Robotics engineering

Project Report - Fall 2022

Trajectory Tracking for Quadroror using Slinding Mode Control

Chinmay kate,
Saurabh Kashid

Guided by
Prof. Siavash Farzan

Problem

Design a sliding mode controller for altitude and attitude control of the Crazyflie 2.0 to enable the quadrotor to track desired trajectories and visit a set of desired waypoints.

Trajectory Generation

Generate Quintic (5th order) polynomial trajectories for the position, velocity and acceleration for translation coordinates X,Y and Z for the Crazyflie quadrotor.

Given Waypoints are:
- p0 = (0, 0, 0) to p1 = (0, 0, 1) in 5 seconds
- p1 = (0, 0, 1) to p2 = (1, 0, 1) in 15 seconds
- p2 = (1, 0, 1) to p3 = (1, 1, 1) in 15 seconds
- p3 = (1, 1, 1) to p4 = (0, 1, 1) in 15 seconds
- p4 = (0, 1, 1) to p5 = (0, 0, 1) in 15 seconds

Solution:

The Physical Parameters of crazyflie 2.0 hardware platform listed below in table.

Parameter	*Symbol*	*Value*
Quadrotor mass	m	27 g
Quadrotor arm length	l	46 mm
Quadrotor inertia along x-axis	I_x	16.571710×10^{-6} kg $\cdot$ m^2
Quadrotor inertia along y-axis	I_y	16.571710×10^{-6} kg $\cdot$ m^2
Quadrotor inertia along z-axis	I_z	29.261652×10^{-6} kg $\cdot$ m^2
Propeller moment of inertia	I_p	12.65625×10^{-8} kg $\cdot$ m^2
Propeller thrust factor	k_F	1.28192×10^{-8} N $\cdot$ s^2
Propeller moment factor	k_M	5.964552×10^{-3} m
Rotor maximum speed	ω_{max}	2618 rad/s
Rotor minimum speed	ω_{min}	0 rad/s

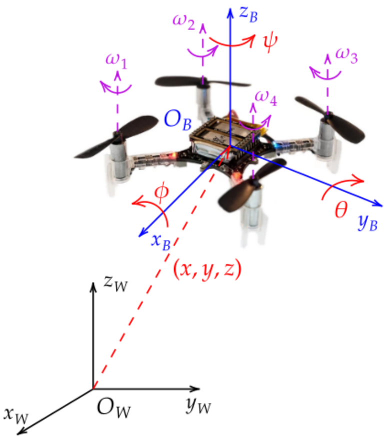

Figure 1: Crazyflie in the body O_b and the world O_w coordinate frames. The angular velocities of each rotor/propeller and the generalized coordinates x, y, z, ϕ, θ, ψ are depicted.)

Writing Quintic Polynomial expression:

$$\begin{bmatrix} 1 & t_0 & {t_0}^2 & {t_0}^3 & {t_0}^4 & {t_0}^5 \\ 0 & 1 & 2\,t_0 & 3\,{t_0}^2 & 4\,{t_0}^3 & 5\,{t_0}^4 \\ 0 & 0 & 2 & 6\,t_0 & 12\,{t_0}^2 & 20\,{t_0}^3 \\ 1 & \mathrm{tf} & \mathrm{tf}^2 & \mathrm{tf}^3 & \mathrm{tf}^4 & \mathrm{tf}^5 \\ 0 & 1 & 2\,\mathrm{tf} & 3\,\mathrm{tf}^2 & 4\,\mathrm{tf}^3 & 5\,\mathrm{tf}^4 \\ 0 & 0 & 2 & 6\,\mathrm{tf} & 12\,\mathrm{tf}^2 & 20\,\mathrm{tf}^3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ x_f \\ \dot{x}_f \\ \ddot{x}_f \end{bmatrix}$$

Here t_0 and t_f are the start and finish time of the trajectory. $a_0\,, a_1\,, a_2\,, a_3\,, a_4\,, a_5$ represent the parameters of the quintic trajectory. $\dot{x}_0\,, \dot{x}_f\,, \ddot{x}_0\,, \ddot{x}_f$ will be zero and thus above equation will be evaluated.

