**AER1217: Lab 2 Report**

Team E: Aditya Jain, Apurv Mishra, Praharsha Abbireddy

# 1 Introduction

The project consists the design and implementation of a quadrotor position controller in ROS.

# 2 Quadrotor Dynamics and Control

From Newton's law we get,

$$F = ma$$
$$T - mg = ma$$
$$a = -g + T/m$$

where T is the total motor thrust, g is acceleration due to gravity and m is the mass of the quadrotor. We define a rotation matrix R (1) that maps parameters from the body frame to inertial frame. This gives us equation (2).

$$R = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi cos\phi & sin\psi sin\theta cos\phi - cos\psi sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ T/m \end{bmatrix} \tag{2}$$

Substituting (1) in (2) and defining mass-normalized thrust $f = T/m$, we get

$$\ddot{x} = f(cos\psi sin\theta cos\phi + sin\psi sin\phi)$$
$$\ddot{y} = f(sin\psi sin\theta cos\phi - cos\psi sin\phi)$$
$$\ddot{z} = -g + f cos\theta cos\phi$$

Solving for $\psi = 0$,

$$\ddot{x} = f cos\phi sin\theta$$
$$\ddot{y} = -f sin\phi$$
$$\ddot{z} = -g + f cos\theta cos\phi$$

Thus, the final equations are:

$$\phi = sin^{-}1(\frac{-\ddot{y}}{f}) \tag{3}$$

$$\theta = sin^{-}1(\frac{\ddot{x}}{f cos\phi}) \tag{4}$$

$$f = \frac{\ddot{z} + g}{cos\theta cos\phi} \tag{5}$$

# 3 Simulation Results and Control Parameters

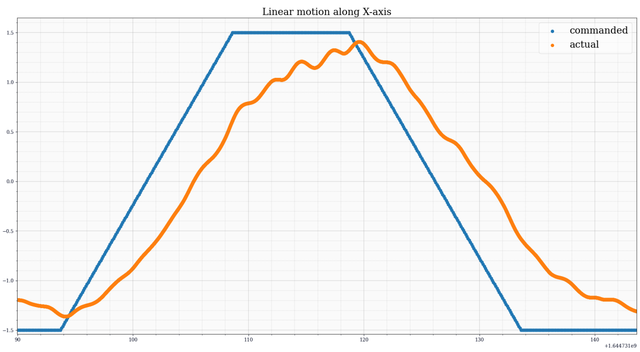
The following graphs plot the data for the desired and actual pose, position and orientation, of the quadrotor for the linear and circular trajectory. The legend labelled as 'actual' are from the VICON system and the ones labelled as 'commanded' are from the trajectory generator.

PD controllers were used for the control of translational motion along X and Y axes. For motion along the Z-axis and the rate of yaw angle, proportional controllers were used. The control parameters chosen were:

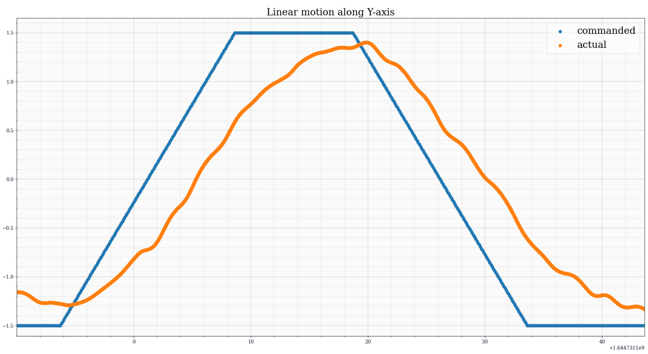
$k_{p,x} = 1.3,\ k_{d,x} = 8.0,\ k_{p,y} = 1.3,\ k_{d,y} = 8.0$
$k_{p,z} = 6.0,\ k_{p,\psi} = 4.0$

While tuning the parameters in PD control, a high proportional gain gave a smaller rise time while sometimes going beyond the destination position. For the derivative gain, a high value gave a lower settling time and the vehicle took longer in stabilizing at the goal position. In the trajectory requirements, a higher speed needs a more controlled motion. When trying to speed up trajectory tracking of multi-rotor, the algorithm ended up moving to the next goal position before reaching the current goal completely. It also caused sudden movements if the number of intermediate points were set too low for a faster motion.

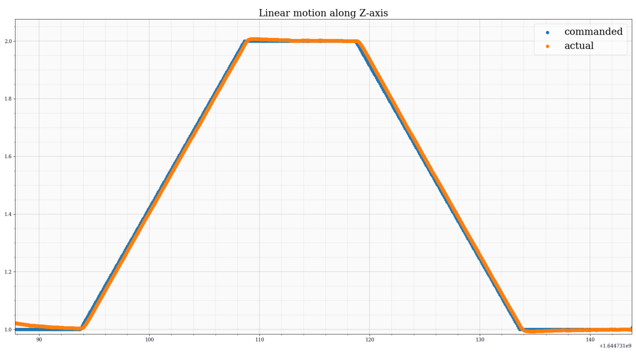
Figure 1 compares the desired and actual pose of the quadrotor for the linear trajectory and figure 2 is for the circular motion. Figure 3 is the error plot in the pose for the two motions.




