Part - a

Let the desired trajectories be

$$x = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

$$y = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0$$

$$z = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

$$\underset{3\times 1}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}} = \underset{3\times 6}{\underbrace{\begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ c_5 & c_4 & c_3 & c_2 & c_1 & c_0 \end{bmatrix}}_{\text{coefficient matrix}}} \underset{6\times 1}{\begin{bmatrix} t^5 \\ t^4 \\ t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}}$$

Given

at $t=0 \Rightarrow (x, y, z) = (0,0,0) \rightarrow t_1$

at $t=5 \Rightarrow (x, y, z) = (0,0,1) \rightarrow t_2$

at $t=20 \Rightarrow (x, y, z) = (1,0,1) \rightarrow t_3$

at $t=35 \Rightarrow (x, y, z) = (1,1,1) \rightarrow t_4$

at $t=50 \Rightarrow (x, y, z) = (0,1,1) \rightarrow t_5$

at $t=65 \Rightarrow (x, y, z) = (0,0,1) \rightarrow t_6$

In matrix form

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 \end{bmatrix} = \begin{bmatrix} a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ b_5 & b_4 & b_3 & b_2 & b_1 & b_0 \\ c_5 & c_4 & c_3 & c_2 & c_1 & c_0 \end{bmatrix} * \begin{bmatrix} t_1^5 & t_2^5 & t_3^5 & t_4^5 & t_5^5 & t_6^5 \\ t_1^4 & t_2^4 & t_3^4 & t_4^4 & t_5^4 & t_6^4 \\ t_1^3 & t_2^3 & t_3^3 & t_4^3 & t_5^3 & t_6^3 \\ t_1^2 & t_2^2 & t_3^2 & t_4^2 & t_5^2 & t_6^2 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\underbrace{}_{3\times 6} \qquad \underbrace{}_{3\times 6} \qquad \underbrace{}_{6\times 6}$$

values are given (P) — unknown (C) — given (T)