Trajectory 1: From p0 $(0, 0, 0)$ to p1 $(0, 0, 1)$ in 5 seconds

$$x_d = 0 \qquad y_d = 0 \qquad z_d = \frac{6\,t^5}{3125} - \frac{3\,t^4}{125} + \frac{2\,t^3}{25}$$

$$\dot{x}_d = 0 \qquad \dot{y}_d = 0 \qquad \dot{z}_d = \frac{6\,t^4}{625} - \frac{12\,t^3}{125} + \frac{6\,t^2}{25}$$

$$\ddot{x}_d = 0 \qquad \ddot{y}_d = 0 \qquad \ddot{z}_d = \frac{24\,t^3}{625} - \frac{36\,t^2}{125} + \frac{12\,t}{25}$$

Trajectory 2: From p1 $(0, 0, 1)$ to p2 $(1, 0, 1)$ in 15 seconds

$$x_d = \frac{1166016415523369\,t^5}{147573952589676412928} - \frac{t^4}{2025} + \frac{22\,t^3}{2025} - \frac{8\,t^2}{81} - \frac{47}{81} \qquad y_d = 0 \qquad z_d = 1$$

$$\dot{x}_d = \frac{5830082077616845\,t^4}{147573952589676412928} - \frac{4\,t^3}{2025} + \frac{22\,t^2}{675} - \frac{16\,t}{81} + \frac{32}{81} \qquad \dot{y}_d = 0 \qquad \dot{z}_d = 0$$

$$\ddot{x}_d = \frac{5830082077616845\,t^3}{36893488147419103232} - \frac{4\,t^2}{675} + \frac{44\,t}{675} - \frac{16}{81} \qquad \ddot{y}_d = 0 \qquad \ddot{z}_d = 0$$

Trajectory 3: From p2 $(1,0,1)$ to p3 $(1,1,1)$ in 15 seconds

$$
\begin{aligned}
x_d &= 1 & y_d &= \frac{1166016415523369\,t^5}{147573952589676412928} - \frac{11\,t^4}{10125} + \frac{118\,t^3}{2025} + \frac{616\,t^2}{405} + \frac{1568\,t}{81} + \frac{7808}{81} & z_d &= 1 \\
\dot{x}_d &= 0 & \dot{y}_d &= \frac{5830082077616845\,t^4}{147573952589676412928} - \frac{44\,t^3}{10125} + \frac{118\,t^2}{675} - \frac{1232\,t}{405} + \frac{1569}{81} & \dot{z}_d &= 0 \\
\ddot{x}_d &= 0 & \ddot{y}_d &= \frac{5830082077616845\,t^3}{36893488147419103232} - \frac{44\,t^2}{3375} + \frac{236\,t}{675} - \frac{1232}{405} & \ddot{z}_d &= 0
\end{aligned}
$$

Trajectory 4: From p3 $(1,1,1)$ to p4 $(0,1,1)$ in 15 seconds

$$
\begin{aligned}
x_d &= -\frac{1166016415523369\,t^5}{147573952589676412928} + \frac{17\,t^4}{10125} - \frac{286\,t^3}{2025} + \frac{476\,t^2}{81} - \frac{9800\,t}{81} + \frac{80000}{81} & y_d &= 1 & z_d &= 1 \\
\dot{x}_d &= -\frac{5830082077616845\,t^4}{147573952589676412928} + \frac{68\,t^3}{10125} - \frac{286\,t^2}{675} + \frac{952\,t}{81} - \frac{9800}{81} & \dot{y}_d &= 0 & \dot{z}_d &= 0 \\
\ddot{x}_d &= -\frac{5830082077616845\,t^3}{36893488147419103232} + \frac{68\,t^2}{3375} - \frac{572\,t}{675} + \frac{952}{81} & \ddot{y}_d &= 0 & \ddot{z}_d &= 0
\end{aligned}
$$

Trajectory 5: From p4 $(0,1,1)$ to p5 $(0,0,1)$ in 15 seconds

$$
\begin{aligned}
x_d &= 0 & y_d &= -\frac{1166016415523369\,t^5}{147573952589676412928} + \frac{23\,t^4}{10125} - \frac{526\,t^3}{2025} + \frac{1196\,t^2}{81} - \frac{33800\,t}{81} + \frac{1289825552459047}{274877906944} & z_d &= 1 \\
\dot{x}_d &= 0 & \dot{y}_d &= -\frac{23320328310467385\,t^4}{590295810358705651712} + \frac{92\,t^3}{10125} - \frac{526\,t^2}{675} + \frac{2392\,t}{81} - \frac{33800}{81} & \dot{z}_d &= 0 \\
\ddot{x}_d &= 0 & \ddot{y}_d &= -\frac{23320328310467385\,t^3}{147573952589676412928} + \frac{92\,t^2}{3375} - \frac{1052\,t}{675} + \frac{2392}{81} & \ddot{z}_d &= 0
\end{aligned}
$$

