Drone Butler

By: Brian Francis, Spencer Belleville, Matt Amaro, Remy Allegro, Lalith Athithya Navaneetha Krishnan

Intro:

The idea of the project is to demonstrate the understanding of the theories learned in class by applying robot dynamics to a Quadcopter Drone Butler. As a team, we will be using the robot dynamic theories taught in class to maneuver the DJI F450 drone in simulation. In addition to dynamics, control theory will be applied to the simulated drone to allow a smooth transition without toppling over an object in which it carries.

Dynamics:

The dynamics of a quadcopter refers to the mechanics concerned with the motion of its body under the action of forces. The forces we will be most worried about are drag and gravity. In order to better maneuver the drone we can develop formulas for the drone's linear and angular accelerations that take into account these outside forces. The drone's axes are shown in Figure 1 below.

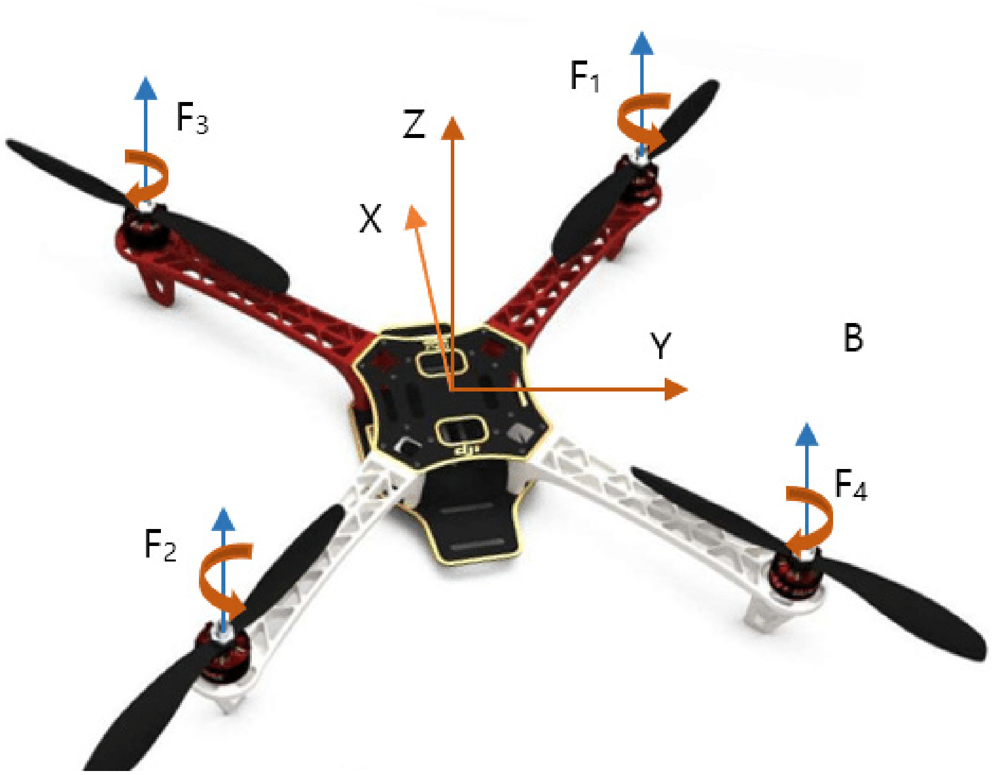


Figure 1: Quadcopter Reference Frame

After establishing the reference frame we are using, we can start establishing some equations of motion. As one can see motors 1 and 2 follow the right hand rule while motors 3 and 4 do not. This is due to the original structure of the blades on the motors. Now by using some of Newton's Laws, we can sum all forces in the same directions to obtain formulas for thrust and torques along the drone's X, Y, and Z axes. These formulas are shown in Figure 2.

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 \\ U_2 = (-F_1 + F_2 + F_3 - F_4) L\sqrt{2}/2 \\ U_3 = (F_1 - F_2 + F_3 - F_4) L\sqrt{2}/2 \\ U_4 = \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{cases} \tag{2}$$

where $\tau_i \approx K_d u_i$ and $F_i \approx K_{th} u_i$ denote the respective torques and forces produced by the ith motor; u_i is the pulse-width modulation; L is the arm length; U_1 is the thrust force; and U_2, U_3, U_4 are the torques in the directions ϕ, θ, ψ, respectively.

