AE352 Group Project: UAV Design

Richie Ma, Minh Nguyen, Dan Anthoney
University of Illinois, Urbana-Champaign, Illinois, 61820

I. Nomenclature

I_{xx}	=	moment of inertia with respect to the x-axis
I_{yy}	=	moment of inertia with respect to the y-axis
I_{zz}	=	moment of inertia with respect to the z-axis
a	=	cylinder diameter
ϕ	=	roll
θ	=	pitch
ψ	=	yaw
$\omega_{1,2,3,4}$	=	angular velocities of rotors
$\omega_{x,y,z}$	=	angular velocities about x, y, z in body frame
k_t	=	propeller thrust coefficient
C_D	=	coefficient of drag
L, ℓ	=	drone arm length
ρ	=	air density
D	=	propeller diameter
F_T	=	force of thrust
F_g	=	gravitational force
T_b	=	thrust on drone in body frame
T_G	=	thrust in the global reference frame
R	=	rotation matrix
$e_{x,y,z}$	=	standard basis vectors
τ	=	torque

II. Introduction

A. Project Aim

Quadcopters are a form of Unmanned Aerial Vehicle (UAV) which uses four rotors to generate lift and maneuver in the air. The vertical placement of their rotors means that the quadcopter or more commonly known as drones, are able to achieve vertical take-off and landing (VTOL) and hover at a stationary point, which most planes are unable to do due to their need to move horizontally to generate lift with their wings. The four rotors are symmetrically arranged in a cross formation, enabling these UAVs to perform with exceptional stability and agility. This configuration facilitates a range of motion and precision that is ideally suited for diverse applications, including aerial photography, videography, search and rescue operations, and even tactical missions in defense sectors.

The objective of this project is to develop a comprehensive dynamical model, often referred to as a "digital twin," of a symmetric, four-rotor drone. This model will meticulously simulate both the translational movements of the drone's center of mass and the rotational dynamics governing its attitude. The uniqueness of this drone's control mechanism is that its state and attitude can only be altered through modifications in the rotor speeds, which are, in turn, driven by changes in the motor torques. Each rotor's speed can be independently adjusted, allowing for precise maneuvering and stability control.

B. Engineering Requirements

The engineering criteria defined for this project are aimed at ensuring that the drone model is not only functional but also adheres to realistic operational standards:

1) Six Degrees of Freedom: The model will account for the drone's three translational and three rotational movements, capturing the full range of motion necessary for complex maneuvering.
2) Control Through Rotor Speeds: The drone's movement and orientation will be exclusively controlled through the adjustment of rotor speeds, influenced by the motor torques. This reflects a realistic flight control system seen in actual drones.
3) Sustainable Performance: It is imperative that the drone achieves specified performance objectives without depleting its power resources, thereby optimizing both efficiency and endurance.
4) Realistic and Implementable Design: The drone should be designed with parameters that are not only theoretically sound but also practically implementable, utilizing commercially available components. This includes considerations for mass, power consumption, motor torque, and rotor dimensions, with the entire system weighing between 0.1 kg and 10 kg.

C. Performance Goals

To validate the fidelity and accuracy of the proposed dynamical model, the drone will be tested against a series of performance goals:

1) Hover Capability: The drone must be able to hover 1 meter above the ground for a continuous period of 2 minutes.
2) Controlled Flight Path: It should demonstrate the ability to navigate in a circular path with a 2-meter radius at an altitude of 1 meter, maintaining a consistent speed of 0.5 m/s for at least 1 minute.
3) Complex Maneuvering: The drone must execute a series of maneuvers starting from a vertical takeoff to a specified altitude, followed by linear movement, a controlled yaw rotation, and a subsequent linear path, concluding with a gentle landing.

III. Quadcopter Component Selection

A. Quadcopter Body

Suppose our frame is a a four axis symmetric drone and each arm is a cylinder. The arms are 0.3 meters long each and intersect at the middle with 90 degrees. In order for the motor to be mounter properly, we set the diameter of the tubes to be 3 centimeters (0.03m). Because we were not set with a budget, we can choose to make our tubes carbon fiber.

B. Motor

The motor of choice is the T-motor MT-2216-11 motor. Below is the corresponding thrust table associated with the motor.

Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (G)	RPM	Efficiency (G/W)	Operating temperature(℃)
MT2216 KV900	11.1	T-MOTOR 10*3.3CF	50%	2.8	31	300	5200	9.68	38
			65%	3.7	42	360	5700	8.57	
			75%	4.7	53	420	6300	7.92	
			85%	6.2	69	520	6900	7.54	
			100%	7.4	81	600	7400	7.41	
		T-MOTOR 11*3.7CF	50%	3	35	350	4900	10.00	40
			65%	4.4	50	420	5400	8.40	
			75%	5.7	64	530	5900	8.28	
			85%	7.3	82	630	6500	7.68	
			100%	8.9	98	720	7000	7.35	
		T-MOTOR 12*4CF	50%	3.5	41	420	4200	10.24	43
			65%	5.8	65	560	5000	8.62	
			75%	7.8	86	680	5450	7.91	
			85%	10	110	820	5900	7.45	
			100%	12	129	920	6350	7.13	
	14.8	T-MOTOR 9*3CF	50%	3.4	52	370	7000	7.12	46
			65%	4.4	65	410	7800	6.31	
			75%	5.3	79	470	8500	5.95	
			85%	7.4	108	610	9300	5.65	
			100%	8.5	126	700	10000	5.56	
		T-MOTOR 10*3.3CF	50%	4.1	62	460	6500	7.42	50
			65%	5.6	84	570	7300	6.79	
			75%	7.1	106	690	7900	6.51	
			85%	9.5	139	830	8600	5.97	
			100%	11.2	165	940	9300	5.70	

Notes:The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.

Fig. 1 Thrust table for a MT-2216 Motor

Each of these motors weigh 75 grams, so the total weight from the four will be 300 grams (0.3 kg).

The shape of the motor can be approximated as a solid cylinder with a diameter of 27.5 mm (0.0275 m) and a height of 34 mm (0.034 m)

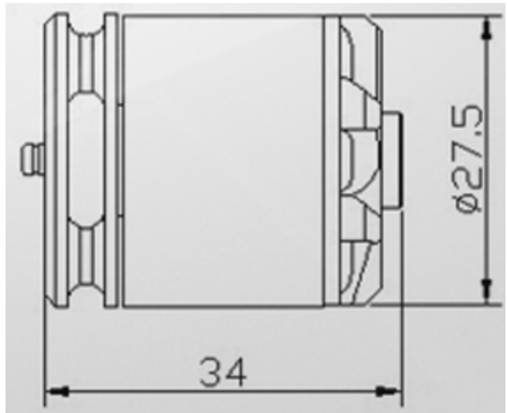

Fig. 2 Dimensions of the MT-2216 Motor

C. Battery

From what we can see in the thrust table above, a 11.1 volt battery matched with a 11*3.8 propeller will give us great thrust as well as a comparably higher efficiency. In drone terms, an 11.1 volt battery is denoted as 3s, and a 5200mAh would give us: 3 (cell) * 3.6 (V/cell) * 5.2 (A-h) = 56.16 (W-h).

One such battery that fulfills these requirements is the Lumenier 5200mAh 3s 35c Lipo Battery, which has

Length: 152 mm (0.152 m)

Width: 48 mm (0.048 m)

Height: 25 mm (0.025 m)

Weight: 392g (0.392 kg)

To simplify the model, because the carbon fiber tubes have a larger diameter than the height of th battery, the center of mass of the battery can be placed at the center of mass of the drone.

D. Propeller

The propeller that we will be using is is the APC Slow Flyer 11 X 3.8. The first number indicates the diameter in inches and the second number indicates the pitch in inches per rotation. 11 gives us an 11 inch diameter, or 0.14 meter radius. the 3.8 gives us 3.8 inch/revolution, or 0.09652 m/revolution.

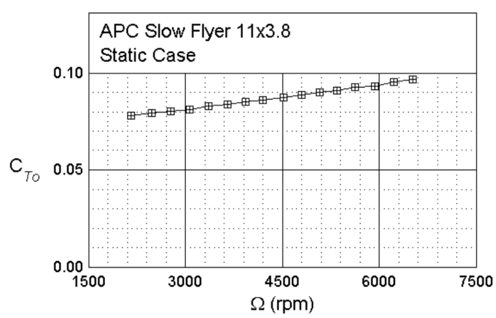

Fig. 3 Thrust Coefficient of the 11 X 3.8 Propeller

IV. Developing the Equations of Motion

A. Design Specifications

- Mass of arms: Since the material is carbon fiber, which is light and stiff, let's assume a reasonable mass based on typical values for drones of this size. We'll assume each arm has a mass of about 50 grams (0.05 kg)

- Motor positioning: Motors are mounted at the ends of each arm, and we will assume they are positioned 0.15 m (half the length of the arm) from the center.
- Battery positioning: The battery is centrally located and contributes to the inertia without additional positional calculations.
- Center of mass: We'll assume the center of mass is approximately at the geometric center of the drone due to symmetry, making calculations more straightforward

