Eklavya Project

Drone Design and Simulation

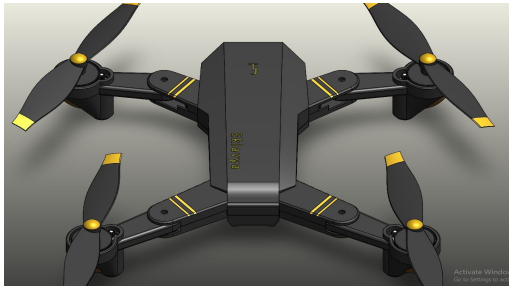
By Toshan Luktuke
Jash Shah

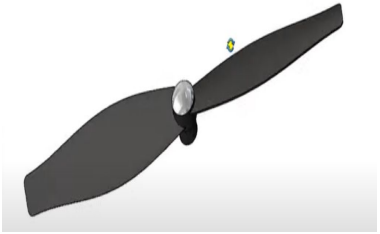
Initial Model

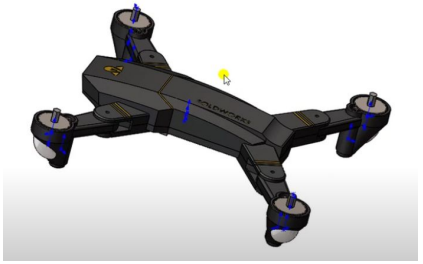
This was our first model:

It was designed in solidworks exported to a URDF and finally brought into the simulation

Abandoned due to issues with plugins, takeoff, motors, rendering and flight



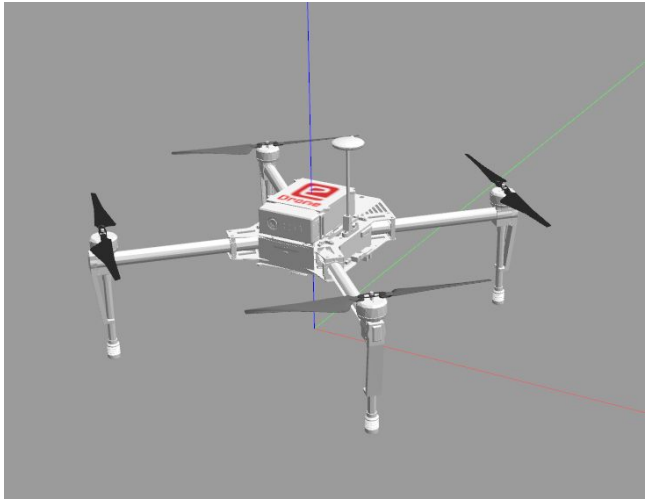




E-Yantra Drone

After letting go of our initial model, we took the iris arducopter model and drone propulsion plugins from E-Yantra IIT-B.

This model worked well and we were able to have the propellers spin as well as have the drone take-off successfully



Starting Scripts

We first designed scripts for takeoff and to read sensor data.

Then we moved onto the PID controllers

First we made the Altitude controller followed by controllers for Roll, Pitch and Yaw

```
1
2  def takeoff():
3      pub = rospy.Publisher('/edrone/pwm', prop_speed, queue_size=10)
4      rospy.init_node('takeoff_node', anonymous=True)
5      rate = rospy.Rate(1) # 10hz
6      speed = prop_speed()
7      speed.prop1 = 510
8      speed.prop2 = 510
9      speed.prop3 = 510
10     speed.prop4 = 510
11     i = 0
12     while not rospy.is_shutdown():
13         if i == 10:
14             speed.prop1 = 508.5
15             speed.prop2 = 508.5
16             speed.prop3 = 508.5
17             speed.prop4 = 508.5
18         rospy.loginfo(speed)
19         pub.publish(speed)
20         i+=1
21         rate.sleep()
22
```