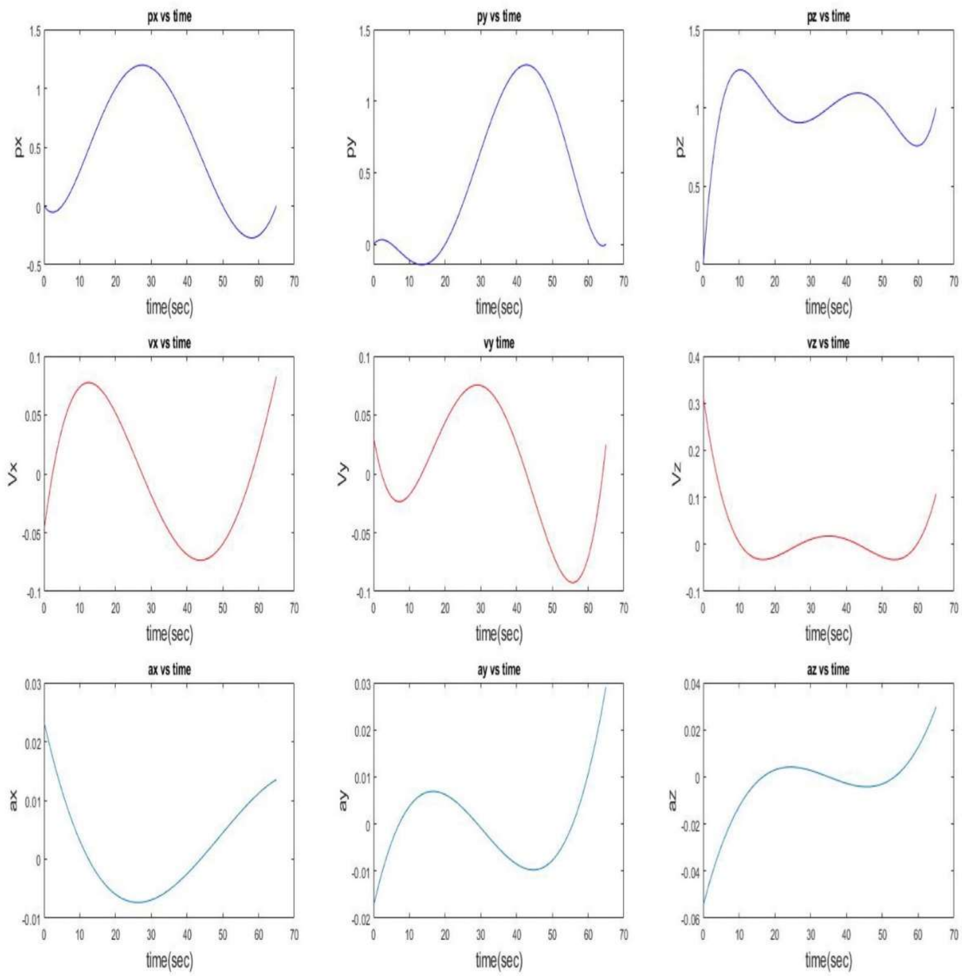
$$P = C * T$$

$$\boxed{\underset{3\times6}{C} = \underset{3\times6}{P} * \underset{6\times6}{T^{-1}}}$$

using matlab

$$\text{coefficient matrix} = \begin{bmatrix} 0.0 & 0.0 & 0.0004 & 0.0116 & -0.0478 & 0 \\ 0.0 & 0.0 & 0.0006 & -0.0087 & -0.0309 & 0 \\ 0.0 & 0.0 & 0.0010 & -0.0273 & 0.3140 & 0 \end{bmatrix}$$

Desired trajectory plots

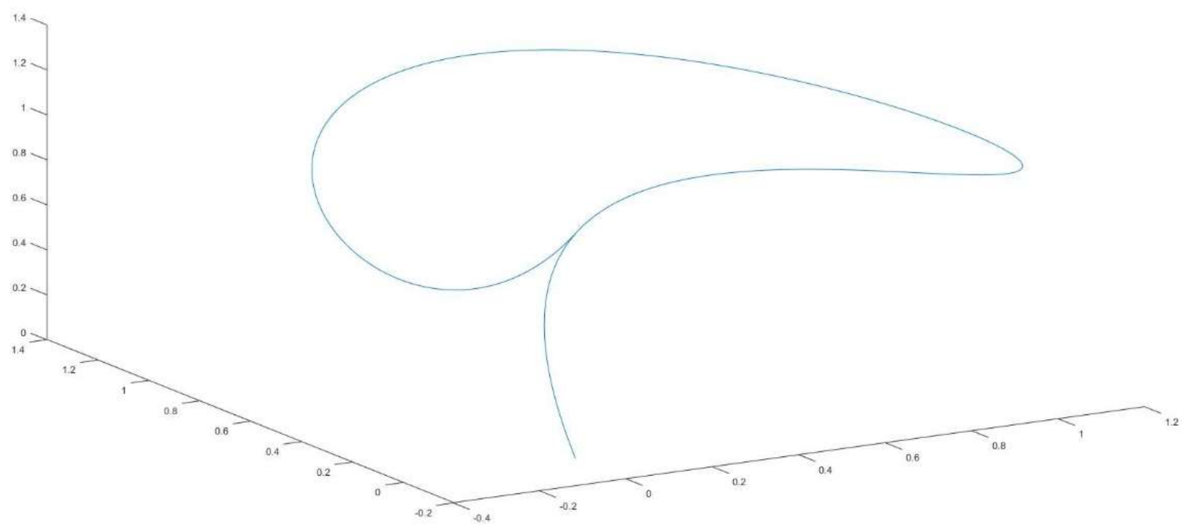


Pros of the generated trajectory.

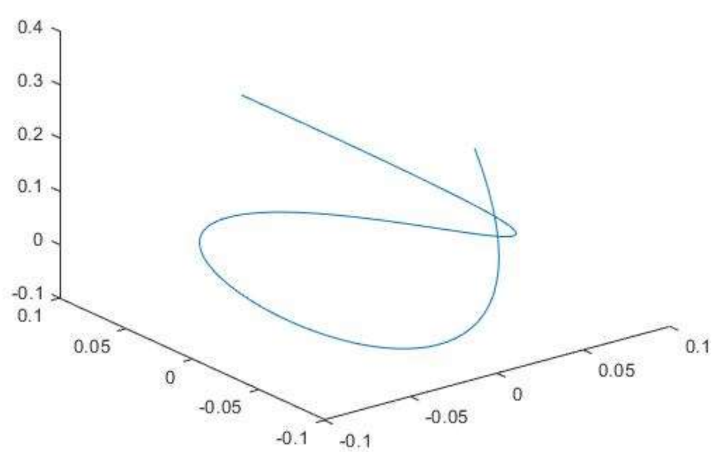
1.Generates smooth motion of the quadrotor.

