Robust Trajectory Tracking for Quadrotor UAVs using Sliding Mode Control

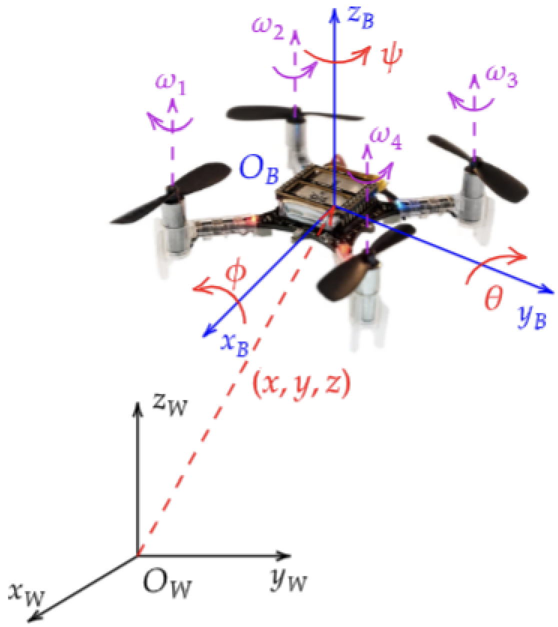


Figure 1: Crazyflie coordinate frames.

Step 1 a: Generalised Coordinates and Generalised Inputs:

$$q = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{1}$$

Step 1 b: State variable:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

Step 2: State Vector:

$$\dot{X} = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{3}$$

where,

$$\ddot{x} = \frac{1}{m}\{\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi\}u_1$$

$$\ddot{y} = \frac{1}{m}\{\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi\}u_1$$

$$\ddot{z} = \frac{1}{m}(\cos\phi \cos\theta)u_1 - g \tag{4}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\Omega\dot{\theta} + \frac{1}{I_x}u_2 \tag{5}$$

where $\Omega = \omega_1 - \omega_1 + \omega_3 - \omega_4$,
I_x, I_y, I_z are the quadrotor moment of Inertia in X,Y,Z directions.

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi} + \frac{1}{I_y}u_3 \tag{6}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \frac{1}{I_z}u_4 \tag{7}$$

1 PART 1:

We generate quintic polynomial trajectories for the translation coordinates of quadrotor.
We provide the time span and the given initial conditions of the quadrotor to the `generate_trajectory.m`, this code will provide us an output with the coefficient of the trajectory equation for respective Translational Coordinates.

$$T \cdot A = B$$

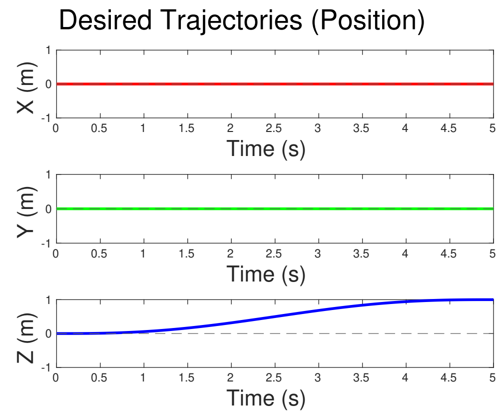
$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q_0} \\ \ddot{q_0} \\ q_f \\ \dot{q_f} \\ \ddot{q_f} \end{bmatrix}$$

where, t_0 and t_f are the time start and time end.
And q_0, $\dot{q_0}$,$\ddot{q_0}$, are initial position, velocity and acceleration while q_f,$\dot{q_f}$,$\ddot{q_f}$ are Final position, velocity and acceleration .

