2 - Equations of Motion

The purpose of this assignment is to determine how the Parrot Mambo UAV flies. To this purpose, we are calculating the lift and power generated by its propellers, the yaw, roll and pitching moments created, the gyroscopic moments and trim speed of the propellers. Then, we will translate the given equations of motion of the UAV to the coordinate frame of a mounted cannon.

1 Assumptions and constants

To derive the equations of motion for the UAV, we made a number of assumptions; a list of some of the major ones follows:

1. Aerodynamic Forces are negligible; at the low speeds of the UAV, we assume that aerodynamic forces are insignificant since they scale with the velocity and velocity squared of the UAV. This is done to simplify equations of motion and reduce the number of nonlinear terms in our equations. As a part of this assumption, we assume that there is no wind speed since the UAV will be flone mostly indoors. The main two locations this assumption breaks down are close to the ground, where the ground effect becomes significant and at the propellers, which can spin up to nearly mach 0.3.

2. The UAV is a uniform mass block with equal X and Y dimensions; This assumption is clearly not true, but it allows us to simplify the moment of inertia matrix J to only the diagonal terms, as well as make it easier to calculate the diagonal terms. This assumption breaks down when considering the mounted cannon, but we will still use it for the sake of this assignment. This assumption will be refined later in the course.

3. The coefficient of thrust curve, $C_T(n)$, is linear relative to the propeller's rotational velocity, and the coefficient of power curve, $C_P(n)$, is constant relative to the propeller's speed. This assumption is based on a linear fit from propeller data collected within the range of 180 and 480 Hz propeller speed [1]. This assumption also includes the sub-assumption that all propellers and motors are equivalent and produce the exact same thrust and power for the same rotational speed.

4. We assume the following constants and values shown in the matlab script:

Listing 1: Constants and Assumptions

```
% PROP DATA: https://commons.erau.edu/cgi/viewcontent.cgi?article=2057&context=publication
RPMs = [11000 19000 23000 25000 27000 29000];
C_thrust = [0.079 0.083 0.088 0.09 0.092 0.093]; %C_T vs RPMs
C_power = [0.041 0.041 0.041 0.041 0.041 0.041]; %C_P vs RPMs
prop_dia = 0.066; %meters ; prop diameter

% UAV DATA: parrot.com/global/drones/parrot-mambo-fly
UAV_mass = 0.068; %kg ; total mass of UAV
prop_dist = sqrt(2)*0.09; %meters ; distance from CM to prop

prop_mass = 0.001; %kg (1 gram) ; assumed mass of prop

% CONSTANTS
rho = 1.22495; % kg/m3 ; density of air, given in presentation
ns = RPMs/60; %Hz ; converts RPMs to Hz
g = 9.81; %m/s2 ; acceleration due to gravity
```

2 Propeller Data and Analysis

In this section we will calculate the moments created by the propellers, and their lift. First, I took the propeller data from [1] created a linear fit:
**The Matlab code will be truncated to show the calculations, cutting some of the code to produce the plots for ease of reading. The full code will be provided attached.*

Listing 2: Fitting Propeller Data with Linear Approximation

```
% used for linear regression of prop data
X = [ones(1,length(ns))' ns'];

%coefficients in linear fit of coef thrust/power curves
C_thrust_lin = X\C_thrust';
C_power_lin = X\C_power';

%eq'n for coef of thrust/power
C_thrust_eq = @(C_T, n) [(C_T(1)+C_T(2)*n)];
C_power_eq = @(C_P, n) [(C_P(1)+C_P(2)*n)];

%to plot coefficient of thrust/power curves
n = [0:10:500]; %Hz
C_thrust_curve = C_thrust_eq(C_thrust_lin, n);
C_power_curve = C_power_eq(C_power_lin, n);

figure;
subplot(2,1,1); %sublpot 1.1: C_T vs Hz
plot(ns, C_thrust, 'bo');
hold on;
plot(n, C_thrust_curve);

subplot(2,1,2); %subplot 1.2: C_P vs Hz
plot(ns, C_power, 'bo');
hold on;
plot(n, C_power_curve);
```

