Quadplane Trajectory Tracker
Mason Peterson

Electric Vertical Take-Off and Landing (eVTOL) aircraft combine the efficiency of fixed-wing aircraft and the maneuverability of multirotor vehicles. Because of their unique capabilities and their potential uses for package delivery and short-range human transportation, eVTOLs have become the subject of numerous recent studies. My research in the MAGICC Lab has involved an eVTOL trajectory tracker controller designed by members of the lab. So, for my project I decided expand the MAV simulator to implement a simple eVTOL trajectory tracker controller.

Model

The first step to this process was to design the simulated graphical vehicle and the dynamics model. For my project, I implemented the dynamics and visuals of a quadplane eVTOL, which looks like a typical fixed-wing aircraft with the addition of four upward facing rotors for vertical take-off and landing. To make the visual for this vehicle, I added a series of adjacent, congruent isosceles triangles with their narrowest vertex centered around each of five points which represented the center of the vehicle's rotors. The graphic for this vehicle is shown in Figure 1.

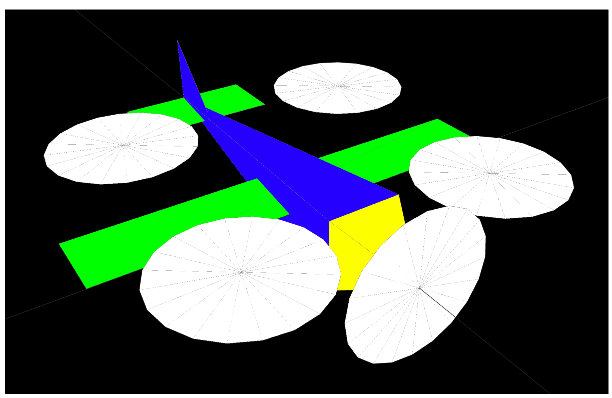

Figure 1: Quadplane simulation graphic

To reflect the aircraft's physical change, I modified the simulation's dynamics to include the thrust and torque produced by each of the rotors. The thrust and torque produced by each rotor is determined using the nonlinear rotor model from the textbook which is a function of throttle and airspeed. The airspeed through each rotor was approximated by rotating the airspeed vector into the frame of each rotor. The thrust and torque produced by each of the rotors is depicted in Figure 2, and can be described by

$$
\begin{gathered}
T_z = -T_p(\delta_r, -w_r) \\
\tau_x = -q_y T_p(\delta_r, -w_r) \\
\tau_y = q_x T_p(\delta_r, -w_r) \\
\tau_z = Q_p(\delta_r, -w_r).
\end{gathered}
$$

where T is thrust produced by the rotor, τ is the torque produced by the rotor, T_p is the nonlinear thrust equation, δ_r is the rotor throttle, w_r is the downward velocity of the body frame, Q_p is the moment produced by the rotor, and $q = [q_x,\ q_y,\ q_z]^T$ is the position of the rotor with respect to the center of gravity.

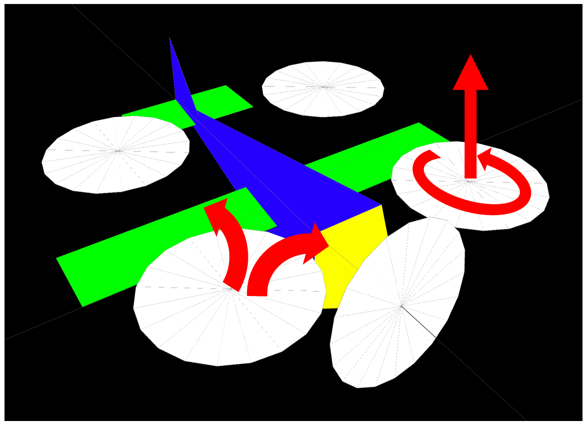
Figure 2: thrusts and torques produced by upward facing rotors

The forces and torques produced by each rotor are summed and added to the original kinematic equations describing the vehicles' motion.

Controller

The controller I wrote was a simplified version of the controller presented in [1], shown in Figure 3. The controller starts by receiving a desired position, velocity, and acceleration. In [1], b-splines are used to provide smooth trajectories for the vehicle. In my project, I connected straight line segments together to provide a rough trajectory for the vehicle. I wrote Python code that takes in a series of points and velocities at each of the points and stitches these together to form a trajectory.