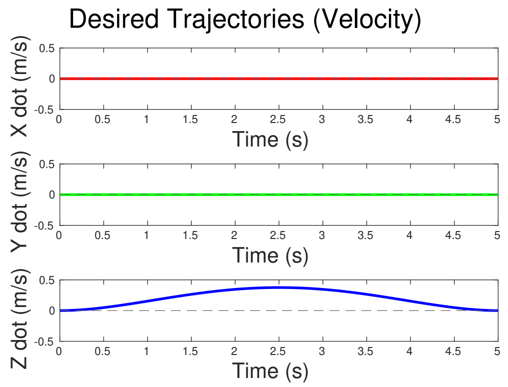
$$A = T^{-1} \cdot B$$

1.1 Reference Trajectories:

We visualize the reference trajectories for all the time spans

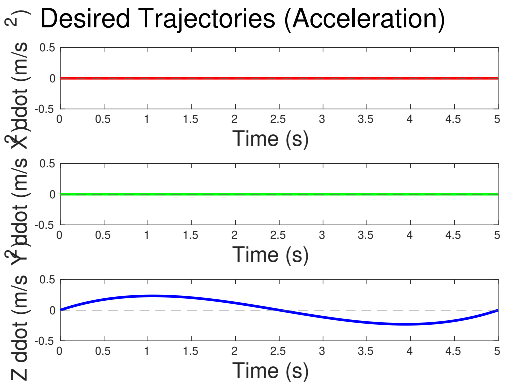


(a) Desired position.

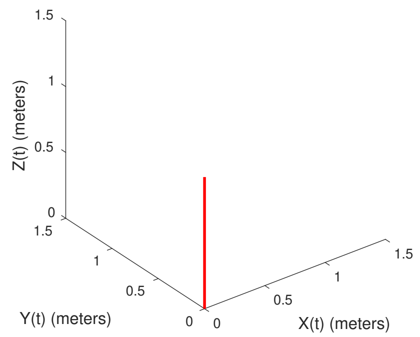


(b) Desired Velocity.

Figure 2: For time t=0 seconds to t=5 seconds desired Trajectories.

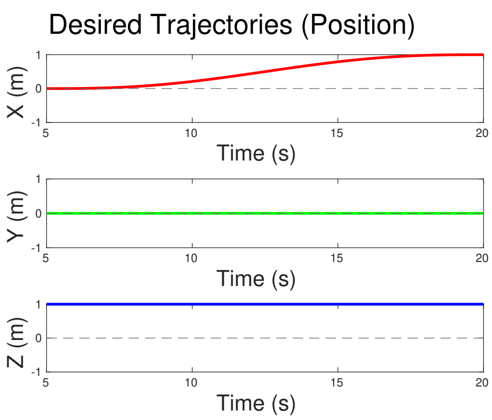


(a) Desired acceleration.

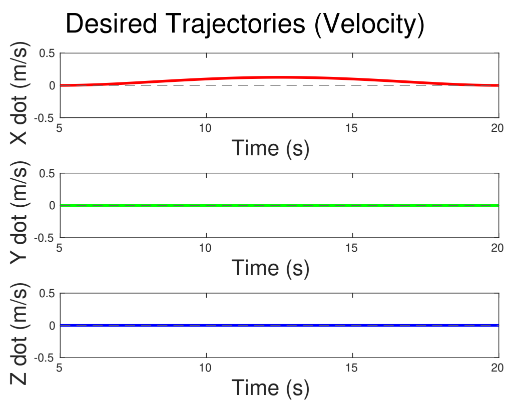


(b) 3D Plot

Figure 3: For time t=0 seconds to t=5 seconds desired Trajectories.

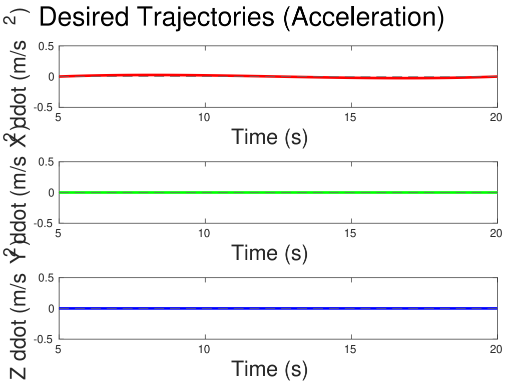


(a) Desired position.

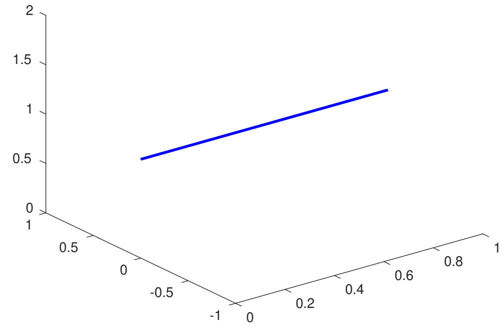


(b) Desired Velocity.

Figure 4: For time t=5 seconds to t=20 seconds desired Trajectories.