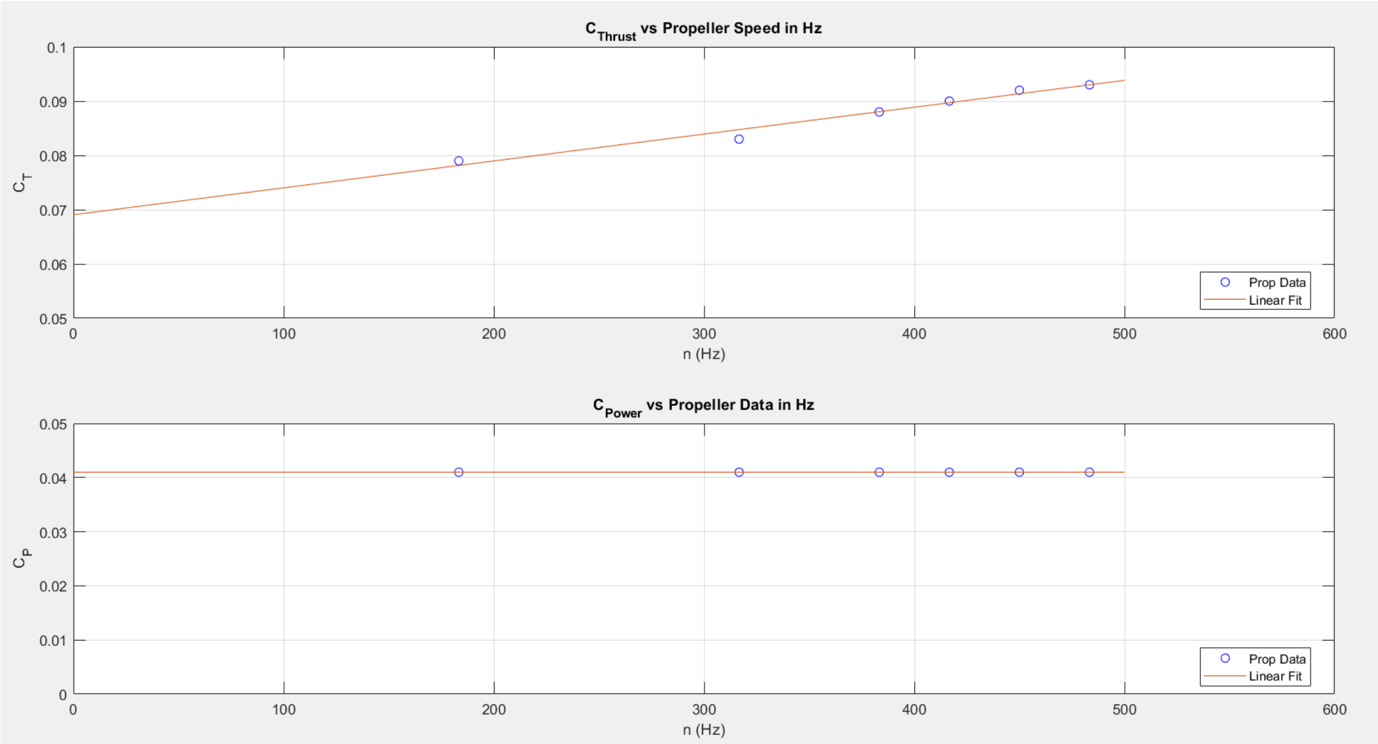

Figure 1: C_T and C_P Curves vs Prop Speed in Hz, with original prop data and linear fit

The important part to note with the graphs in Figure 1 is that, $C_T(n)$ is linear, so any equation that uses it gains an extra order polynomial, while $C_P(n)$ is relatively constant over the range of data. We use these values in our equations for thrust and power,

$$T(n) = C_T(n)\rho n^2 D^4 \tag{1}$$
$$P(n) = C_P(n)\rho n^3 D^5, \tag{2}$$

where ρ is the density of air and D is the propeller's diameter.