Moments of Inertia

1) Arms
   - Longitudinal inertial: $I_{longitudinal} = \frac{1}{12} \cdot 0.05 \cdot (3 \cdot (0.015)^2 + (0.3)^2) = 0.0006375\ kg \cdot m^2$
   - Radial inertia: $I_{radial} = \frac{1}{2} \cdot 0.05 \cdot (0.015)^2 = 0.000005625\ kg \cdot m^2$
2) Motors
   - Longitudinal inertial: $I_{longitudinal} = \frac{1}{2} \cdot 0.075 \cdot (0.1375)^2 = 0.000006961\ kg \cdot m^2$
   - Radial inertia: $I_{radial} = \frac{1}{12} \cdot 0.075 \cdot (3 \cdot (0.01375)^2 + (0.034)^2) = 0.000013102\ kg \cdot m^2$
3) Battery
   - $I_{xx} = \frac{1}{12} \cdot 0.392 \cdot ((0.048)^2 + (0.025)^2) = 0.00010213\ kg \cdot m^2$
   - $I_{yy} = \frac{1}{12} \cdot 0.392 \cdot ((0.152)^2 + (0.025)^2) = 0.00076161\ kg \cdot m^2$
   - $I_{zz} = \frac{1}{12} \cdot 0.392 \cdot ((0.048)^2 + (0.152)^2) = 0.00059053\ kg \cdot m^2$

Using the parallel axis theorem to account for the contributions from the motors at 0.15 meters away from the center:

$$I = I_{radial} + (0.075) * (0.15)^2$$

The final Moments of Inertia after accounting for four motors is:

- $I_{xx} = 0.00914\ kg \cdot m^2$
- $I_{yy} = 0.00982\ kg \cdot m^2$
- $I_{zz} = 0.00348\ kg \cdot m^2$

and the quadcopter has a total mass of 0.792 kg.

B. Equations of Motion: Translational

We first determine the state variables of the quadcopter. ψ, θ, ϕ are used to represent the yaw, pitch and roll of the quadcopter respectively. Their time derivatives are also represented as $\dot{\psi}, \dot{\theta}, \dot{\phi}$. The translational elements will be x, y, z components with their standard derivatives.

The force produced by the four propellers acts in the positive z-direction with the effects of gravity acting in the negative z-direction. This equates to the force equation of:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = F_T - F_g$$

The direction of the thrust force, however, is dependent on the roll, pitch, and yaw of the drone, which points the normal of the propellers in different orientations. As a result, the thrust forces changes as the orientation of the drone changes through time. The force of thrust attached to the body frame of the drone is then:

$$F_{Tb} = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$$

where T is a combination of the thrust generated by the four different sets of motors and propellers:

$$T = k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

The constant k_t is known as the thrust coefficient, which relates spin-rate of the propellers with the amount of thrust it will produce at the specific ω.

In order to convert the thrust force T_b in the fixed body frame to the thrust T_g in the global reference frame, we construct a rotation matrix to convert the vector from the body frame to the global frame. The rotation matrix is constructed as:

$$R = R_z(\psi)R_y(\theta)R_x(\phi)$$

The corresponding rotation matrices correspond to rotations about the x, y, z axes:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which are multiplied together in the order described in the equation above to give us:

$$R = \begin{bmatrix} -\sin(\phi)\sin(\psi)\cos(\theta) + \cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi) - \sin(\psi)\cos(\phi)\cos(\theta) & \sin(\psi)\sin(\theta) \\ \sin(\phi)\cos(\psi)\cos(\theta) + \sin(\psi)\cos(\phi) & -\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\cos(\theta) & -\sin(\theta)\cos(\psi) \\ \sin(\phi)\sin(\theta) & \sin(\theta)\cos(\phi) & \cos(\theta) \end{bmatrix}$$

The global thrust vector can then be found by multiplying the fixed-body vector with T_b to give us:

$$T_G = [R]\begin{bmatrix} 0 \\ 0 \\ T_b \end{bmatrix} = T_b \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\phi) \\ \cos(\phi)\cos(\theta) \end{bmatrix}$$

Using Newton's Second Law, $F = ma$ we can arrange the following equation:

$$ma = T_G - g$$

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = T_b \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\phi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{T_b}{m} \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\phi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

with $\frac{T_b}{m}$ being:

$$T_b = \begin{bmatrix} 0 \\ 0 \\ k_t(w_1^2 + w_2^2 + w_3^2 + w_4^2) \end{bmatrix} / m = \begin{bmatrix} 0 \\ 0 \\ \frac{k_t}{m}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix}$$

giving us the final translational equations of motion of:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{k_t}{m}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \cos(\psi)\sin(\phi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

C. Equations of Motion: Rotational

ψ, θ, ϕ are used to represent the yaw, pitch and roll of the quadcopter respectively. Their time derivatives are also represented as $\dot{\psi}, \dot{\theta}, \dot{\phi}$.

