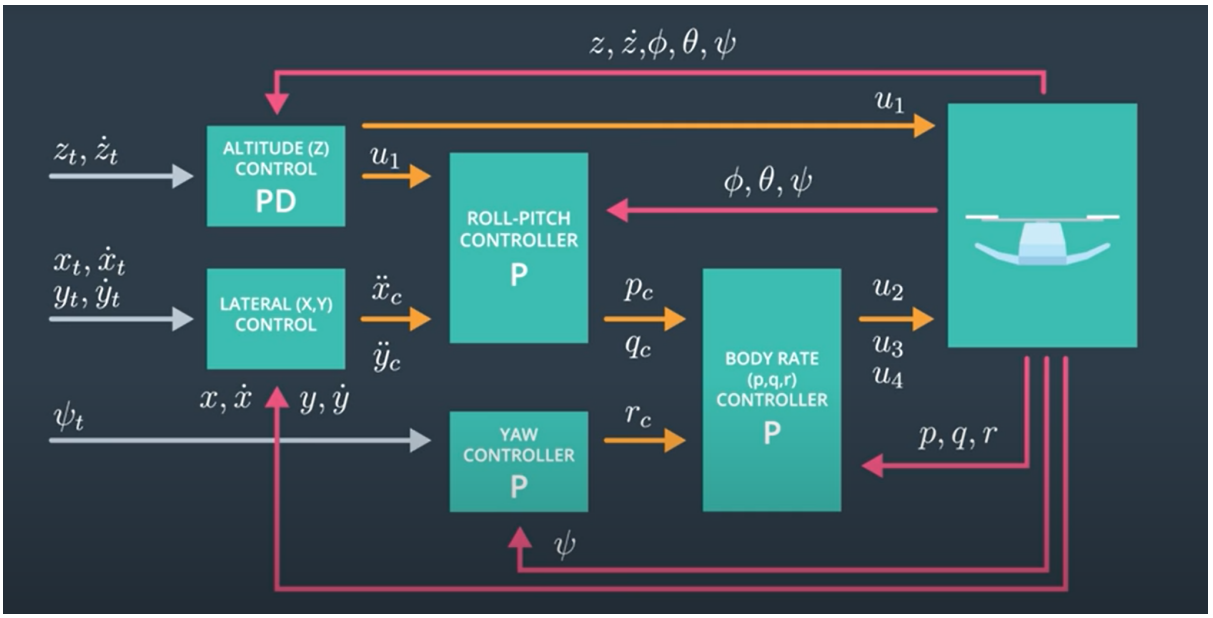
Project 3: Control of a 3D Quadrotor

This project aims to design a controller for a 6DOF drone model. Cascaded controllers and PD, PID controllers are used to controlling the drone. Various scenarios are applied to cover different aspects of controller designs. Tips and tricks and some rules of thumb are given in the document.

Controller Architecture



Altitude controller: 2nd order system => PD Controller at least

Lateral controller: 2nd order system => PD Controller at least

Yaw controller: 1st order system => P Controller

Body rate controller: 1st order system => P Controller

Roll/Pitch controller: 1st order system => P Controller

	Type	Gains
Altitude controller	PD / PID	kpPosZ
		kpVelZ
		kiPosZ
Lateral controller	PD	kpPosXY
		kpVelXY
Yaw controller	P	kpYaw
Roll/Pitch controller	P	kpBank
Body rate controller	P	kpP
	P	kpQ
	P	kpR

Flight Scenarios

Scenario	Task Name	Expected Result	Tuned Parameters
#1	Hover	Altitude hold	Mass
#2	Body rate and Roll/pitch control	Stabilize the rotational motion and bring the vehicle back to level attitude	kpP, kpQ kpBank
#3	Position/velocity and yaw angle control	Quads should be going to their destination points and tracking error should be going down	kpPosXY, kpPosZ kpVelXY, kpVelZ kpYaw, kpR
#4	Non-idealities and robustness	Robustness	kiPosZ
#5	Tracking trajectories	Quad should follow the trajectory	

Performance Metrics

The specific performance metrics are as follows:

- Scenario 2
  - roll should less than 0.025 radian of nominal for 0.75 seconds (3/4 of the duration of the loop)
  - roll rate should less than 2.5 radian/sec for 0.75 seconds
- Scenario 3
  - X position of both drones should be within 0.1 meters of the target for at least 1.25 seconds
  - Quad2 yaw should be within 0.1 of the target for at least 1 second
- Scenario 4
  - position error for all 3 quads should be less than 0.1 meters for at least 1.5 seconds
- Scenario 5
  - position error of the quad should be less than 0.25 meters for at least 3 seconds

Scenario-2:

- *GenerateMotorCommands* function was written. This function takes commanded total thrust and moments, outputs thrust for each propeller.Following equation are used to obtain F_1, F_2, F_3, F_4.

$$\tau_x = (F_1 + F_4 - F_2 - F_3)l$$
$$\tau_y = (F_1 + F_2 - F_3 - F_4)l$$
$$\tau_z = \tau_1 + \tau_2 + \tau_3 + \tau_4$$
$$F_t = (F_1 + F_2 + F_3 + F_4)$$
$$\tau_1 = -k_m\omega_1^2, \tau_2 = k_m\omega_2^2, \tau_3 = -k_m\omega_3^2, \tau_4 = k_m\omega_4^2$$

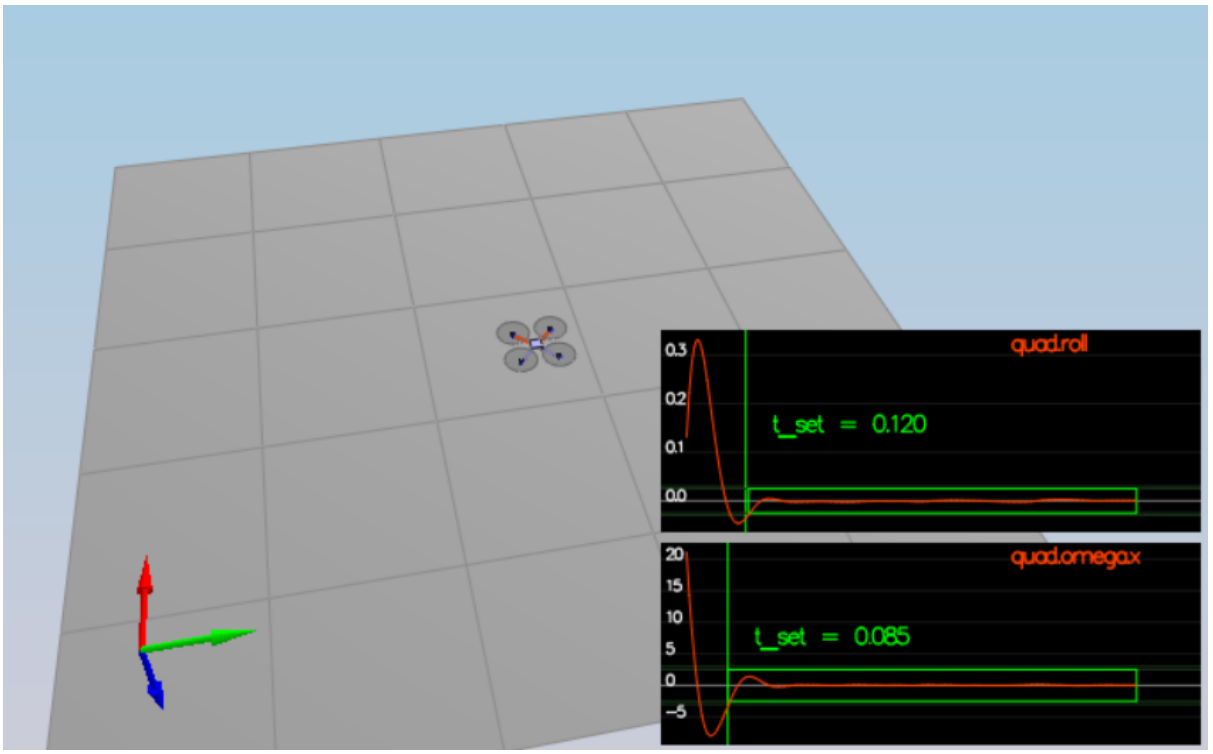
  where *l* is a distance between x-axis and propeller location, which is equal to half of the distance between neighboring propellers. k_f and k_m are force and moment constants of propellers.

- A propotinal controller is implemented in *BodyRateControl* funciton. This function takes commanded body rates and outputs the required moments.
- A proportional controller is implemented in *RollPitchControl* function.This function takes commanded x and y linear accelerations, commanded total thrust and outputs body rate commands.

$$\begin{pmatrix} p_c \\ q_c \end{pmatrix} = \frac{1}{R_{33}} \begin{pmatrix} R_{21} & -R_{11} \\ R_{22} & -R_{12} \end{pmatrix} \begin{pmatrix} \dot{b}_c^x \\ \dot{b}_c^y \end{pmatrix}$$

$$\dot{b}_c^x = k_p(b_c^x - b_a^x)$$
$$\dot{b}_c^y = k_p(b_c^y - b_a^y)$$
$$b_a^x = R_{13}, b_a^y = R_{23}$$
$$b_c^x = \frac{\ddot{x}_c}{c}, b_c^y = \frac{\ddot{y}_c}{c}$$
$$c = \frac{-F_t}{m}$$

- b_c commands are constrained with maximum tilt angle limits.
- kpP, kpQ and kpBank parameters are tuned to meet performace metrics. Since a cascaded controller is used, kpP and kpQ are tuned to have fast response, kpBank should have a slower dynamic compared to body rate control loop.