To calculate the Yawing Moment, we use the power, and the relationship,

$$P = \omega\tau \tag{3}$$
$$\omega = 2\pi n \tag{4}$$

where ω is the angular velocity and τ is the applied torque, or moment. We solve these equations for τ to get the yawing moment of the propeller,

$$\tau = \frac{P}{2\pi n}. \tag{5}$$

An important note here is the fact that the direction of yaw moment on the $\pm z$-axis is related to whether the propeller is spinning clockwise or counter-clockwise. A clockwise-spinning propeller has a positive angular, since the $+z$-axis is pointing down, this will cause it to create a negative yaw moment, since the drag it experiences creates an equal and opposite torque from its direction of travel. Likewise, the counter-clockwise propellers will have positive moments.

Because of the symmetry of the UAV, the roll and pitching moments are calculated from the same equation. The moments are created when a propeller generates a thrust, which is a distance d_{prop} away from the center of mass of the UAV. because the generated thrust, T, is always along the z axis of the UAV, which we can assume to be perpendicular to the distance vector from the center of mass to the propeller, the generated torque will just be

$$\tau = T d_{prop}. \tag{6}$$

The direction of the torque will be half in the x, and half in the y, with plus and minus signs on each related to which propeller is being referenced, because they are all 45^o from the x or y axis of the UAV. THe Matlab code used to calculate these is shown below.

Listing 3: Calculating and Plotting Moments

```
%define eq'ns for prop power and thrust ; from class presentation
prop_power_eq =  @(C_P, n) [C_power_eq(C_P,n).*(n.^3)*rho*(prop_dia^5)];
prop_thrust_eq = @(C_T, n) [C_thrust_eq(C_T,n).*(n.^2)*rho*(prop_dia^4)];

%calculate curve to plot
prop_yaw_moment_n = prop_power_eq(C_power_lin,n)./(n*2*pi);

%calculate pitch/roll moment as thrust(force)*dist from CM
prop_pitch_moment_n = prop_thrust_eq(C_thrust_lin,n).*prop_dist;

figure; %figure 2: Yaw and Pitch/Roll moments vs Prop speed
subplot(2,1,1); %subplot 2.1: Yaw moment in N-m vs prop speed in Hz
plot(n,prop_yaw_moment_n);

subplot(2,1,2); %subplot 2.2: pitch/roll moments in N-m vs prop speed in Hz
plot(n,prop_pitch_moment_n);
```

In this section of code, and in Listing 2, I used anonymous functions to define equations rather than arrays for flexibility. This ended up being to my benefit later on when I was able to use Matlab's solve function to find the trim speed of the propellers.