2.No need for boundary conditions.

3D plot representing Desired Position Trajectory:



3D plot representing Desired Velocity Trajectory:



Part. b

Quad rotor model $q = [x \ y \ z \ \phi \ \theta \ \psi]^T$

control inputs $U = [u_1 \ u_2 \ u_3 \ u_4]$

desired position trajectories can be converted into desired roll and pitch angles using below equations

$$F_x = m\left(-k_p(x - x_d) - k_d(\dot{x} - \dot{x}_d) + \ddot{x}_d\right), \quad (1)$$

$$F_y = m\left(-k_p(y - y_d) - k_d(\dot{y} - \dot{y}_d) + \ddot{y}_d\right), \quad (2)$$

$$\theta_d = \sin^{-1}\left(\frac{F_x}{u_1}\right) \quad (3)$$

$$\phi_d = \sin^{-1}\left(\frac{-F_y}{u_1}\right) \quad (4)$$

$$u_1 = \sqrt{F_x^2 + F_y^2}$$

variables to be tracked are z, ϕ, θ, ψ

Given

$\psi_d = 0$; $\dot{\phi}_d = \dot{\theta}_d = \dot{\psi}_d = 0$ and $\ddot{\phi}_d = \ddot{\theta}_d = \ddot{\psi}_d = 0$

Allocation matrix given

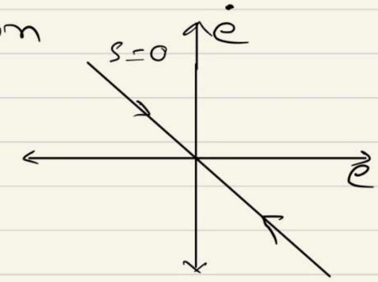
$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_F} & -\frac{\sqrt{2}}{4k_F l} & -\frac{\sqrt{2}}{4k_F l} & -\frac{1}{4k_M k_F} \\ \frac{1}{4k_F} & -\frac{\sqrt{2}}{4k_F l} & \frac{\sqrt{2}}{4k_F l} & \frac{1}{4k_M k_F} \\ \frac{1}{4k_F} & \frac{\sqrt{2}}{4k_F l} & \frac{\sqrt{2}}{4k_F l} & -\frac{1}{4k_M k_F} \\ \frac{1}{4k_F} & \frac{\sqrt{2}}{4k_F l} & -\frac{\sqrt{2}}{4k_F l} & \frac{1}{4k_M k_F} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

Sliding mode controller design

consider equation

$$\ddot{q} = f(q,\dot{q}) + g(q,\dot{q})u \longrightarrow (1)$$

select s as a surface equation



$$\boxed{s = \dot{e} + \lambda e \,, \quad \lambda > 0}$$

where

$e = q - q_d \qquad \dot{e} = \dot{q} - \dot{q}_d$

Here λ is a tuning parameter

consider

$$s\dot{s} = s[\ddot{e} + \lambda \dot{e}]$$

$$= s[\ddot{q} - \ddot{q}_d + \lambda(\dot{q} - \dot{q}_d)]$$

$$s\dot{s} = s[f(q,\dot{q}) + g(q,\dot{q})u - \ddot{q}_d + \lambda(\dot{q} - \dot{q}_d)]$$

select

$$\boxed{u = \left(-f(q,\dot{q}) - \lambda(\dot{q} - \dot{q}_d) + \ddot{q}_d + v_r\right)\frac{1}{g(q,\dot{q})}}$$

substitute u in above equation

we get

$$s\dot{s} = s[v_r] \quad \text{as } f - \hat{f} = 0$$

$$Y_r = -(K)\,\mathrm{sgn}(s)$$

$$s\dot{s} = s[-K\,\mathrm{sgn}(s)] \le -K|s|$$

where K is a tuning parameter

choose a control law ($u(t)$) outside of $S(t)$ such that sliding condition is satisfied

$$\boxed{s\dot{s} \le -K|s(t)| \quad K>0}$$

$$U = \left(-f(q,\dot{q}) - \lambda(\dot{q}-\dot{q}_d) + \ddot{q}_d - K\,\mathrm{sgn}(s)\right)\frac{1}{g(q,\dot{q})}$$

