conduct a series of simulated experiments and test assumptions made about the dynamics of the system in order to validate the equations of motion. These simple results are more of a sanity check than anything, to ensure that the simplest quadcopter maneuvers are accurately captured by the dynamics. Each trajectory is represented by four “traces” showing the path of each rotor, which starts at a blue color and gradually becomes pink as time progresses.

The first assumption is that applying a net upward force slightly greater than the weight of the quadcopter split across each rotor (with zero torque) should lead to a straight takeoff. Figure 3 supports this hypothesis, showing an increase in the  $z$  coordinate that grows steeper over time.

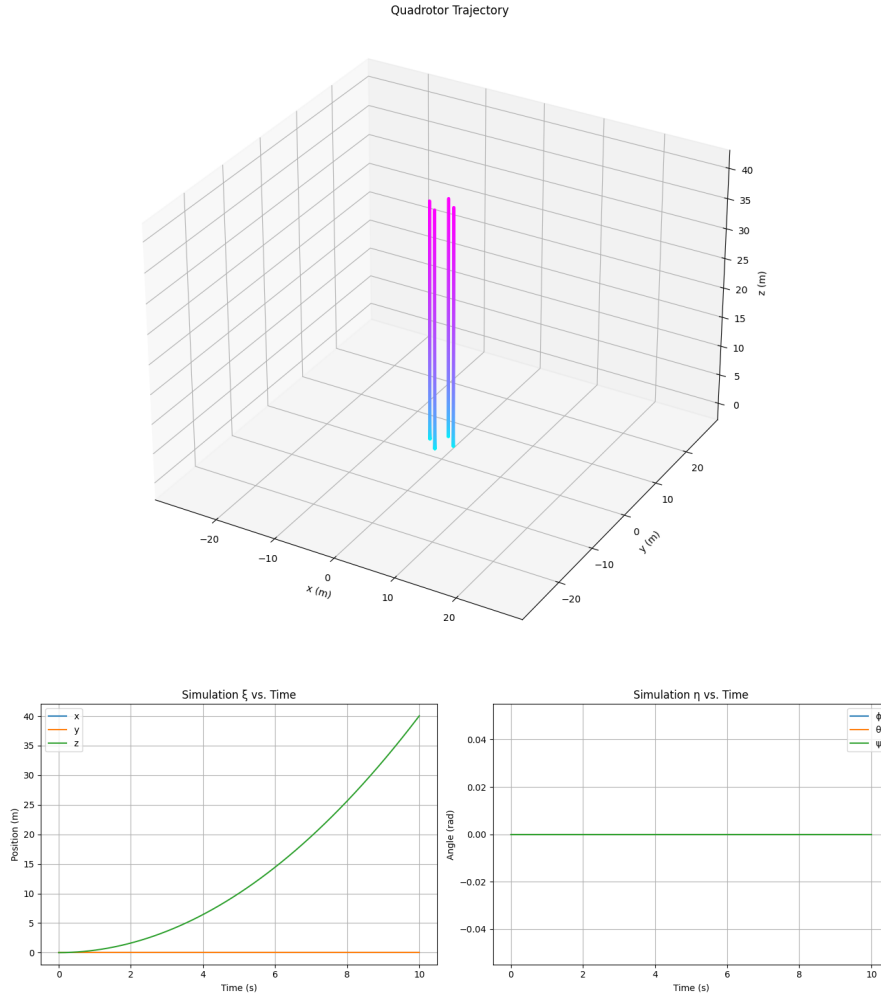


Figure 3: Straight Takeoff

In the next experiment, I will simulate a similar takeoff condition, except with an initial angular velocity imposed on the quadcopter, such that the trajectory immediately becomes unstable. These conditions test the limits of the numerical integration technique, however, with enough precision it is possible to determine the flight path of the drone and its wild oscillations in this state. The results of this experiment also show the instability that nearly all modes of quadrotor flight exhibit. Once tumbling in such a manner, it is difficult even for a well-tuned controller to stabilize the drone.