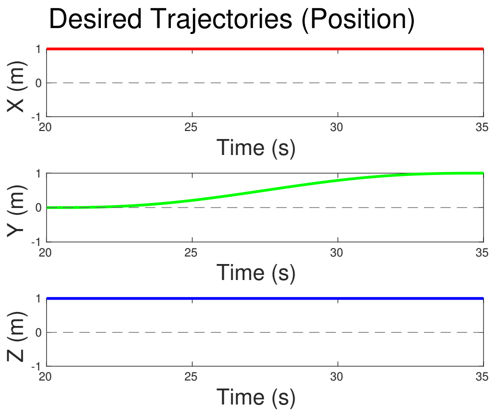


(a) Desired acceleration.

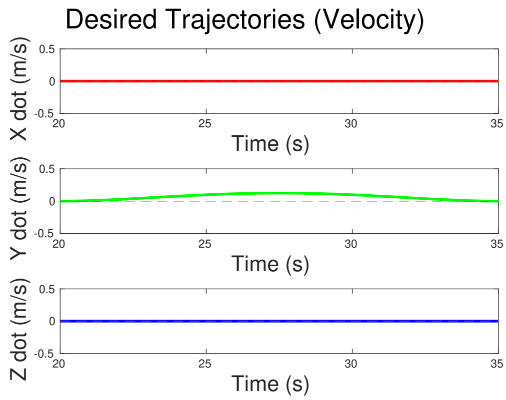


(b) 3D Plot.

Figure 5: For time t=5 seconds to t=20 seconds desired Trajectories.

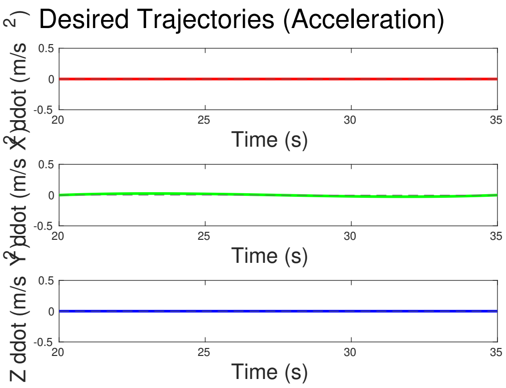


(a) Desired position.

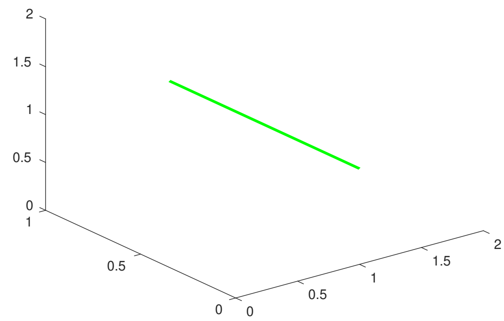


(b) Desired Velocity.

Figure 6: For time t=20 seconds to t=35 seconds desired Trajectories.

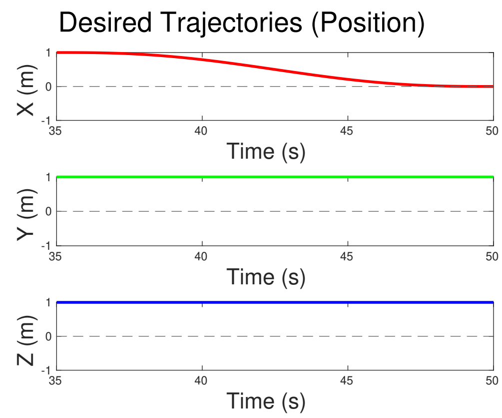


(a) Desired acceleration.

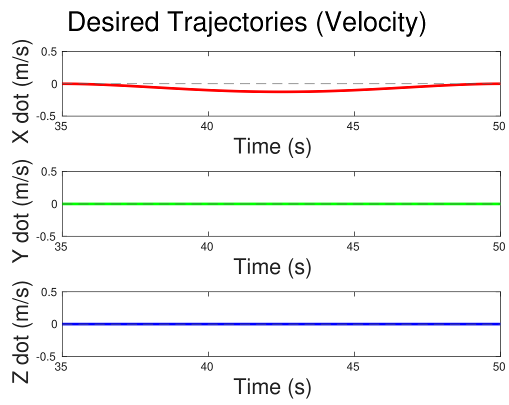


(b) 3D Plot

Figure 7: For time t=20 seconds to t=35 seconds desired Trajectories.