We create a rotation matrix that aligns angular velocities from the body frame to the angular rate of change of the Euler angles which can be found using the already described rotation matrices R_x, R_y, R_z above. However, they are organized in a different form of:

$$[R] = \begin{bmatrix} e_x & R^T e_y & (R_y R_x)^T e_z \end{bmatrix}$$

and so our angular velocity vector in the body frame can be expressed as:

$$[\omega] = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

The torque of the drone consists of $\tau_\phi, \tau_\theta, \tau_\psi$, ,which correspond to the rotational angles within the original body frame. The torques are obtained by taking the outer product between the magnitude of the thrust force of the propeller with the distance from the center of rotation. The drone's torques are calculated as:

$$[\tau] = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_t(\omega_1^2 - \omega_3^2)\ell \\ k_t(\omega_2^2 - \omega_4^2)\ell \\ C_D(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix}$$

The Euler's rotational equations are known as:

$$I\dot{\omega} + \omega \times (I\omega) = \tau$$

which can be rearranged to look like:

$$\dot{\omega} = I^{-1}(\tau - \omega \times (I\omega)$$

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} \cdot (\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - [\omega] \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} [\omega])$$

we can then substitute the expression for the ω vector from above, resulting in:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}^{-1} \cdot (\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix})$$

and then expressing the torques fully (with the moments of inertia redacted for display purposes) to give us:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \mathbf{I}^{-1} \cdot (\begin{bmatrix} k_t(\omega_1^2 - \omega_3^2)\ell \\ k_t(\omega_2^2 - \omega_4^2)\ell \\ k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} - \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \times \mathbf{I} \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix})$$

V. Results

A. Method of Simulation

In order to simulate the dynamics of the quadcopter in response to input rotor velocities, we modeled its state over time numerically using the Forward Euler Method. First, we initialized each state variable $x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \theta$ to an initial condition: $x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0, \phi_0, \theta_0, \psi_0, \theta_0$. For each trial, we initialized the rotor velocities $\omega_1, \omega_2, \omega_3, \omega_4$ to a constant value depending on the objective of the simulation. Then, we implemented the Forward Euler Method in Python from (1) and (2).

$$x_{n+1} = x_n + \dot{x}_n dt + \frac{1}{2}\ddot{x}_n t^2 \tag{1}$$

$$\dot{x}_{n+1} = \dot{x}_n + \ddot{x}_n dt \tag{2}$$

For this numerical method, the timestep value was chosen to be $dt = 0.01$. Using a loop in Python, each state variable was updated at each timestep using the Forward Euler Method according to its Equation of Motion. The value of the variable was recorded at that timestep. After the loop was completed, the recorded state values were extracted and plotted as a function of time in 2D and 3D to visualize the quadcopter's behavior during the simulation.

B. Requirements Verification

We must verify that our quadcopter dynamic model and simulation method meets all 4 Engineering Requirements outlined in section II.B.

1. ER1

The first Engineering Requirement states that the quadcopter model must have six degrees of freedom, accounting for three translational and three rotational movements. The model must also account for gravity and lift generated from the 4 rotors. We have developed six Equations of Motion in section IV, three for translational movements and three for rotational movements, and gravity and lift are accounted for within these Equations. Therefore, this requirement has been met.

2. ER2

The second Engineering Requirement states that the drone must be able to modify its state and attitude solely through increasing or decreasing the number of revolutions per second of its 4 rotor blades. The method of simulation only allows for adjustment of the state of the quadcopter through variation of the individual rotor velocities. Therefore, this requirement has been met.

3. ER3

The third Engineering Requirement states that the modeled drone must be able to achieve the Performance Goals (outlined in II.C) without exhausting its power supply.

To determine whether the drone's power consumption exceeded its power supply from the battery, we consulted a reputable 3rd-party resource called xcopterCalc. XcopterCalc is widely used in quadcopter and RC aircraft design as a tool to approximate drone performance and specifications. By inputting the commercial components used, namely the battery, motors, propellers, and body, xcopterCalc determines how long the quadcopter can fly, how much power it uses, and other useful parameters.

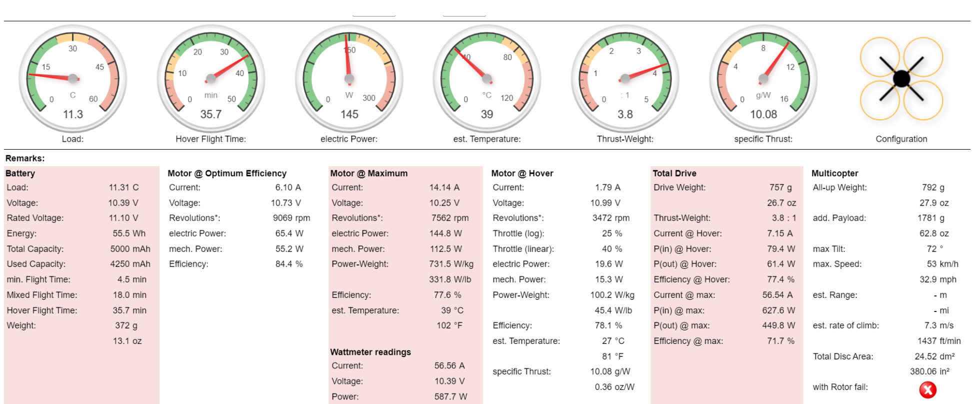

Fig. 4 xcopterCalc Results

After we input the components we selected in Section III, xcopterCalc returned information about our drone capabilities as seen in Fig 4. As demonstrated in the figure, the quadcopter has a hover time of over 35 minutes.