Scenario-3:

- A PD controller is implemented in *LateralPositionControl* funciton.This function calculates the required x and y linear acceleration for commanded x and y positions. Commanded accelerations are limited accordign to maximum acceleration limits.

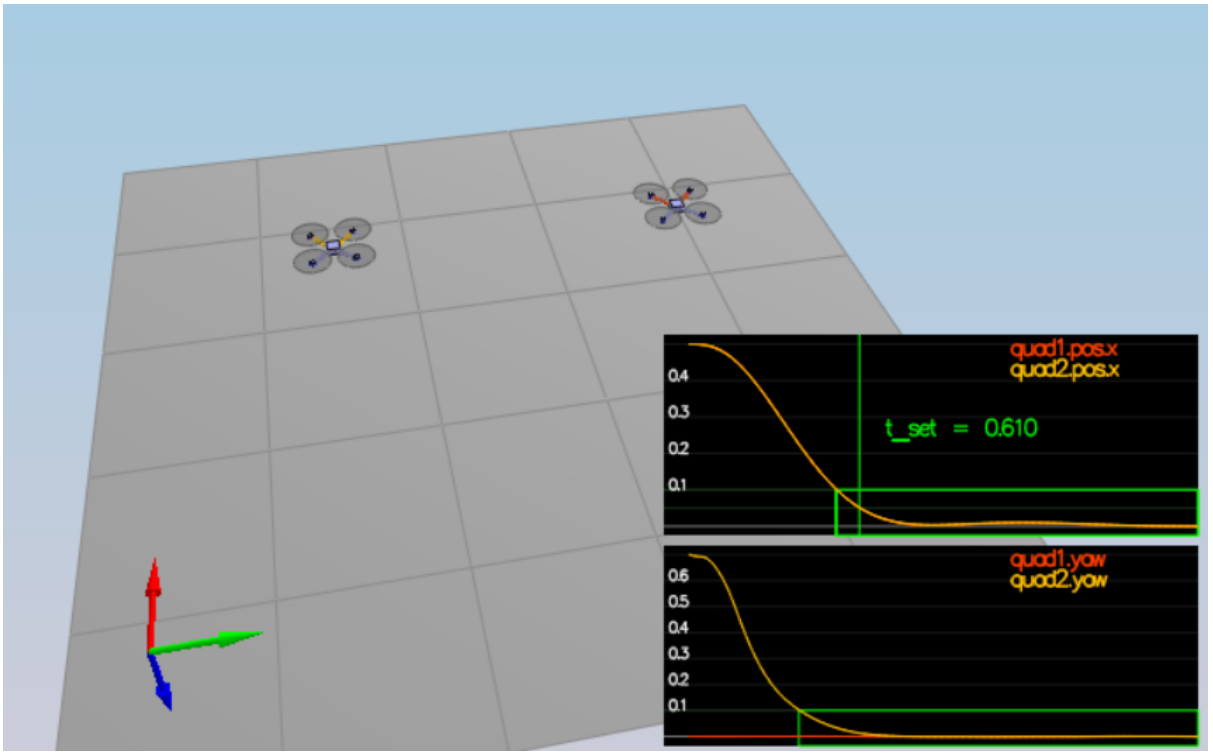
- A PD controller is implemented in *AltitudeControl* funciton.This function calculates to required total thrust to reach to commanded altitude.
- maxAscentRate and maxDescentRate are used to limit the thrust command.
- kpPosXY, kpPosZ, kpVelXY, kpVelZ are tuned.While tuning the PD parameters following relationship is considered to obtain damping ratio between 0.7-1.0.