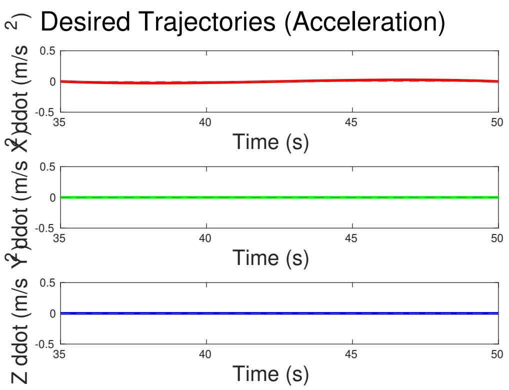


(a) Desired position.

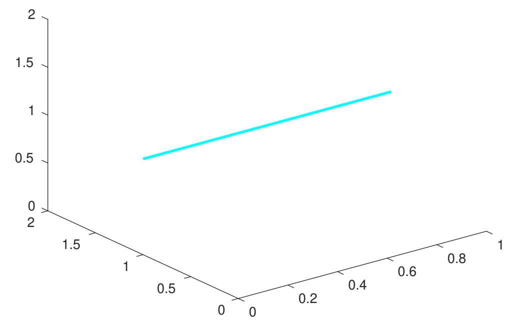


(b) Desired Velocity.

Figure 8: For time t=35 seconds to t=50 seconds desired Trajectories.

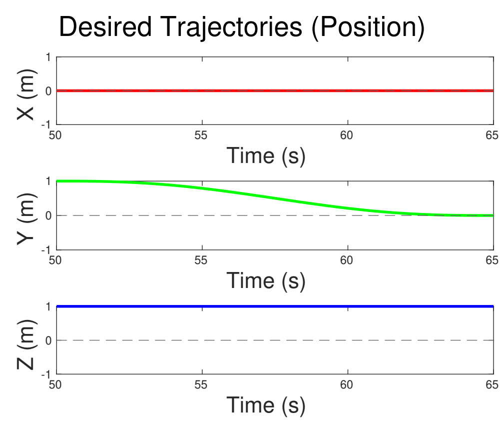


(a) Desired acceleration.

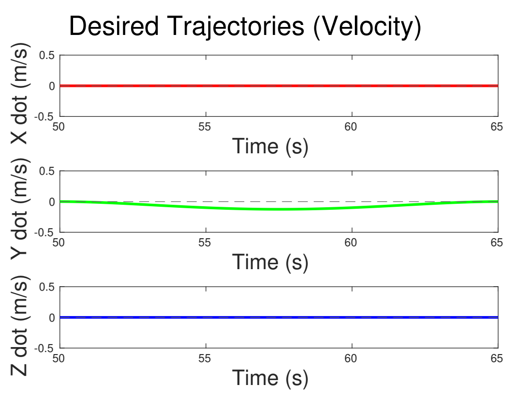


(b) 3D Plot

Figure 9: For time t=35 seconds to t=50 seconds desired Trajectories.

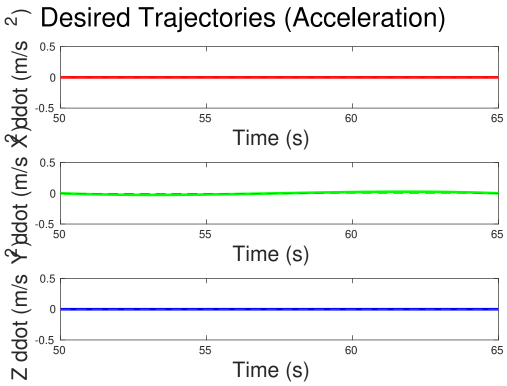


(a) Desired position.

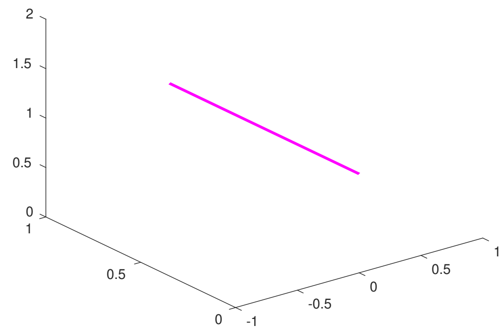


(b) Desired Velocity.

Figure 10: For time t=50 seconds to t=65 seconds desired Trajectories.



(a) Desired acceleration.



(b) 3D Plot

Figure 11: For time t=50 seconds to t=65 seconds desired Trajectories.