Since all our performance goals take 2 minutes or less and do not use much more power than is required to hover, the quadcopter is demonstrably able to complete all Performance Goals without depleting its power supply. Therefore, this requirement has been met.

4. ER4

The fourth Engineering Requirement states that the modeled drone must be realistic and implementable through the use of off-the-shelf parts, and the total mass of the drone must be between 0.1 and 10 kg. Section III identifies commercial parts for all primary components of the quadcopter, and the total mass of all parts is 792 g, or 0.792 kg. Therefore, this requirement has been met.

C. Completing Performance Goals

Using the numerical simulation setup, we attempted to achieve Performance Goals 1, 2, and 3 as outlined in section II.C.

1. Performance Goal I: Hover

Performance Goal I requires that the quadcopter hover and remain stationary at an altitude of 1 m for at least 2 minutes, or 120 seconds. To achieve this goal, we first determined the rotor velocity required to maintain a steady altitude without accelerating upwards or downwards. Since there is no translational or rotational motion, we can assume all rotor velocities to be equal to some constant value ω_{hover}. Then, this rotor velocity was found by solving the equilibrium equation (3) in the z direction. The thrust force (4) must exactly match the quadcopter weight mg.

$$T = mg \tag{3}$$

$$T = K_T(4\omega^2) \tag{4}$$

Solving, the value for ω_{hover} was obtained to be roughly 804.64899 $\frac{rad}{s}$.

In the simulation environment, the initial state conditions were as follows:

$$\dot{x}_0 = \dot{y}_0 = \dot{z}_0 = \omega_{x_0} = \omega_{y_0} = \omega_{z_0} = x_0 = y_0 = 0 \tag{5}$$

$$z_0 = 1 \tag{6}$$

The rotor velocities $\omega_{1,2,3,4}$ were initialized to to ω_{hover} calculated previously.

After the simulation completed, the quadcopter's position in (x, y, z) and Euler angles (ϕ, θ, ψ) were graphed as a function of time in Figure 5.

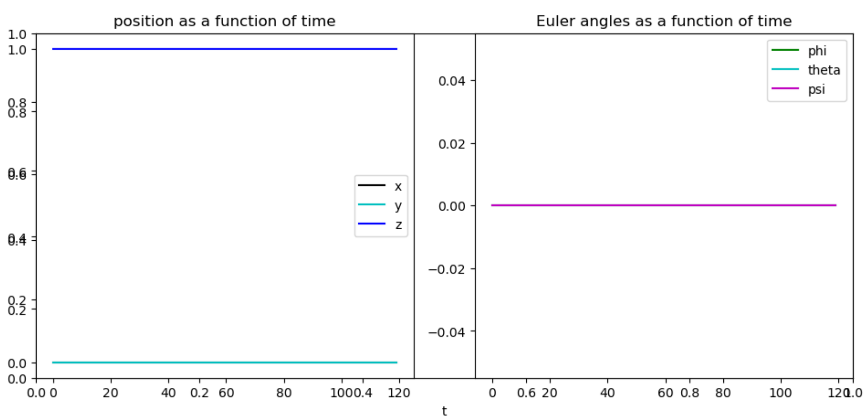

Fig. 5 Position and Euler Angles over time for Hover

As shown in the figure, the quadcopter's altitude z remains steady at 1 m across the duration of 120 seconds. Positions x and y remain constant at 0, as do the Euler angles ϕ, θ, ψ. Therefore, Performance Goal I has been met.

2. Performance Goal II: Circle

Performance Goal II requires that the quadcopter must fly in a circle of radius 2m, at an altitude of 1m above ground at a speed of 0.5 m/s for at least 1 minute. To this end, we first had to determine the initial conditions that would allow the quadcopter to maintain a circular path.

The first step was to calculate what initial bank angle ϕ_0 would allow the quadcopter to exert the exact centripetal acceleration needed to fly in a circle of radius 2m at a tangential velocity of 0.5 m/s. Using trigonometric properties, ϕ_0 was found to be defined in (7), where v_t is the tangential velocity, g is gravitational acceleration, and R is the radius of the circular path.

$$\tan\phi_0 = \frac{{v_t}^2}{gR} \tag{7}$$

Solving (7), ϕ_0 was found to be roughly 0.73 degrees.