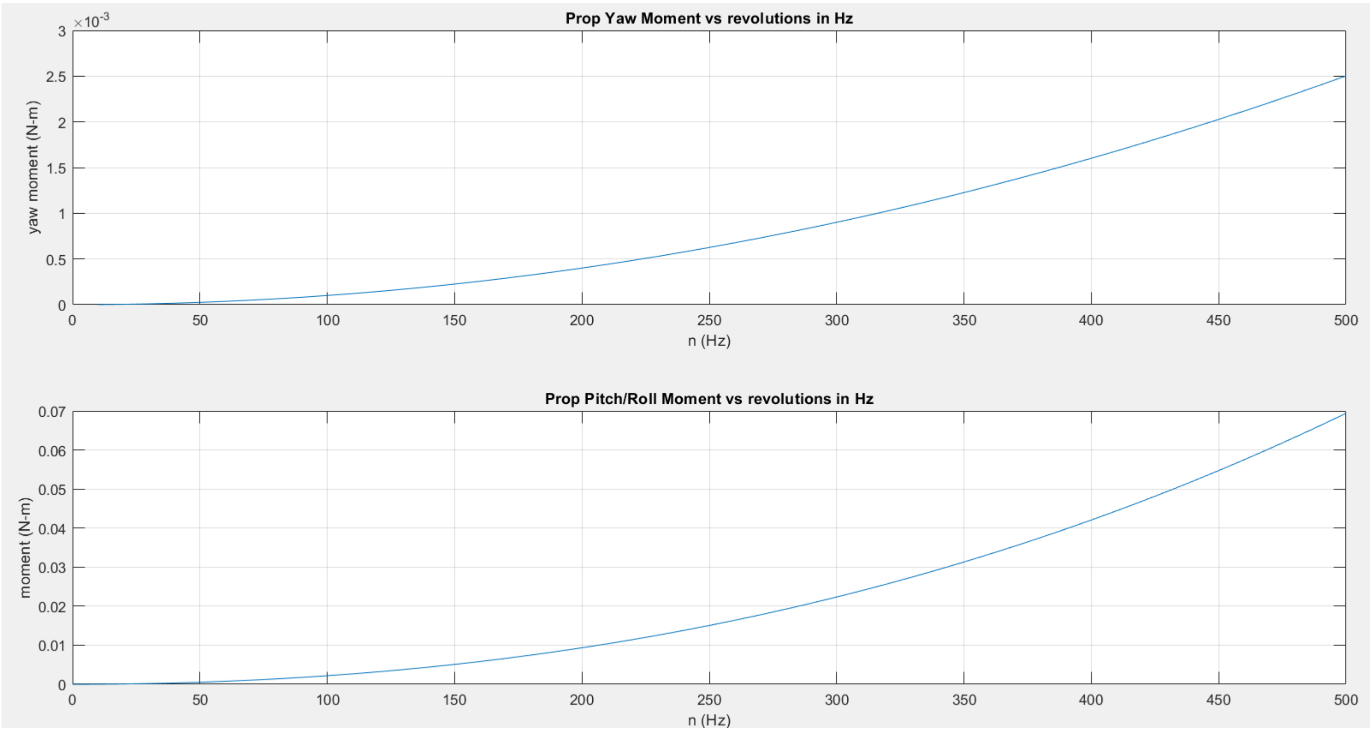

Figure 2: Propulsive Moments Generated by the Propellers

Next, we calculated the mass moment of inertia of the propellers, in their own frames and in the UAV frame. We start with two assumptions: first, that the propellers have a mass of 1 gram, which we made because there was not reliable specs on the propeller; and second, that the propeller is a 1D rod, which makes calculating its moment of inertia much simpler. The first assumption can be later updated once reliable measurements have been made, and the second is reasonable since most of the mass is distributed similarly to an imaginary 1D rod.

To calculate the moment of inertia of the rod, I_rod, we solve the integral definition of moment of inertia,

$$I = \int r^2 dm, \tag{7}$$

where r is the distance from the center of mass, and dm is the infinitesimal mass. For a rod, we integrate along its length, $-L/2$ to $L/2$, and have $dm = \frac{M}{L}dr$:

$$I_{rod} = \frac{M}{L}\int_{-L/2}^{L/2} r^2 dr \tag{8}$$

$$I_{rod} = \frac{1}{12}L^2 M. \tag{9}$$

Translating this to our propeller, let L be the diamater of the propeller, D, and M be the mass of the propeller.

We must then translate this to the center of mass of the UAV. We can do this using the Parallel Axis Theorem:

$$I_b = I_{cm} + md_{cm,b}^2, \tag{10}$$

describing the moment of inertia at some point, b, given the moment of inertia at the center of mass, cm, and the distance between the two points.

In our case, we use I_{cm} as the moment of inertia of the propeller, from the center of mass of the propeller, and b to be the center of mass of the UAV, to calculate the moment of inertia of the propeller from the axis of the UAV's center of mass:

$$I_{prop,UAV} = I_{prop} + m_{prop}d_{prop,UAV}^2 \tag{11}$$

$$I_{prop,UAV} = \frac{1}{12}m_{prop}D_{prop}^2 + m_{prop}d_{prop,UAV}^2 \tag{12}$$

$$\tag{13}$$

This value is important, because it helps determine the angular momentum of the propellers from the perspective of the UAV. I calculated them in Matlab using the assumption that the mass of the propeller is 1g, the diamater is 6.6cm from [1], and the assumption that the distance from the UAV center of mass to the propeller center of mass was 25cm, half the diagonal length of the UAV from [1].

Listing 4: Propeller Mass Moment from Center of Mass and UAV CM

```
%I_rod = 1/12*L^2*M ; derived in notes
prop_mass_moment = 1/12*(prop_dia^2)*prop_mass;
% RESULT: 1.2705e-06 N-m

%parallel axis thrm to translate it to UAV CM:
prop_mass_moment_uav = prop_mass_moment + prop_mass*(prop_dist^2);
% RESULT: 5.79705e-05 N-m
```

.