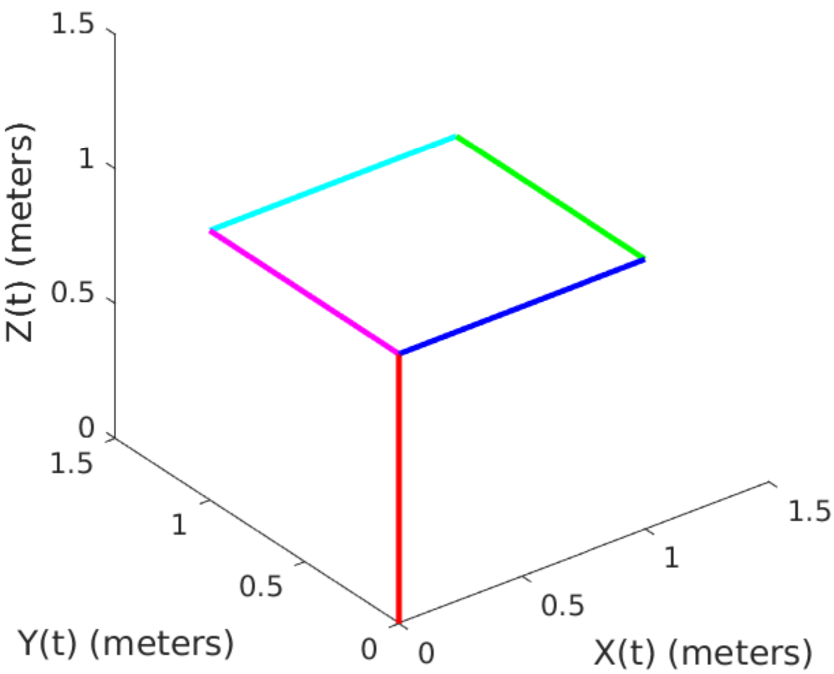


Figure 12: 3D Reference Trajectory.

2 PART 2: Sliding Mode Control Law

2.1 Calculating the sliding surface

2.1.1 Designing sliding surface for z (altitude)

Selecting s as a surface equation, where $\lambda > 0$ and e is the error between desired and controlled Trajectories position. **Sliding Surface Equation**

$$s = \dot{e} + \lambda e \tag{8}$$

$$\begin{aligned} s &= (\dot{z} - \dot{z}_d) + \lambda(z - z_d) \\ \dot{s} &= \ddot{e} + \lambda \dot{e} \\ \dot{s} &= (\ddot{z} - \ddot{z}_d) + \lambda(\dot{z} - \dot{z}_d) \end{aligned}$$

$$\begin{aligned} s_1 &= (\dot{z} - \dot{z}_d) + \lambda(z - z_d) \\ \dot{s_1} &= (\ddot{z} - \ddot{z}_d) + \lambda(\dot{z} - \dot{z}_d) \end{aligned}$$

$$\begin{aligned} \dot{s}_1 =& \ddot{z} - \ddot{z}_d + \lambda_1 (z - z_d) \\ \dot{s}_1 =& \frac{1}{m} \cos(\phi) \cdot \cos(\theta) \, x_1 - g - \ddot{z}_d + \lambda_1 (\dot{z} - \dot{z}_d) \end{aligned} \tag{9}$$

Sliding Condition, $s\dot{s} \leq -k|s|$

$$s_1 \dot{s}_1 = s_1 \cdot \left\{ \frac{\cos(\phi) \cdot \cos(\theta)}{m} \left\{ u_1 - \frac{m\{g - \lambda_1 (\dot{z} - \dot{z}_d) + \ddot{z}_d\}}{\cos(\phi) \cdot \cos(\theta)} \right\} \right\} \tag{10}$$

$$\rho \geq \left\{ \frac{m\{\lambda_1 (x_5 - \dot{z}_d) - \ddot{z}_d\} - g}{\cos(x_2) \cdot \cos(x_3)} \right\} \tag{11}$$

We need to design u_1 such that $s_1 \dot{s_1} \leq -\mathrm{k}_1 |s_1|$

$$u_1 = \frac{m\{g + \ddot{z}_d - \lambda_1 (x_5 - \dot{z}_d) - K_1 \cdot \mathrm{sign}(s_1)\}}{\cos(x_2) \cdot \cos(x_3)} \tag{12}$$

To prevent chattering we use sat function in place of $sign$

$$u_1 = \frac{m\{g + \ddot{z}_d - \lambda_1 (x_5 - \dot{z}_d) - K_1 \cdot sat(\frac{s_1}{\alpha_z})\}}{\cos(x_2) \cdot \cos(x_3)} \tag{13}$$

where $\alpha_1 = 0.9$ is boundary layer constant. .