Since the quadcopter is slightly banked, the rotor velocities must also be higher to maintain altitude at a different orientation. More specifically, the global vertical component of the thrust vector must be of the same magnitude as the force due to gravity mg. Once again, since there should be no rotational acceleration in any axis throughout the maneuver, all 4 rotors will have the same velocity $\omega_{1,2,3,4} = \omega_0$. From trigonometry with the bank angle ϕ_0 and rearranging, an expression for the rotor velocity ω_0 is found in (8). Solving this equation, ω_0 was found to be about 804.68165 $\frac{rad}{s}$.

$$\omega_0 = \sqrt{\frac{mg}{\cos(\phi_0)4K_T}} \tag{8}$$

Since the quadcopter must be banked toward the center of the circle throughout the turn, it must also be yawing at a constant rate, and therefore has a constant angular velocity about the z axis ω_{z0} throughout the flight. This value is simply defined in (9), where v_t is once again the tangential velocity and R is the radius of the circle. This expression was evaluated to $\omega_{z_0} = 0.25$.

$$\omega_{z0} = \frac{v_t}{R} \tag{9}$$

Furthermore, the centripetal acceleration of the quadcopter also means that the initial velocities $\dot{x}_0$ and $\dot{y}_0$ are nonzero and must be determined. Once again from simple trignonometry, these values were defined with respect to the tangential velocity v_t:

$$\dot{x}_0 = v_t\cos(\phi_0) \tag{10}$$

$$\dot{y}_0 = v_t\sin(\phi_0) \tag{11}$$

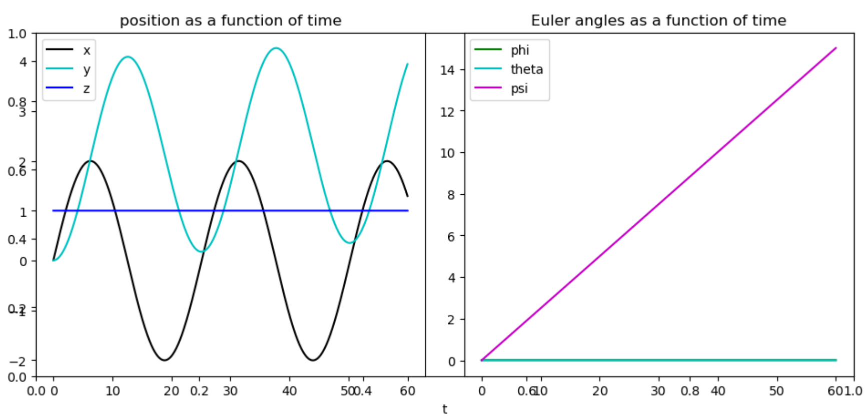

Fig. 6 Position and Euler Angles over time for Circle

Since the altitude of the turn must be at 1m, z_0 is trivially initialized to 1.

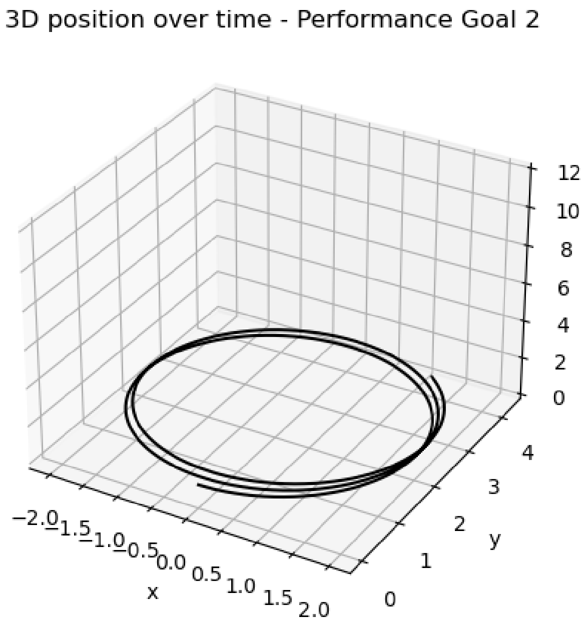

Fig. 7 3D Flight Path for Circle

With all relevant initial conditions defined for $\dot{x}_0$, $\dot{y}_0$, ϕ_0, $\omega_{(1,2,3,4)_0}$, z_0, $and \omega_{z_0}$, all other state variables are initialized to 0:

$$x_0 = y_0 = \dot{z}_0 = \theta_0 = \psi_0 = \omega_{x_0} = \omega_{y_0} = 0 \tag{12}$$

Since all initial conditions are now defined, the simulation could now be run.