Very first script written for takeoff

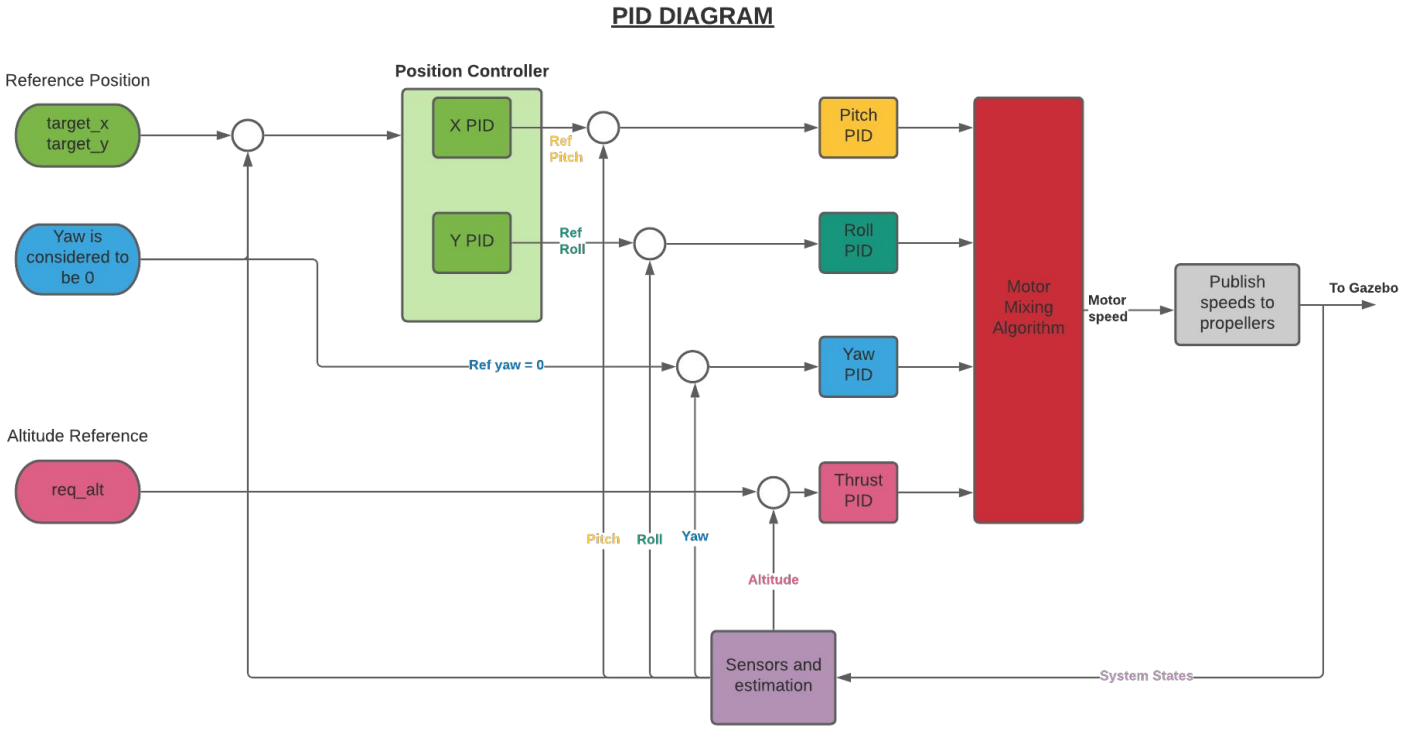
PID for X and Y

Our last steps involved adding PID controllers for X and Y coordinates. To direct the drone towards said coords.

Our biggest challenge while implementing this step was getting the drone to stop at the desired coordinates

```
#equation for correction
if(abs(err_x) < 4 and abs(vel_x) > 0.35):
    dampner = (1/vel_x) * 0.01
    print("Dampner: ", dampner)
    setpoint_pitch = -(vel_x * 1.01  - dampner) #in the direction opposite to velocity
    setpoint_pitch = 10 if (setpoint_pitch > 10) else setpoint_pitch
    setpoint_pitch = -10 if (setpoint_pitch < -10) else setpoint_pitch



#setpoint_pitch = err_x + dErr_x

if(abs(err_y) > 4 ):#and abs(vel_y) < 1.5):
    setpoint_roll = output_y
else:
    setpoint_roll = 0
    #setpoint_roll = output_vel_y

if(abs(err_y) < 4 and abs(vel_y) > 0.35):
    dampner_y = (1/vel_y) * 0.01
    # setpoint_roll = (vel_y * 2.35  - dampner_y) #in the direction opposite to velocity
    setpoint_roll = (vel_y * 2.0  - dampner_y) #in the direction opposite to velocity
    setpoint_roll = 10 if (setpoint_roll>10) else setpoint_roll
    setpoint_roll = -10 if (setpoint_roll <-10) else setpoint_roll
```

Algorithm Flowchart



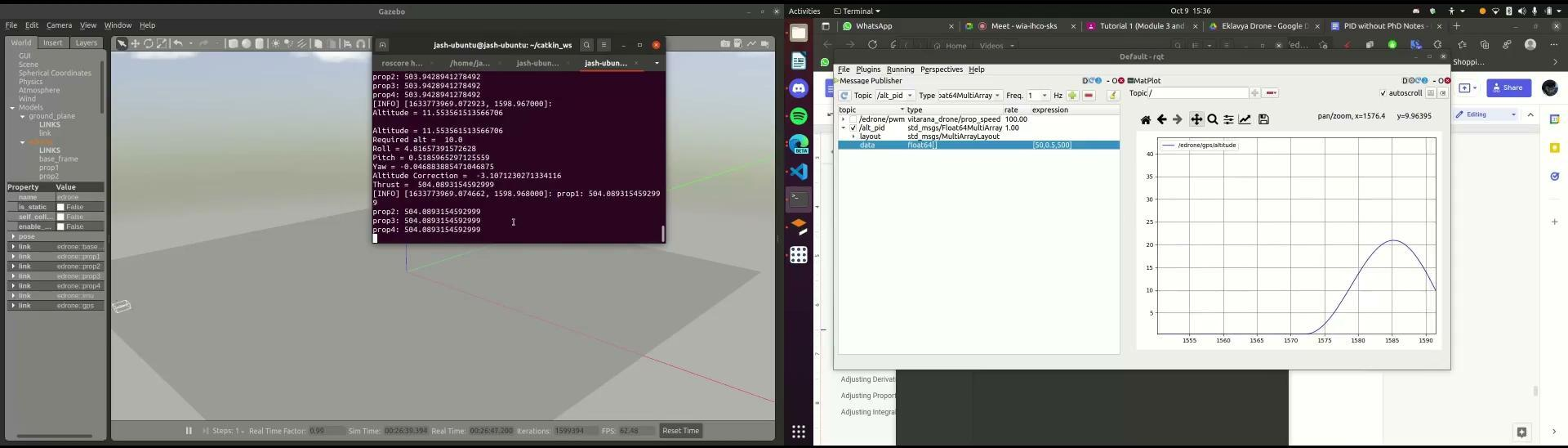
Future Work

Better Tuning: The PIDs can still use some more tuning.

Precision Landing: Once the drone goes in place we can have it land using the camera to calculate position.

Obstacle Avoidance: We have yet to implement any obstacle avoidance features in the drone’s algorithm.

Timelapse of Testing



First test after adding altitude PID

Thank You