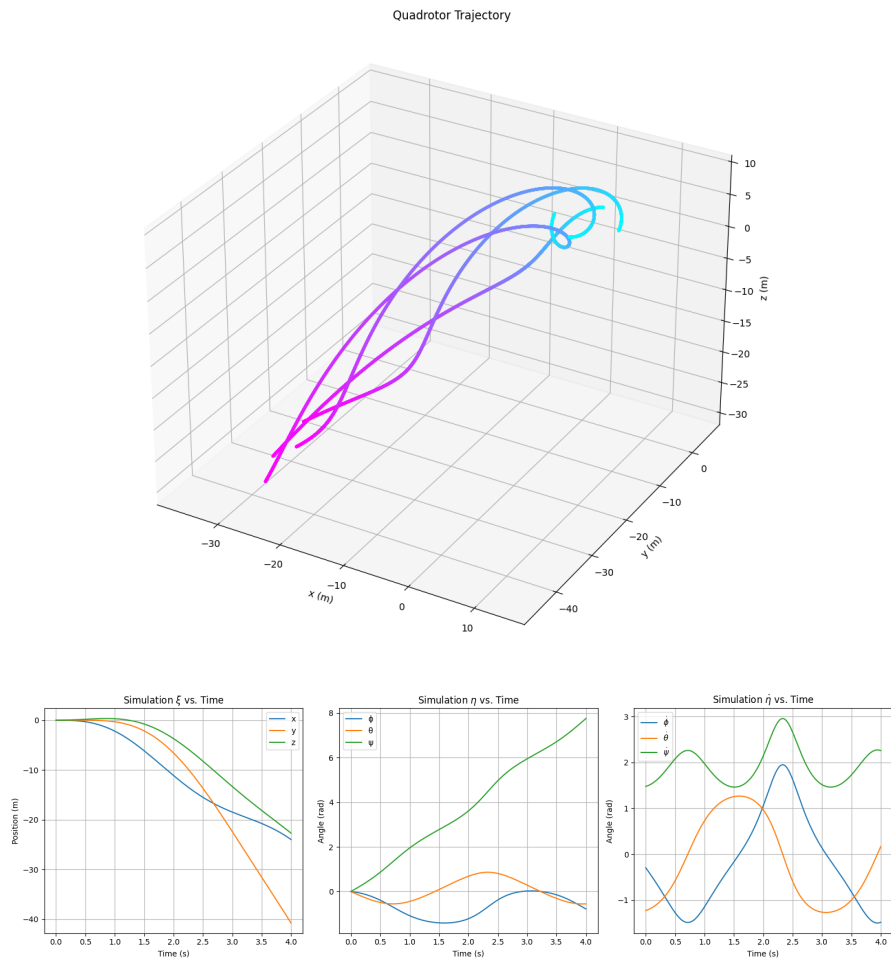


Figure 4: Tumble

Again, these results are not meant to provide a useful example of the quadrotor dynamics, but rather to display empirical results of the derived equations of motion and to verify that simple expectations about the flight of a quadcopter can be reproduced in the simulation. In the next section, I will present experiments carried out with an actually useful controller.

## 5.2 Simulated Control

One of the simplest tests to carry out using the controller described in Section 4 is a step response, in which the quadcopter starts out with a nonzero initial angular position and velocity and uses the controller to drive this error to zero. In this experiment, the initial angular velocity used was twice the amount of the angular velocity used in the “Tumble” experiment (Figure 4).