The resulting trajectory was graphed as a function of time in Fig 6. The 3D flight path was also visualized in Fig 7.

As clearly demonstrated in the 3D flight path, the quadcopter was able to successfully execute a circular flight path for 60 seconds with a radius of 2 m and a tangential velocity of 0.5 m/s, all while maintaining a steady altitude of 1 m. Therefore, Performance Goal II has been met.

D. Performance Goal III: Complex Flight Path

Performance Goal III requires that the quadcopter must launch from the ground and ascend vertically until 1m above ground. The quadcopter must then move in a straight line 1m above ground at an average speed of 1 m/s for 5m, then stop and yaw 90 degrees to the left, move in another straight line for another 5m, stop, and then land vertically with a speed of no more than 1 cm/s, or 0.01 m/s.

To this end, we had to adjust the rotor velocities at various stages throughout the drone's flight path. Starting with the liftoff from the ground, we decided to accelerate the quadcopter upwards by $0.5\frac{m}{s^2}$ for 1 second, halt acceleration for 1 second, and then decelerate by $0.5\frac{m}{s^2}$ for 1 more second. This would bring the quadcopter exactly to an altitude of 1 m above the ground. To accelerate the quadcopter by $0.5\frac{m}{s^2}$, we solved the equilibrium equation in (13) for thrust T, and then solved (4) for the rotor velocity ω_{climb}.

$$\frac{T - mg}{m} = 0.5 \tag{13}$$

The resulting rotor velocity ω_{climb} was found to be 824.9 $\frac{rad}{s}$. We solved (13) again for -0.5 instead of 0.5 to obtain the rotor velocity required to decelerate by $0.5\frac{m}{s^2}$. This value ω_{sink} was found to be 783.875 $\frac{rad}{s}$

In order to perform the yaw maneuver halfway through the flight, the rotor velocities for the propellers spinning counterclockwise were adjusted to be slightly different from those spinning clockwise. This induced a net torque about the z axis of the drone, causing it to yaw. We decided to accelerate the angular velocity ω_z by $0.5\frac{rad}{s^2}$ for 30 degrees of

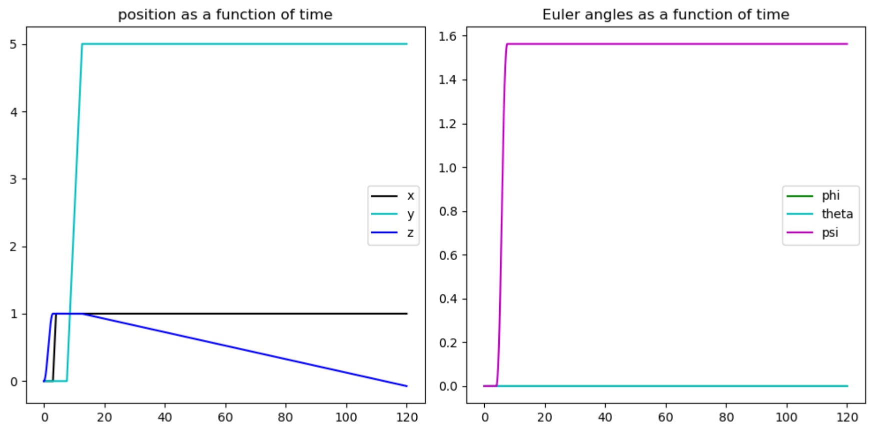

Fig. 8 Position and Euler Angles over time for Complex Path

the turn, halt angular acceleration for another 30 degrees, then decelerate ω_z by $0.5\frac{rad}{s^2}$ for the last 30 degrees of the turn, so the quadcopter will yaw a total of 90 degrees to the left.

To determine the body torque necessary to accelerate ω_z by the specified amount, we solved (14) for torque τ_z, where the body angular acceleration in the z axis α_z was 0.5.

$$\tau_z = I_{zz}\alpha_z \tag{14}$$

This torque value τ_z was then substituted into (15), where ω_{cw} and ω_{ccw} are the rotor velocities of the clockwise and counterclockwise propellers required to induce that torque.

$$\tau_z = C_D(2\omega_{cw}^2 + 2\omega_{ccw}^2) \tag{15}$$

These values were found to be $\omega_{cw} = 804.821845\frac{rad}{s}$ and $\omega_{ccw} = 804.47610\frac{rad}{s}$.

After the rotor velocities for each maneuver were determined, it was time to establish the sequence in the simulation for which the rotor velocities would be set to those values to create the desired flight path.