Next, we calculated the speed to trim the UAV at $1g$. To do this, we used the thrust curve shown in Eq 1, and calculated in Listing 3, line 3; the total thrust generated by the UAV is $4 * T(n)$. We then set this equal to the weight of the UAV, mg, and solve for the necessary speed of the rotors:

$$4T(n) = mg \tag{14}$$

We used Matlab's solver function to solve this equation for its positive values, and checked that the value was within the operating range of the propellers.

Listing 5: Calculating Trim Speed Propellers for UAV

```
%want to set total thrust to be 1g*UAV_mass Newtons
syms x
n_trim_speed = vpa(solve([4*prop_thrust_eq(C_thrust_lin,x)==g*UAV_mass; x>1]))
% RESULT: n = 293 Hz
```

.

I then extended this to plot the vertical acceleration of the UAV for any given propeller speed, n:

$$a_{-z} = 4T(n)/m - g. \tag{15}$$

This plots acceleration in the $-z$ axis, taking into account gravity. This assumption is based on the UAV being stable and having no pitch and roll angles of 0.

Listing 6: Plotting Acceleration of UAV vs Prop Speed

```
%acceleration of UAV from 4 props and gravity at n Hz
vertical_acceleration_curve = 4*prop_thrust_eq(C_thrust_lin,n)/(UAV_mass)-g;

figure; %figure 3: UAV acceleration curve (g's) vs prop speed
plot(n,vertical_acceleration_curve);
xlabel('n (Hz)');
ylabel('UAV acceleration (m/s^2)');
title('UAV Vertical Acceleration ((thrust/mass) - g) vs Prop Speed');
grid on;
```

.

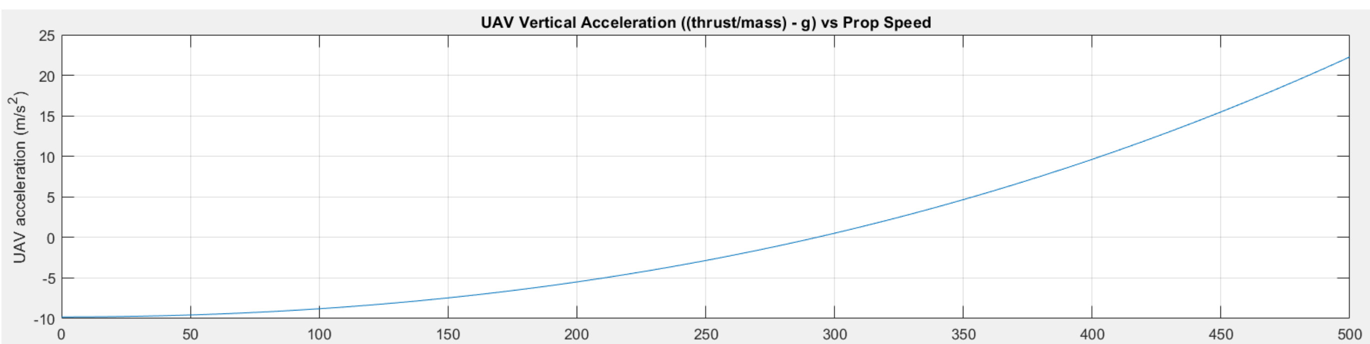

Figure 3: Vertical Acceleration of UAV in $-z$ axis vs Prop Speed

While this graph plots down to 0 Hz, it is likely not accurate or useful outside of the 180 to 480 Hz range of the propeller data. This is for two reasons: first, the linear fit of C_T likely will be inaccurate at that point, and second, the propellers are unlikely to be spinning that slowly while the drone is in flight.

3 Gyroscopic Moments of the Propellers

Another aspect of having propellers is that gyroscopic moments are generated when the UAV rotates. This comes from the equation

$$\vec{M}_{gyro} = \vec{H} \times \vec{\alpha} I \tag{16}$$

where H is the angular momentum, $alpha$ is the angular acceleration and I is the moment of inertia.This has the effect that when a torque $\tau = \alpha I$ is applied to the UAV, or any other object with angular momentum, a moment is generated perpendicular to the current angular momentum and the torque as defined by a cross product. *NOTE: I couldn't find a source defining this equation exactly, so my analysis is valid for the directions of the moments, but may not be for the magnitudes.