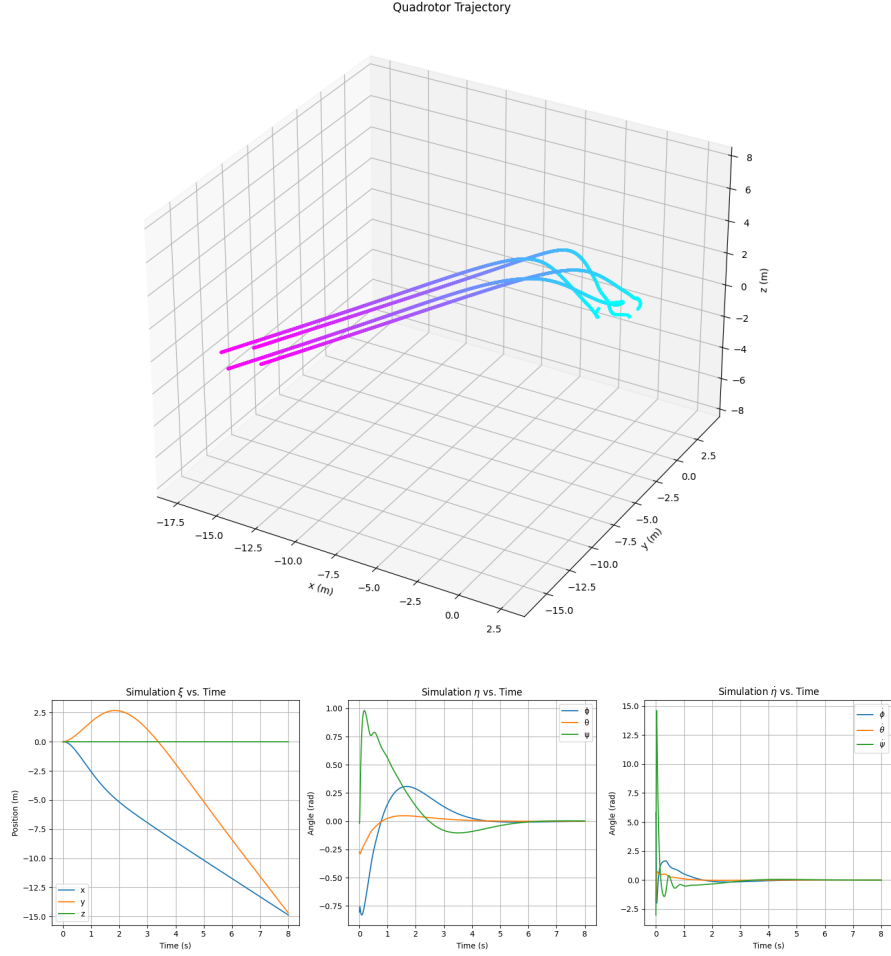


Figure 5: Step Response

In this response, note that the quadcopter was able to also stay steady in the  $z$ -axis; this is because the controller is designed to always produce an upwards thrust equal to the quadcopter’s weight, regardless of its orientation. Within 5 seconds of this simulation, the quadcopter is able to drive both its angular velocity and position to zero as hypothesized. Despite the efforts of the controller, however, there is still some “drift” during the maneuver. This is to be expected, since the controller does not depend on the  $x$  or  $y$  position and only depends on the rotation.

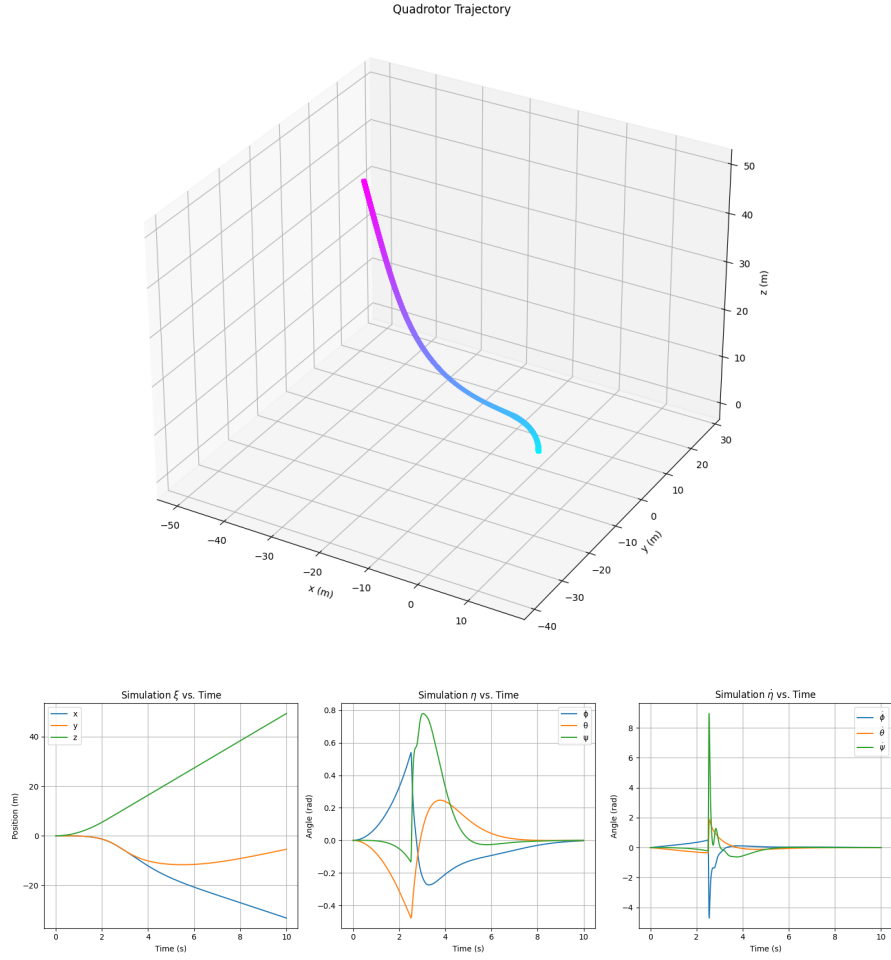


Figure 6: Takeoff Recovery

For the final test, I simulate a “faulty takeoff,” (or simply, a delayed step response) in which two neighboring propellers are jammed and produce less thrust than the opposite two, leading to a tilt and imminent tumble. A quarter of the way through the simulation, however, the stabilizing controller turns on and is able to quickly correct the misalignment.