Given equations of motions

(1) $$\ddot{z} = \frac{1}{m}(\cos\phi\cos\theta)u_1 - g$$

$$\ddot{z} = -g + \left(\frac{1}{m}\right)(\cos\phi\cos\theta).u_1$$

$$\ddot{q}_1 = f_1(q_1,\dot{q}_1) + g_1(q_1,\dot{q}_1)\,u_1$$

(2) $$\ddot{\phi} = \underbrace{\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\Omega\dot{\theta}}_{f_2(q,\dot{q})} + \underbrace{\frac{1}{I_x}}_{g_2(q_2,\dot{q}_2)}u_2$$

$$\ddot{q}_2$$

(3) $$\underset{\ddot{q}_3}{\underset{\downarrow}{\ddot{\theta}}} = \underbrace{\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi}}_{f_3(q_3,\dot{q}_3)} + \underset{g_3(q_3,\dot{q}_3)}{\underset{\downarrow}{\frac{1}{I_y}}} u_3$$

(4) $$\underset{\ddot{q}_4}{\underset{\downarrow}{\ddot{\psi}}} = \underbrace{\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z}}_{f_4(q_4,\dot{q}_4)} + \underset{g_4(q_4,\dot{q}_4)}{\underset{\downarrow}{\frac{1}{I_z}}} u_4$$

consider sliding equations for each input.

choose $$u = \left[-f(q,\dot{q}) - \lambda(\dot{q} - \dot{q}_d) + \ddot{q}_d - K\,\mathrm{sgn}(s)\right]\frac{1}{g(q,\dot{q})}$$

$$\boxed{u_1 = \left[g - \lambda_1(\dot{z} - \dot{z}_d) + \ddot{z}_d - K_1\,\mathrm{sgn}(s_1)\right]\cdot\frac{1}{\frac{1}{m}\cos\phi\cos\theta}}$$

$$\boxed{u_2 = \left[-\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} + \frac{I_p}{I_x}\Omega\dot{\theta} - \lambda_2(\dot{\phi}) - K_2\,\mathrm{sgn}(s_2)\right]\cdot\frac{1}{\frac{1}{I_x}}}$$

(Given $\ddot{\phi}_d = 0$, $\dot{\phi}_d = 0$, $\ddot{\theta}_d = 0$, $\dot{\theta}_d = 0$)

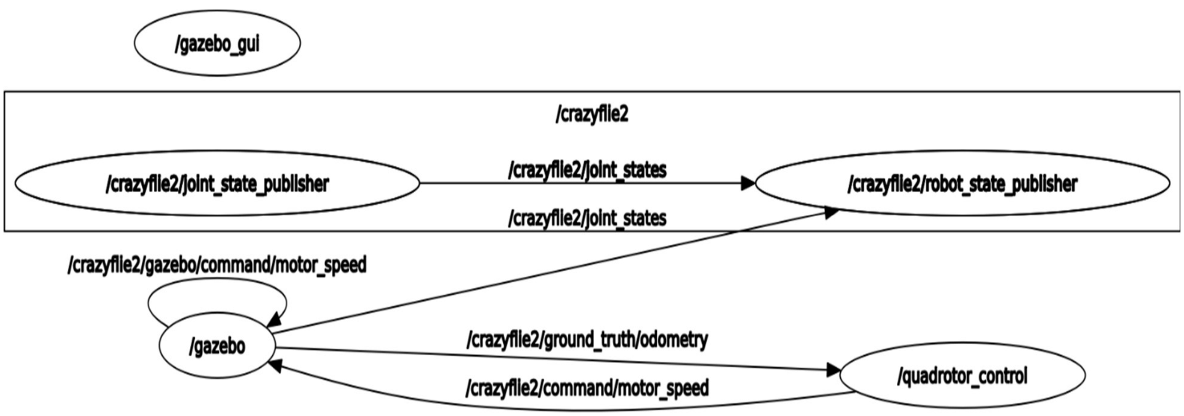
$$\boxed{u_3 = \left[-\left(\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi}\right) - \lambda_3(\dot{\theta}) - K_3\,\mathrm{sgn}(s_3)\right]\cdot\frac{1}{1/I_y}}$$

$$u_4 = \left[-\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} - \lambda_4 \dot{\psi} - K_4 \,\mathrm{sgn}(s_4) \right] \cdot \frac{1}{1/I_z}$$

above 4 equations represents, control law designed to track desired trajectories (derived in part a).

$$\Omega = \omega_1 - \omega_2 + \omega_3 - \omega_4$$

Gazebo -Ros Implementation



In **quad_control.py** detailed explanation of each part of code is explained in the comments.