For example, if someone were to apply a torque to the UAV while it is at trim to spin it around the $+x$ axis, the propellers spin in the $\pm z$ axis, so the generated moment will be in the $\pm y$ axis:

$$\hat{z} \times \hat{x} = \hat{y} \tag{17}$$

For clockwise spinning propellers, who have positive angular momentum, will create a positive torque in the $+\hat{y}$, and vice versa. Figure 4 shows this relationship.

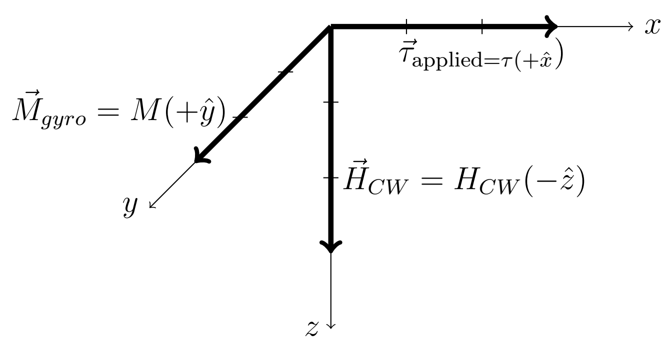

Figure 4: Direction of Gyroscopic Moments for propellers *graph modified from [3]

This analysis is considering only a singular propeller. If we want to analyze the gyroscopic moments generated by all four propellers, we can do this two ways: first, we can consider the net angular momentum of the UAV, or we can consider each propeller individually. The first approach is easier if we assume that all propellers are spinning at the same speed at trim. This assumption, along with the fact that two propellers are spinning clockwise and two are counter-clockwise, results in the assumption that the net angular momentum of the UAV is $\vec{0}$ at trim. Alternatively, if we consider each propeller separately, we get equal opposing moments from the clockwise and counter-clockwise propellers to arrive at the same result.

In practice, I believe a hybrid of these two methods will be best. Since the propellers are approximately equal, we can just add the net angular velocity of all four propellers and multiply by their mass moment of inertia to get their contribution to the angular momentum of the UAV:

$$H = 2\pi(\sum_{i=1}^{4}(-1)^{i+1}n_i)I_{prop} \tag{18}$$

in the $\pm z$ axis, where the $(-1)^{i+1}$ term is used to determine if the propeller is spinning clockwise or counter-clockwise.

4 Translating the Equations of Motion

In this section we will define the changes necessary to translate the equations of motion of the UAV to the coordinate frame of the tip of the cannon, with the x axis being the direction of its barrel as shown in Figure 5. The simplified equations of motion of the UAV are defined below, as given in this assignment:

$$m\dot{V} + \omega \times mV = \vec{F}_G(\phi, \theta, \psi) + \vec{F}_T \tag{19}$$

$$J\dot{\omega} + \omega \times J\omega = \vec{M}_T + \vec{M}_{gyro} \tag{20}$$

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{21}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{22}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\ phi}{\cos\theta} \tag{23}$$

$$\begin{bmatrix} \dot{X}_{NED} \\ \dot{Y}_{NED} \\ \dot{Z}_{NED} \end{bmatrix} = R^T(\psi)R^T(\theta)R^T(\phi)\begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{24}$$