2.1.2 Designing sliding surface for ϕ

Surface Equation

$$\begin{aligned} s_2 &= \dot{e} + \lambda e \\ s_2 &= (\dot{\phi} - \dot{\phi}_d) + \lambda_2(\phi - \phi_d) \end{aligned}$$

$$\begin{aligned} \dot{s_2} &= (\ddot{\phi} - \ddot{\phi}_d) + \lambda_2(\dot{\phi} - \dot{\phi}_d) \\ &= \left\{ \frac{1}{I_x} \{ (\dot{\theta}\dot{\psi}(I_y - I_z) - I_p \cdot \Omega\dot{\theta} + u_2) \} - \ddot{\phi}_d + \lambda_2(\dot{\phi} - \dot{\phi}_d) \right\} \end{aligned}$$

$$s_2 \dot{s_2} = s_2 \left\{ \frac{1}{I_x} \{ (\dot{\theta}\dot{\psi}(I_y - I_z) - I_p \cdot \Omega\dot{\theta} + u_2) \} - \ddot{\phi}_d + \lambda_2(\dot{\phi} - \dot{\phi}_d) \right\}$$

Sliding Condition, $s\dot{s} \leq -k|s|$

$$s_2\dot{s_2} = \frac{s_2}{I_x}\Big\{\Big(\dot{\theta}\dot{\psi}(I_y - I_z) - -I_p \cdot \sigma\dot{\theta} + \lambda_2 I_x x_6\Big) + u_2\Big\} \tag{14}$$

We need to design u_2 such that $s_2\dot{s_2} \leq -\mathrm{k}_2|s_2|$

$$u_2 = \big\{-\dot{\theta}\dot{\psi}(I_y - I_z) + I_p \cdot \Omega\dot{\theta} + I_x\ddot{\phi_d} - \lambda_2 I_x(\dot{\phi} - \dot{\phi_d}) - I_x k_2 \cdot \mathrm{sign}(s_2)\big\} \tag{15}$$

To prevent chattering we use sat function in place of $sign$

$$u_2 = \big\{-\dot{\theta}\dot{\psi}(I_y - I_z) + I_p \cdot \Omega\dot{\theta} + I_x\ddot{\phi_d} - \lambda_2 I_x(\dot{\phi} - \dot{\phi_d}) - I_x k_2 \cdot sat(\frac{s_2}{\alpha_2})\big\} \tag{16}$$

where $\alpha_2 = 0.9$ is boundary layer constant

2.1.3 Designing sliding surface for θ

Surface Equation

$$s_3 = \dot{e} + \lambda_3 e$$
$$s_3 = (\dot{\theta} - \dot{\theta_d}) + \lambda_2(\theta - \theta_d)$$

$$\begin{aligned}\dot{s_3} &= (\ddot{\theta} - \ddot{\theta_d}) + \lambda_2(\dot{\theta} - \dot{\theta_d}) \\ &= \frac{1}{I_y}\big\{\dot{\phi}\dot{\psi}(I_z - I_x) + I_p \cdot \Omega \cdot \dot{\phi} + u_3\big\} - \ddot{\theta_d} + \lambda_3(\dot{\theta} - \dot{\theta_d})\end{aligned}$$

Sliding Condition, $s\dot{s} \leq -k|s|$

$$s_3\dot{s_3} = s_3\frac{1}{I_y}\big\{\dot{\phi}\dot{\psi}(I_z - I_x) + I_p \cdot \Omega \cdot \dot{\phi} + u_3\big\} - \ddot{\theta_d} + \lambda_3(\dot{\theta} - \dot{\theta_d})$$

We need to design u_3 such that $s_3\dot{s_3} \leq -\mathrm{k}_3|s_3|$

$$u_3 = \big\{-\dot{\phi}\dot{\psi}(I_z - I_x) - I_p \cdot \Omega\dot{\phi} + I_y\ddot{\theta_d} - \lambda_3 I_y(\dot{\theta} - \dot{\theta_d}) + I_y k_3 \cdot \mathrm{sign}(s_3)\big\}$$

(17)