(a) Desired (commanded) and actual x-positions



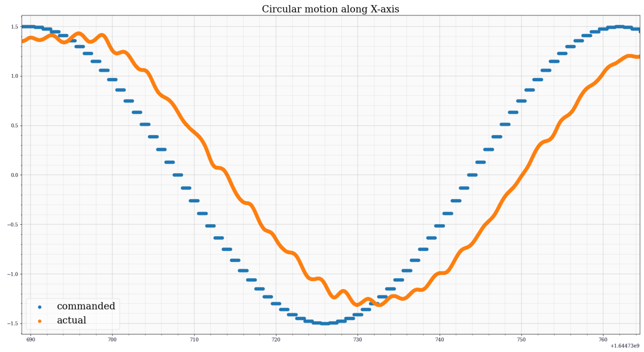

(b) Desired (commanded) and actual y-positions



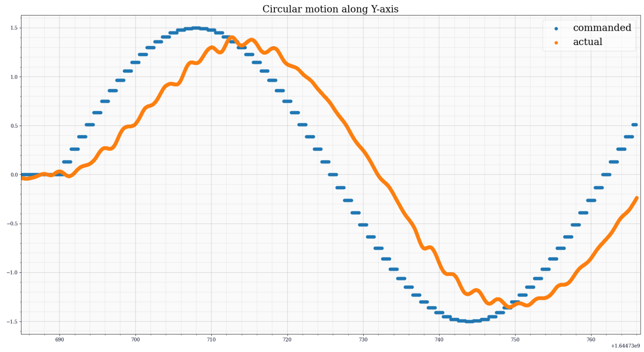

(c) Desired (commanded) and actual z-positions


(d) Desired (commanded) and actual yaw angle

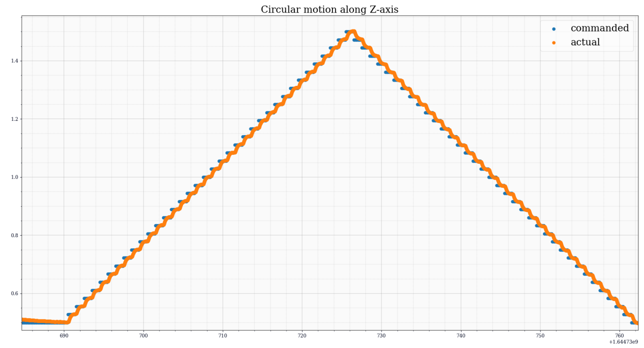
Figure 1: Graphs of desired and actual x, y, z positions and yaw angle for LINEAR trajectory



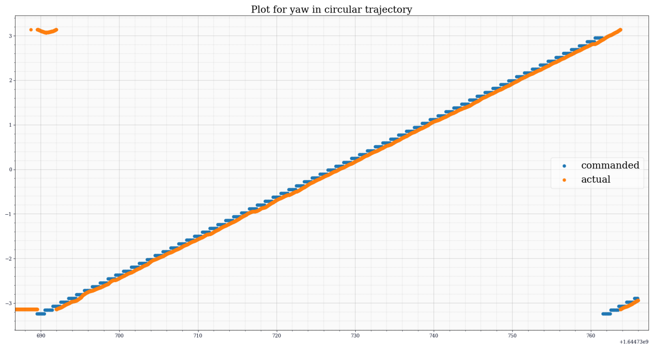

(a) Desired (commanded) and actual x-positions




(b) Desired (commanded) and actual y-positions



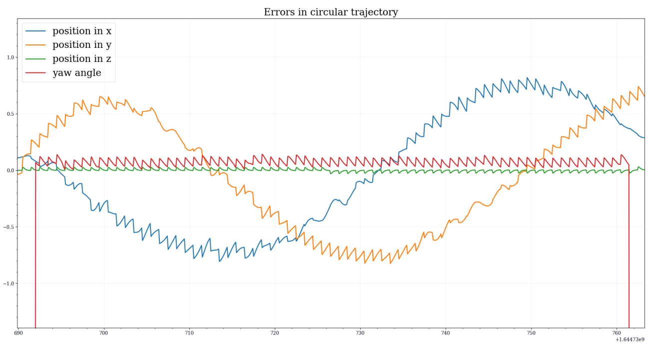

(c) Desired (commanded) and actual z-positions




(d) Desired (commanded) and actual yaw angle

Figure 2: Graphs of desired and actual x, y, z positions and yaw angle for CIRCULAR trajectory


(a) Linear trajectory




(b) Circular trajectory

Figure 3: Errors between desired and actual x, y, z positions and yaw angle

# 4 Video Demonstration

Linear trajectory: `https://youtu.be/hMoVzzk4aso`
Circular trajectory: `https://youtu.be/mBCLUqhjctQ`