## 6 Discussion

In this project I have shown how to derive the equations of motion of a quadcopter using the Lagrangian formulation of dynamics and verified the result using numerical methods. Additionally, I have implemented a simple PID control system to stabilize the drone horizontally and likewise verified its effectiveness in simulations. While the dynamics of a quadcopter can be derived in varying levels of complexity, this work shows that even a simple approach using the Euler-Lagrange equations can yield substantial results when combined with numerical simulation, and although the derived equations are simple enough, they describe an extremely detailed evolution of the physical system that can be used for many purposes.

Some directions for future work involve more complex methods of representing the rotations of a rigid body in 3D, for example, quaternions, which don't suffer from gimbal lock, and of course more effective controllers which can be implemented using these parametrizations [2]. Also, there are countless specific applications for quadcopters which can be studied more directly from the perspective of the dynamics, for example in FPV drone racing where it is necessary to maximize the speed of the drone to gain a competitive edge, or in surveillance where sound generated by the rotors could be a main concern.

Another course of interest could be to focus on the numerical simulation itself, as numerical methods are key to realizing the power of any dynamics derivation, and there are more advanced integration methods (for example, Runge-Kutta) than the naive Euler integration used in this project, which could potentially improve the realism of the simulated results as well as prevent divergence when functions are not smooth.

## 7 Bibliography

- [1] R. Beard. Quadrotor dynamics and control rev 0.1. 2008.
- [2] E. Fresk and G. Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *2013 European control conference (ECC)*, pages 3864–3869. IEEE, 2013.
- [3] A. Gibiansky. Andrew gibiansky:: Math→[code]. *Quadcopter Dynamics and Simulation*. *Andrew Gibiansky*, 23, 2012.
- [4] T. Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22:22, 2011.
- [5] S. Nakano. *Quad-Rotor Dynamical Model with Euler-Lagrange Approach*. PhD thesis, Tokyo Institute of Technology Tokyo, 2013.

## 8 Appendix

### 8.1 Coriolis Matrix

The entire Coriolis Matrix  $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$  is given by the following expression [5]:

$$C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{31} & c_{33} \end{bmatrix}$$

$$\begin{aligned} c_{11} &= 0 \\ c_{12} &= (I_{yy} - I_{zz})(\dot{\theta}C\phi S\phi + \dot{\psi}S^2\phi C\theta) + (I_{zz} - I_{yy})\dot{\psi}C^2\phi C\theta - I_{xx}\dot{\psi}C\theta \\ c_{13} &= (I_{zz} - I_{yy})\dot{\psi}C\phi S\phi C^2\theta \\ c_{21} &= (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi + \dot{\psi}S^2\phi C\theta) + (I_{yy} - I_{zz})\dot{\psi}C^2\phi C\theta + I_{xx}\dot{\psi}C\theta \\ c_{22} &= (I_{zz} - I_{yy})\dot{\phi}C\phi S\phi \\ c_{23} &= -I_{xx}\dot{\psi}S\theta C\theta + I_{yy}\dot{\psi}S^2\phi C\theta S\theta + I_{zz}\dot{\psi}C^2\phi S\theta C\theta \\ c_{31} &= (I_{yy} - I_{zz})\dot{\psi}C\phi S\phi C^2\theta - I_{xx}\dot{\theta}C\theta \\ c_{32} &= (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi S\theta + \dot{\phi}S^2\phi C\theta) + (I_{yy} - I_{zz})\dot{\phi}C^2\phi C\theta \\ &\quad + I_{xx}\dot{\psi}C\theta S\theta - I_{yy}\dot{\psi}S^2\phi C\theta S\theta - I_{zz}\dot{\psi}C^2\phi C\theta S\theta \\ c_{33} &= (I_{yy} - I_{zz})\dot{\phi}C\phi S\phi C^2\theta - I_{yy}\dot{\theta}S^2\phi C\theta S\theta - I_{zz}\dot{\theta}C^2\phi C\theta S\theta + I_{xx}\dot{\theta}C\theta S\theta \end{aligned}$$

Figure 7: Coriolis Matrix