$$\ddot{e} + K_d\dot{e} + K_p e = 0$$

$$K_d = 2\zeta\omega_n, K_p = {\omega_n}^2$$

$$\zeta = \frac{K_d}{2\sqrt{K_p}}$$

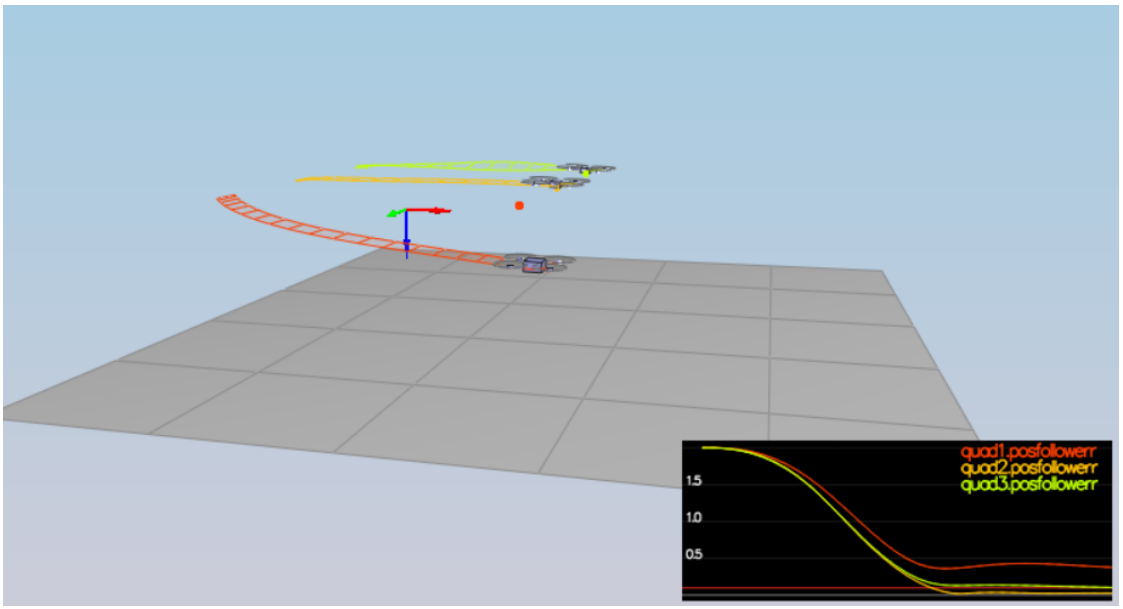
- kpYaw, kpR are tuned.



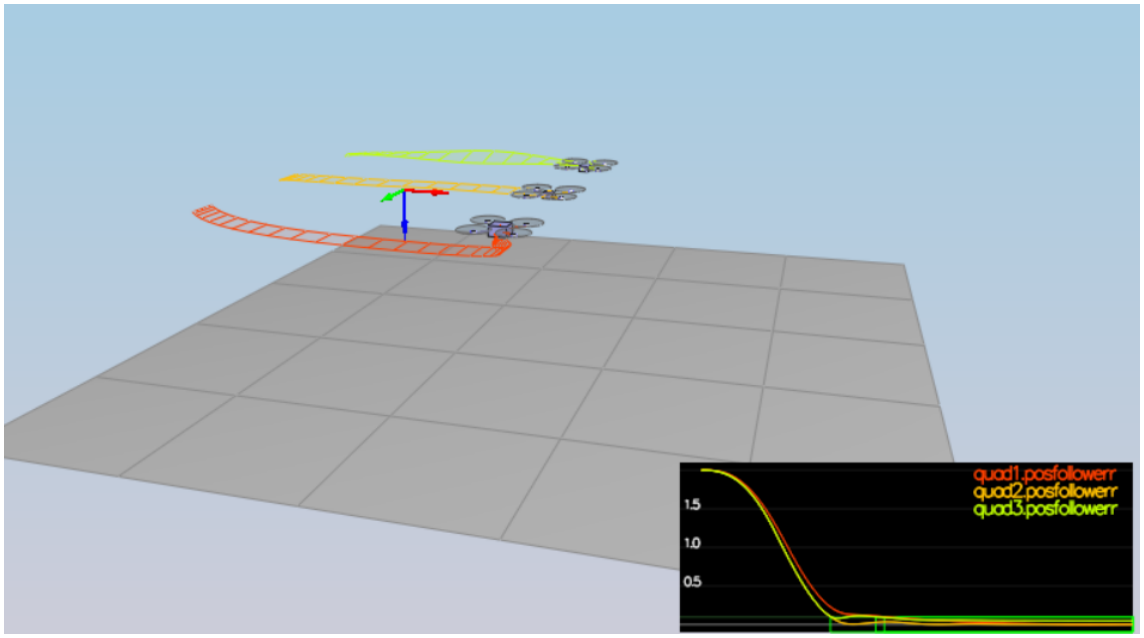
Scenario-4:

- This scenario aims to examine non-idealities and robustness of the designed controller.The green quad has its center of mass shifted back.The orange vehicle is an ideal quad. The red vehicle is heavier than usual.

- As shown in the figure below orange quad cannot reach to the target position due to modeling error.



- Integrator term is added to Altitude controller to compansate for modeling errors.



Scenario-5:

- This task aims to observe tracking performance of 2 quads. The orange one is following traj/FigureEight.txt. The yellow one is following traj/FigureEightFF.txt. The difference between these two text files is that FigureEightFF contains

velocity command too, in addition to position command. Given both velocity and position commands to follow, quad is able to follow the trajectory. But, the orange one could not follow the shape.