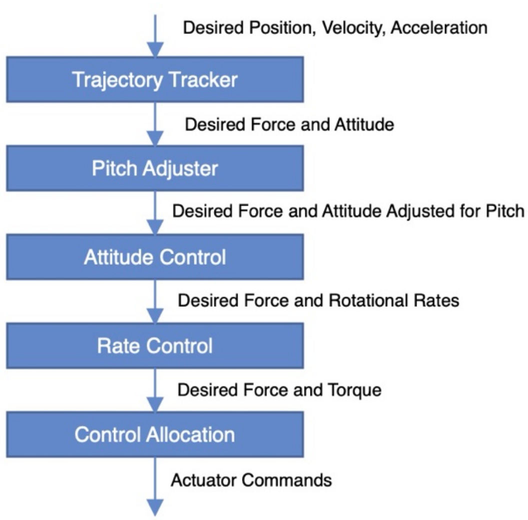

Figure 3: Control Diagram

At each time segment, the desired position, velocity, and acceleration are retrieved from the trajectory generator and the trajectory tracker then computes the force needed to track the trajectory and the attitude in which the vehicle needs to be oriented to produce the desired force. The pitch adjuster then rotates the desired attitude about the body y-axis to ensure that the force vector is within the 90 degrees of producible thrust. The attitude control then computes the angular rates needed to achieve the desired attitude, and the rate control uses PID controls about each of the body axes to calculate the torque needed to achieve those angular rates.

Finally, the control allocation takes the desired thrust and torque vectors and computes the optimal actuator setpoints to achieve that thrust and torque by solving the optimization problem

$$\begin{aligned}
\min\ & \frac{1}{2}\left[\begin{bmatrix}\boldsymbol{F}_d^b \\ \boldsymbol{\tau}_d^b\end{bmatrix} - \mathbf{b}\left(\boldsymbol{\delta}, \mathbf{v}_{b/w}^b\right)\right]^\top K_\tau \left[\begin{bmatrix}\boldsymbol{F}_d^b \\ \boldsymbol{\tau}_d^b\end{bmatrix} - \mathbf{b}\left(\boldsymbol{\delta}, \mathbf{v}_{b/w}^b\right)\right] \\
& \quad + [\boldsymbol{\delta}_{\text{ideal}} - \boldsymbol{\delta}]^\top K_\delta [\boldsymbol{\delta}_{\text{ideal}} - \boldsymbol{\delta}], \\
\text{subject to}\ & \boldsymbol{\delta}_{\min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{\max}
\end{aligned}$$

where b is a nonlinear function of the actuator setpoint vector δ and the airspeed vector $v_{b/w}^b$ representing the control achieved by the actuators, K_τ is a weighting matrix for the different vehicle axes, and K_δ is a weighting matrix to prioritize different actuators.

Simulation Results

This simplified eVTOL trajectory tracker controller shows the capability of tracking simple straight-line trajectories. The simulation results for one trajectory are shown in Figure 4.

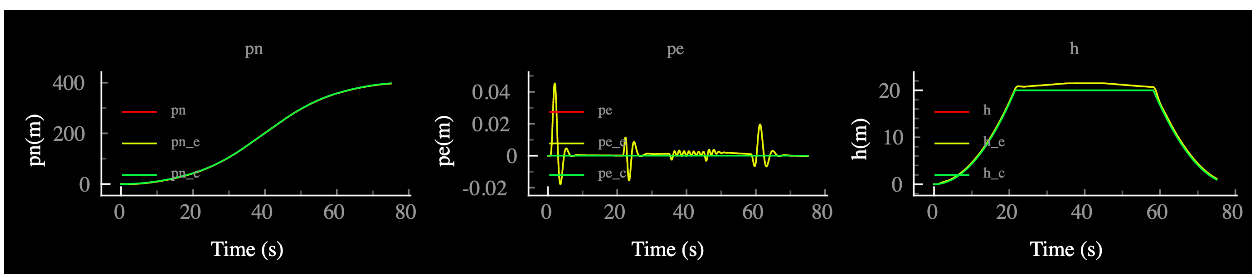

Figure 4: Trajectory Tracking Results

Conclusion

In this writeup, I present my implementation of an eVTOL trajectory tracker controller for a quadplane. The controller demonstrates the capability of tracking straight-line trajectories with modest accelerations. The source code for this project can be found at https://github.com/mbpeterson70/quadplane_project. A video of this project can be found at https://youtu.be/Ril7SRwQxD4.

References

[1] J. B. Willis and R. W. Beard, "Nonlinear Trajectory Tracking Control for Winged eVTOL UAVs," *2021 American Control Conference (ACC)*, 2021, pp. 1687-1692, doi: 10.23919/ACC50511.2021.9482620.