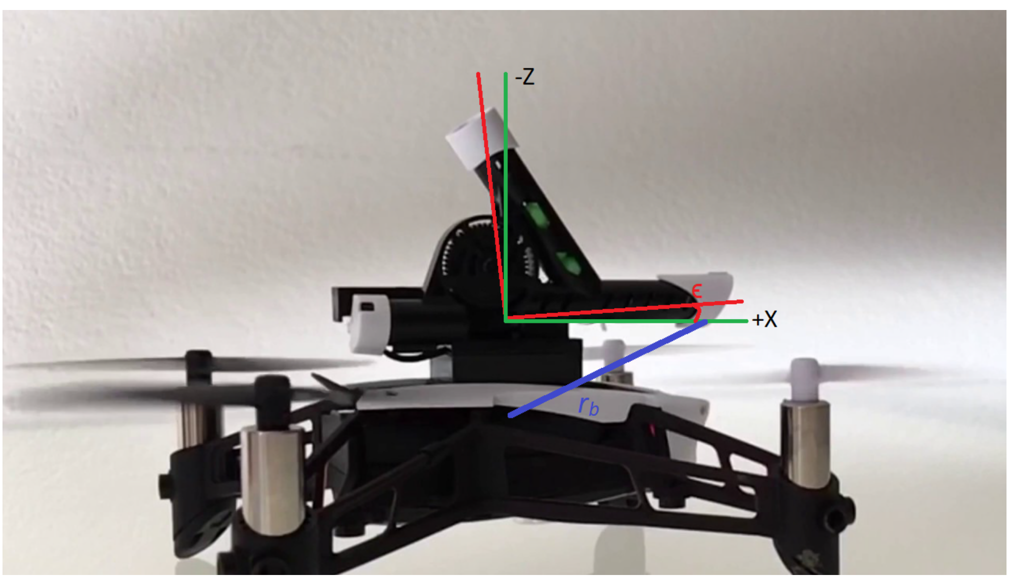

Figure 5: Translations to Frame of the Tip of the Cannon

To make this transformation, we consider the differences between the two coordinate frames: the ϕ angle of the cannon is increased by ϵ, and the cannon is offset from the UAV's center of mass by a distance r_b in the $+x$ and $-z$ direction. For the angle change, we simply define the ϕ angle to be

$$\phi_{\text{cannon}} = \phi_{\text{UAV}} + \epsilon, \tag{25}$$

and for the offset, we must use the Parallel Axis Theorem in three dimensions [4] to find the moment of inertia of the UAV relative to the cannon's position,

$$\begin{align} J_{X\text{cannon}} &= J_{X_{\text{UAV}}} + m_{\text{UAV}}(r_z^2) \tag{26} \\ J_{Y\text{cannon}} &= J_{Y_{\text{UAV}}} + m_{\text{UAV}}(r_z^2 + r_x^2) \tag{27} \\ J_{Z\text{cannon}} &= J_{Z_{\text{UAV}}} + m_{\text{UAV}}(r_x^2) \tag{28} \\ J_{XY\text{cannon}} &= 0 \tag{29} \\ J_{YZ\text{cannon}} &= 0 \tag{30} \\ J_{ZX\text{cannon}} &= m_{\text{UAV}}(r_x r_z) \tag{31} \\ & \tag{32} \end{align}$$

with the assumptions that the UAV had no diagonal components for J from its center of mass, the offset from the cannon's tip to the UAV's center of mass is only in the x, z directions, with components r_x and r_z respectively, and the offset angle ϵ is small enough that we can assume our axes are parallel, and that the cannon itself has no mass.

Since the direction of the position offset is a static change relative to the NED position, we don't need to consider it for the dynamics, aside from in calculating J.

We can then use equation 25 to replace values of ϕ in our equations of motion, and equations 26-32 to construct and replace J to get the equations of motion relative to the cannon's tip.

The important part of this analysis is that if we want to make a constant translation of our axes, there is some constant adjustment to our equations of motion that we need to can make, and they will hold for the new translated position.

Additionally, we want to be able to fire the cannon. If we assume it fires a projectile at a velocity v_b and has a mass m_b, we want to know what the projected force will be on the UAV. However, because the UAV's IMU is not continuous, we will be measuring the change in momentum of the UAV from launching the ball,

$$\Delta p = m_b v_b. \tag{33}$$

To get the approximate force, we divide the momentum by the sampling interval, ΔT:

$$F = \frac{m_b v_b}{\Delta T} \tag{34}$$

References

[1] https://commons.erau.edu/cgi/viewcontent.cgi?article=2057&context=publication

[2] http://adaptivemap.ma.psu.edu/websites/A2_moment_intergrals/parallel_axis_theorem/parallelaxis

[3] https://tex.stackexchange.com/questions/414403/drawing-vectors-on-3-d-coordinate-system

[4] https://www.real-world-physics-problems.com/parallel-axis-and-parallel-plane-theorem.html