Within the Python implementation of the Forward Euler method, we implemented "switches" or conditions for what the rotor velocities would be at each point in the flight depending on time or state. Before the simulation was started, all four rotors $\omega_{1,2,3,4}$ were initialized to ω_{lift} as calculated previously to accelerate the drone upwards by $0.5\frac{m}{s^2}$. After time t passed 1 second, $\omega_{1,2,3,4}$ were reset to the hover condition ω_{hover}. At the 2 second mark, $\omega_{1,2,3,4}$ were then set to ω_{sink} to decelerate the drone by $-0.5\frac{m}{s^2}$. At the 3 second mark, $\omega_{1,2,3,4}$ were again reset to ω_{hover}. This brought the drone exactly to 1m altitude.

In the simulator, we then reset the initial condition for $\dot{x}$ to 1, giving it a velocity in the x direction of 1 $\frac{m}{s}$. Once the value of x reached 5m, we reset $\dot{x}$ back to 0 to stop it at that position.

The yaw maneuver was then initiated. Rotor velocities ω_1 and ω_3 were set to ω_{cw}, while rotor velocities ω_2 and ω_4 were set to ω_{ccw}, and the drone began yawing to the left. These rotor velocities were maintained until ψ reached 30 degrees, at which point all rotors were reset to ω_{hover}. The drone continued to rotate until ψ reached 60 degrees, and then ω_1 and ω_3 were then set to ω_{ccw}, while ω_2 and ω_4 were set to ω_{cw}, slowing down the yawing. Once ψ reached 90 degrees, all rotors were once again reset to ω_{hover}. At this point, the drone had already stopped yawing and was now oriented 90 degrees to the left of where it had stopped.

We then reset the initial condition for $\dot{y}$ to 1, giving it a velocity in the y direction of 1 $\frac{m}{s}$. Once the value of y reached 5m, we reset $\dot{y}$ back to 0 to stop it at that position, completing the second straight run of the flight.

Finally, to land the drone, we performed the same steps as taken to lift off but in reverse. All rotors were set to ω_{sink} for 1 second, then ω_{hover} for 1 second, then to ω_{climb} for 1 more second, bringing the drone exactly to a halt at an altitude of 0m, or on the ground. This maneuver means that the vertical speed at landing is exactly 0 $\frac{m}{s}$, as required by the Performance Goal.

The resulting trajectory throughout these maneuvers was graphed in 2D as a function of time in Fig. 8, and also in 3D in 9.

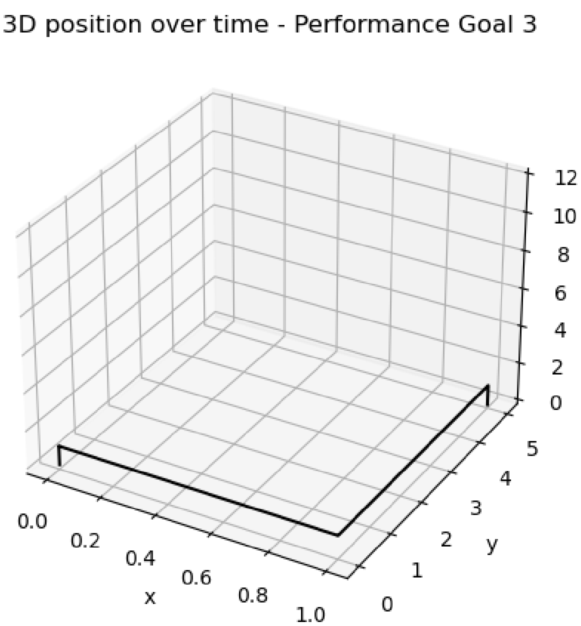

Fig. 9 3D Flight Path for Complex Path

As can be seen in the 3D visualization, the drone took off from the ground to an altitude of 1m, traveled for 5m at a speed of 1 $\frac{m}{s}$ and then stopped. It then yawed 90 degrees to the left, traveled for another 5m, stopped, then landed at a speed of less than 1 $\frac{cm}{s}$. Therefore, Performance Goal III has been met.

VI. Acknowledgments

We would like to thank Professor Siegfried Eggl for his tuition and knowledge that he is willing to share, as well as the material and science to accomplish this project.

We would like to thank the teaching assistants Ani Piromanishvili and Gokul Puthumanaillam for their guidance and their support throughout this project.

VII. List of Contributions

Minh Nguyen: Performance Goals, Results, Power consumption
Richie Ma: Introduction, Part selection, EOM derivation
Dan Anthoney:

VIII. Code Repository

https://github.com/richiem2/AE352-Final-Project/tree/main

References

https://www.aeroflyhobbies.com/brushless-motors/23-t-motor-mt2216-12-800kv-v2-brushless-motor.html
https://xflight-fpv.com/product/lumenier-5200mah-4s-35c-lipo-battery/
https://explorecomposites.com/articles/design-for-composites/basic-numbers-and-calculations-forcomposites/: :text=Carbon
https://www.ecalc.ch/xcoptercalc.php