Final Tuning Parameters for tracking Desired trajectory:

Tuning Parameters	Values
Kp	20
Kd	-3.5
lamda1	0.5
rhok1	1
lamda2	5
rhok2	145
lamda3	10
rhok3	200
lamda4	10
rhok4	5

Explanation:

Initially, low values were selected for all the tuning parameters, which resulted in less error in sliding surface. From this we concluded that the controller was working fine and K, Kp and Kd values requires further tuning.

Next, we increased k values first which resulted in better response in other directions other than z.

Though not as much as needed, we concluded that the problem was converting desired x, y to roll and pitch. Hence, we began tuning the Kp, Kd values.

With the above values quadrotor converges to desired trajectory but it can use further tuning.

Discussion on 3D plot and Controller performance:

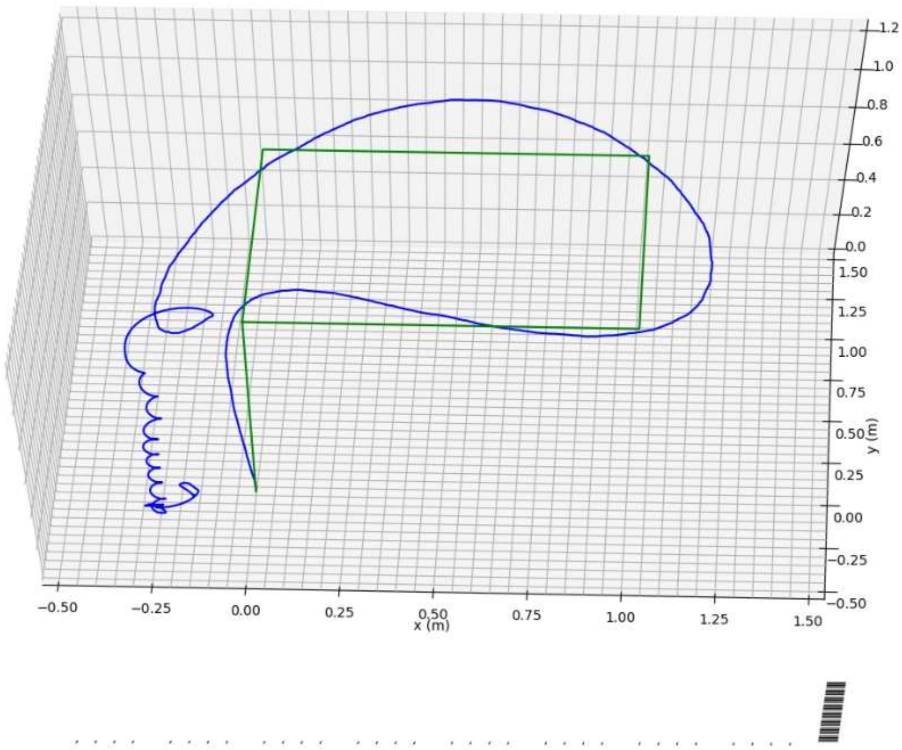


Figure 1 Trajectory Plot of quadrotor in Gazebo simulation

Desired Trajectory Path:

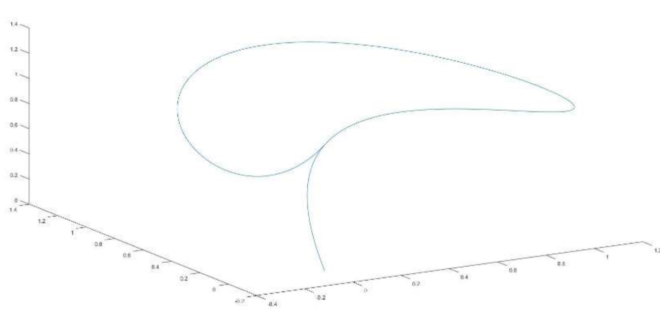


Figure 2 Desired Trajectory generated in MATLAB

Explanation:

Since we generated desired trajectory (fig.2) for entire course of the motion of the quad rotor, so we are getting a curved desired trajectory.

On comparison with figure.2 desired trajectory 3D plot in MATLAB, we can conclude that our gazebo simulated trajectory converges to desired trajectory.

The deviation in last course of the path was because we have set the velocities to 0 after reaching the final waypoint.