To prevent chattering we use sat function in place of $sign$

$$u_3 = \big\{-\dot{\phi}\dot{\psi}(I_z - I_x) - I_p \cdot \Omega\dot{\phi} + I_y\ddot{\theta_d} - \lambda_3 I_y(\dot{\theta} - \dot{\theta_d}) + I_y k_3 \cdot sat(\frac{s_3}{\alpha_3})\big\}$$

(18)

where $\alpha_3 = 0.9$ is boundary layer constant

2.1.4 Designing sliding surface for ψ

Surface Equation

$$s_4 = \dot{e} + \lambda_4 e$$
$$s_4 = (\dot{\psi} - \dot{\psi_d}) + \lambda_4(\psi - \psi_d)$$

$$\begin{aligned}\dot{s_4} &= (\ddot{\psi} - \ddot{\psi_d}) + \lambda_2(\dot{\psi} - \dot{\psi_d}) \\ &= \frac{1}{I_z}\big\{\dot{\phi}\dot{\psi}(I_x - I_y) + u_4\big\} - \ddot{\psi_d} + \lambda_4(\dot{\psi} - \dot{\psi_d})\end{aligned}$$

Sliding Condition, $s\dot{s} \leq -k|s|$

$$s_4\dot{s_4} = s_4\frac{1}{I_z}\big\{\dot{\phi}\dot{\psi}(I_x - I_y) + u_4\big\} - \ddot{\psi_d} + \lambda_4(\dot{\psi} - \dot{\psi_d})$$

We need to design u_4 such that $s_4\dot{s_4} \leq -\mathrm{k}_4|s_4|$

$$u_4 = -\dot{\phi}\dot{\psi}(I_x - I_y) + I_z\ddot{\psi}_d - \lambda_4 I_z(\dot{\psi} - \dot{\psi}_d) + I_z k_4 \cdot \mathrm{sign}(s_4) \quad (19)$$

To prevent chattering we use sat function in place of $sign$

$$u_4 = -\dot{\phi}\dot{\psi}(I_x - I_y) + I_z\ddot{\psi}_d - \lambda_4 I_z(\dot{\psi} - \dot{\psi}_d) + I_z k_4 \cdot sat(\frac{s_4}{\alpha_4}) \quad (20)$$

where $\alpha_4 = 0.9$ is boundary layer constant

3 Part 3: ROS Implementation of Sliding Mode Control

We develop a ROS package `project` and a ros node `smc_control.py` to implement the sliding mode controller. Please find the src code zip `kadam_natu_project_codes.zip`
We implement the sliding surface as in the eq. 8. We compute the forces F_x and F_y using PD controller o convert the desired positions to desired roll and pitch angles. Then implement the control laws for u_1, u_2, u_3, u_4 using the corresponding equations as derived in eq. 13, 16, 2.1.3, 20 respectively. Using the allocation matrix we get the actuator. Then log the trajectories in pickle format.

3.1 Effect of Parameters

We observe the following effect on system performance by these parameters.

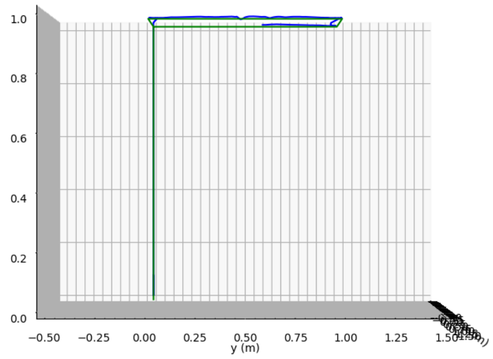
- Kp and Kd: As in the equations for calculating forces we consider the PD control an increase in these gains, a aggressive manoeuvre are observed as the it is increases the roll and pitch angles.
- λ : After the reaching phase, an increase in λ values converges the system faster.
- k: The gains K directly affect the system performance as its increased.
- α We use boundary layer to avoid chattering.

Final control tuning parameters.