Figure 2: Thrust and Torque Equations

Once we have these equations of torques and thrust. We can determine how the combined thrust and torques affect the drone's motion. In a system at equilibrium, the sum of all forces equal zero. We can take it one step further to say that the sum of all forces in one direction for every direction are equal to zero. Now, a force can be simplified to the linear acceleration of the force multiplied by the mass the force is applied to (F = ma) solving for acceleration will come in handy later (a = F/m). However, for angular acceleration, we can use a different equation that uses the torque applied and the inertia of the body the torque is being applied to (a = torque / inertia). With the understanding of this foundation, we can develop the motion equations of the drone. The equations for the linear acceleration in the x, y, and z directions and the angular accelerations around the x, y, and z axes are shown below in Figure 3.

$$\begin{cases} \ddot{x} = \left\{U_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - K_x\dot{x}\right\}/m \\ \ddot{y} = \left\{U_1(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) - K_y\dot{y}\right\}/m \\ \ddot{z} = -g + \left\{U_1(\cos\phi\cos\theta) - K_z\dot{z}\right\}/m \\ \ddot{\phi} = \left(U_2 + (I_2 - I_3)\dot{\theta}\dot{\psi} - K_\phi\dot{\phi}\right)/I_1 \\ \ddot{\theta} = \left(U_3 + (I_3 - I_1)\dot{\phi}\dot{\psi} - K_\theta\dot{\theta}\right)/I_2 \\ \ddot{\psi} = \left(U_4 + (I_1 - I_2)\dot{\phi}\dot{\theta} - K_\psi\dot{\psi}\right)/I_3 \end{cases} \tag{5}$$

where I_1, I_2, I_3 represent the moment inertia along the x, y, z axes; and $K_\phi, K_\theta, K_\psi, K_x, K_y$ represent drag coefficients.

where m is the total mass; g is the gravity;

Figure 3: Dynamic Equations

After establishing the equations of motion for the drone, there are some coefficients that are still unknown. However, in a paper written by a group of university engineers about robust fault estimation (Nguyen et al., 2020), the DJI F450 drone using the above equations was studied. Nguyen et al. (2020) tested and calculated the following values for the quadcopter, including the unknown coefficients.

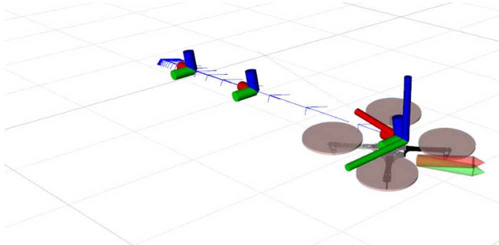
Parameter	Description	Value
L	Arm length	0.225 m
K_{th}	Thrust coefficient	0.0087
K_d	Drag coefficient	0.0055×10^{-2}
m	Total mass	1.776 kg
I_x; I_y; I_z	Moments of inertia	0.0035; 0.0035; 0.0055 kg.m^2
J_r	Rotor inertia	2.8×10^{-6} kg.m^2
υ	Adjustment gain	10
α	Constant	1
ε	Constant	0.02
η	Constant	1

Figure 4: DJI F450 Quadcopter Parameters

Simulation -

The simulation is run using the Multi-robot Systems (MRS) unmanned aerial vehicle (UAV) library as the core. Gazebo runs all of the visualization elements and the computation is completed via ROS noetic. For the testing we utilized the f450 drone model with a pendulum beneath it in a socket joint. The f450 is a consumer grade quadcopter that weighs 996g with a maximum flight speed of 100km/h. The pendulum is a spherical mass of 0.2 kg hung a distance of 1 meter below the chassis of the drone via a cylindrical rod of radius 7.5 cm. In order to test our dynamic analysis we utilized the waypoint flier program that allows us to pass in cartesian coordinates for a waypoint and the system will generate the path to the point from the current location. Our goal was to limit the amount of deflection of the pendulum in a simulation of a payload that should not be swung around violently like a human or fragile cargo.

Controller -



$$\mathbf{f}_d = \overbrace{-\mathbf{k}_p \circ \mathbf{e}_p}^{\text{position feedback}} + \overbrace{-\mathbf{k}_v \circ \mathbf{e}_v}^{\text{velocity feedback}} + \overbrace{m_e \ddot{\mathbf{r}}_d}^{\text{reference feedforward}} + \underbrace{-m_e g \hat{\mathbf{e}}_3}_{\text{gravity compensation}} + \underbrace{-\mathbf{d}_w \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{world disturbance compensation}} + \underbrace{-\mathbf{d}_b \circ \begin{bmatrix}1\\1\\0\end{bmatrix}}_{\text{body disturbance compensation}}$$

The SE(3) controller is a type of control system specifically designed for unmanned aerial vehicles (UAVs). It is a control system that uses the principles of the special Euclidean group, which represents rotations and translations in three-dimensional space. The SE(3) controller is able to provide precise and accurate control of the UAV's orientation, position, and velocity, allowing it to navigate and maneuver in complex environments with a high degree of precision. It also allows us to not only control the traditional values of position, velocity, and acceleration but additionally the jerk as well.

The SE(3) controller uses a variety of sensors and algorithms to continuously monitor and track the UAV's position and orientation, as well as its surrounding environment. It uses this information to calculate the necessary control inputs to maintain the UAV's desired flight trajectory and orientation, while also considering factors such as wind, gravity, and other external forces.