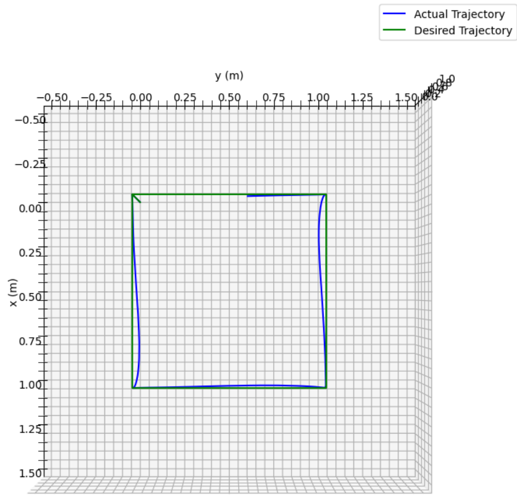
- For F_x we take Kp = 50 and Kd = 15 similarly for F_y we take Kp = 30 and Kd = 10
- We tune $\lambda = [7, 9, 9, 5]$
- K = [12, 370, 350, 5]

4 Part 4: Results visualising tracked trajectory.

Please find the video link. Here we discuss the performance of controller for the visualised trajectories in different perspective for better understanding. The Fig. 13a shows the Altitude control for the robot for about 1m which was the desired waypoint for which we generate the reference trajectory. The figure besides Fig. 13b shows the XY plane with the tracked trajectories. Please check the Fig. 14 it illustrates the visualized plot of trajectory tracking.



(a) Side view.



(b) Top view .

Figure 13: Orthographic view of tracked trajectories. Fig. 13a shows the altitude control and 13b shows the attitude control.

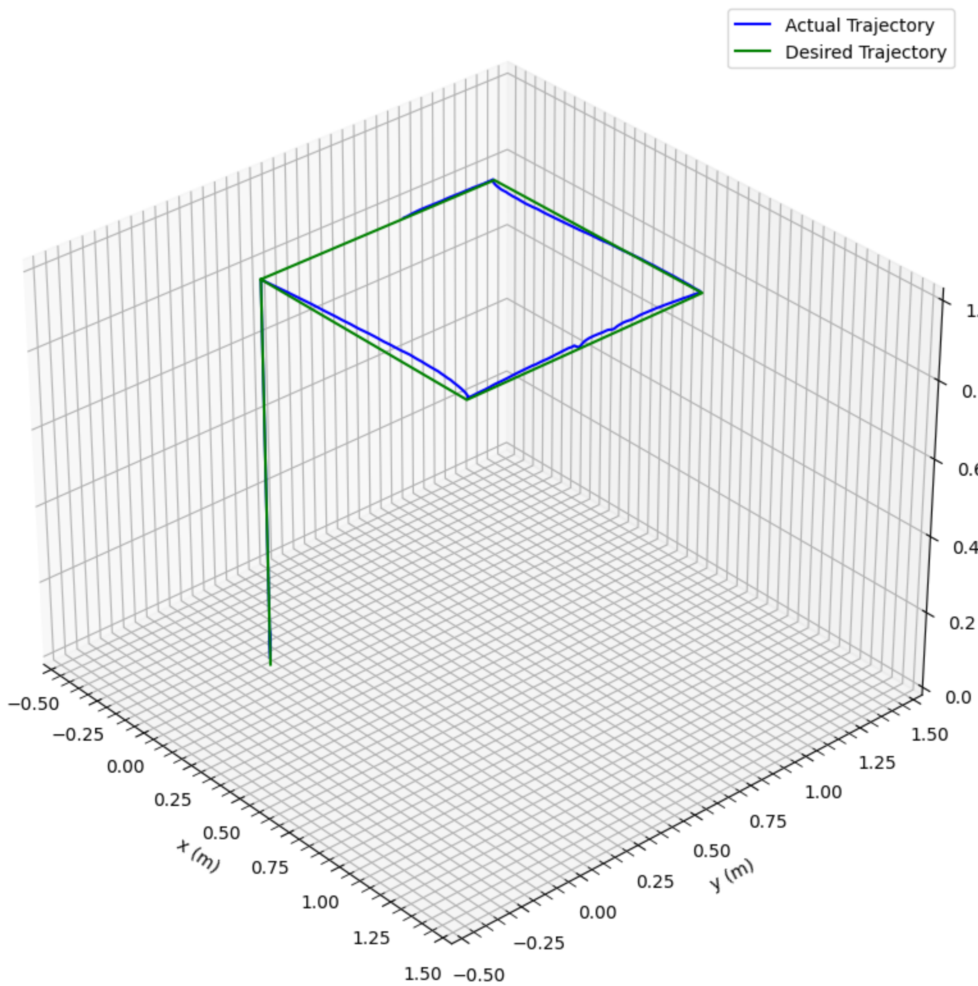


Figure 14: 3D plot of actual trajectory over reference trajectory