One key advantage of the SE(3) controller is its ability to handle complex, dynamic environments. It is able to adapt to changes in the UAV's surroundings, such as obstacles , varying wind conditions, or dynamic payloads and adjust its control inputs accordingly. This allows the UAV to maintain stable and accurate flight, even in challenging situations.

Results:

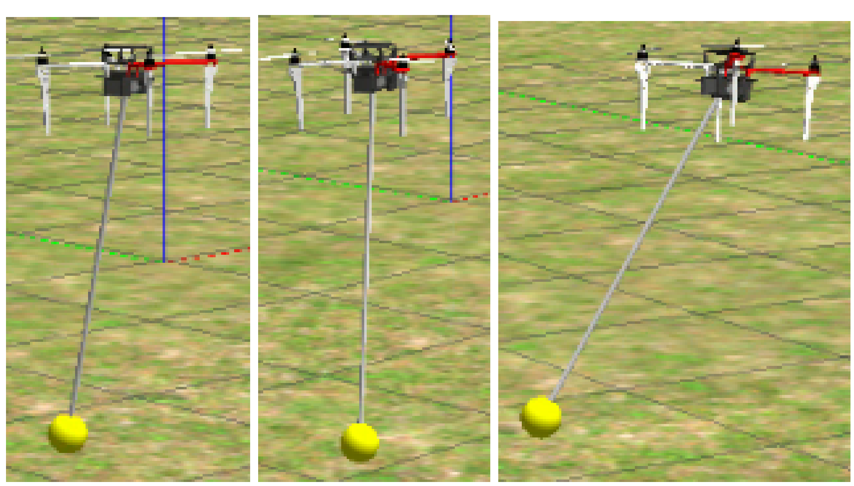


Fig 1: Pendulum displacement during simulation

The experimentation shows that the payload experienced three types of deflection. The first type is shown at the left of figure one is a displacement of 10 degree from stable. This occurred whenever the drone was stopped and then began moving. The drone must go from no acceleration to the desired acceleration value which generates this small deflection. However, once the initial motion has started the acceleration becomes a smooth constant value which brings the deflection to the second stage shown in the middle. As the drone moves to its desired location the pendulum experiences almost no displacement throughout the duration of its flight. The last stage and where the controller failed the most is from rapid turns. As seen on the right, the pendulum experiences a displacement of nearly 25 degrees when the drone attempts to switch directions without slowing down.

While the results are subtly different from the default settings of the simulation there are a few key changes that were seen once we began controlling for jerk. The primary one being that during the middle of the trajectory the pendulum was deflected less as the drone performed less corrective movements. This did cause the drone to get pulled from its initial trajectory by small amounts, but heading corrections caused less payload deflection than rapidly adjusting the thrust. The controller is not perfect however, and did suffer some shortcomings. The initial deflection is believed to be an unavoidable artifact of the nature of acceleration. Since there is an inherent necessity for a change in acceleration from a static position there is going to be some level of deflection that is dependent on what acceleration value is chosen. We decided that the small displacement was an acceptable error size to still maintain a reasonable flight speed. The more concerning value is from the rapid turning that generated such a large deflection. It is unclear where this comes from as the simulation is actually running several controllers in series. The trajectory controller works in tandem with a safety and stability controller to generate the final torque values. So this deflection could come from the drone attempting to stabilize during a turn or another phenomena from when the trajectory controller is not the primary control element. We however believe that the most likely reason is that subsequent flight paths are not interpolated between so as the drone has to come to a stop and then accelerate in a new direction it is resetting the control values and not moving between them smoothly.

Learning Outcomes:

In order to complete this work there were a number of additional challenges in addition to the dynamics and control that we faced. The primary challenge came from working in and modifying the expansive software used to edit and run the simulation. The MRS lab provides singularity images for testing, which required us to familiarize ourselves with not only the singularity software but also how to run and manage scripts in a singularity which was a steep learning curve. Once we had set up our testing environment we then had to localize which elements of the simulation we needed to edit namely the controller and drone model. In order to change the controller elements we had to work with gitman to edit the directory path to include our modifications. The final step involved understanding and editing the SE(3) controller for ROS.

Work Division

Group Member	Task
Brian Francis	Simulation and experimentation
Spencer Belleville	Dynamics research
Matt Amaro	Dynamics research
Remy Allegro	Drone and payload modeling
Lalith Athithya Navaneetha Krishnan	Drone and payload modeling

References:

Nguyen, Ngoc Phi & Huynh, Tuan-Tu & Do, Xuan & Xuan Mung, Nguyen. (2020). Robust Fault Estimation Using the Intermediate Observer: Application to the Quadcopter. Sensors. 20. 4917. 10.3390/s20174917.

T. Lee, M. Leok and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 5420-5425, doi: 10.1109/CDC.2010.5717652.