The trajectories described by the above equations can be visualized as follows:

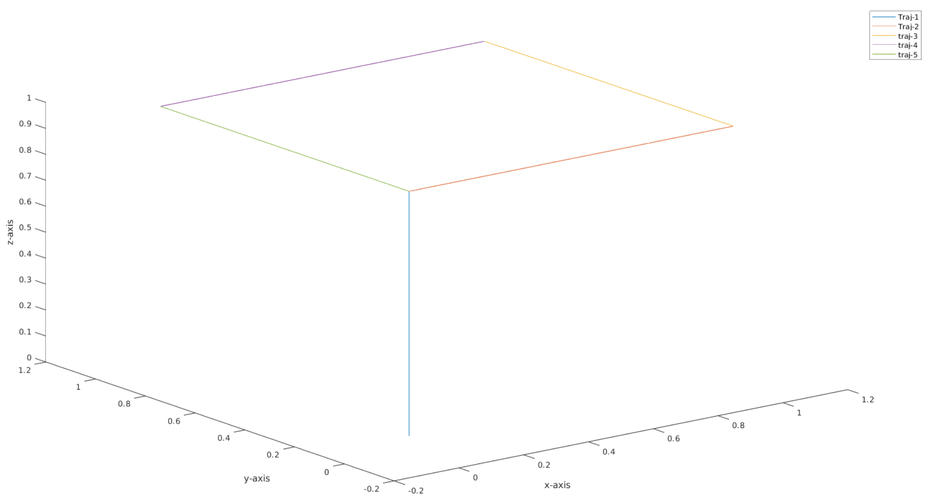

Figure 2: Desired Position Trajectories in 3D

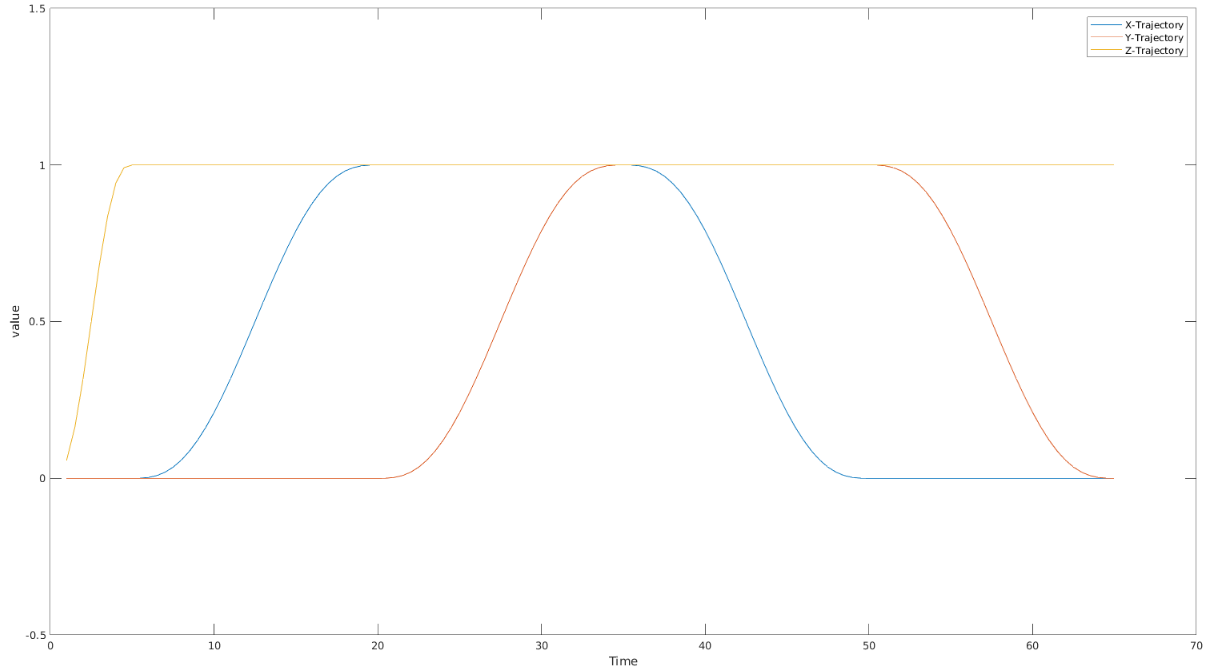

Figure 3: Desired Position 2D Trajectories

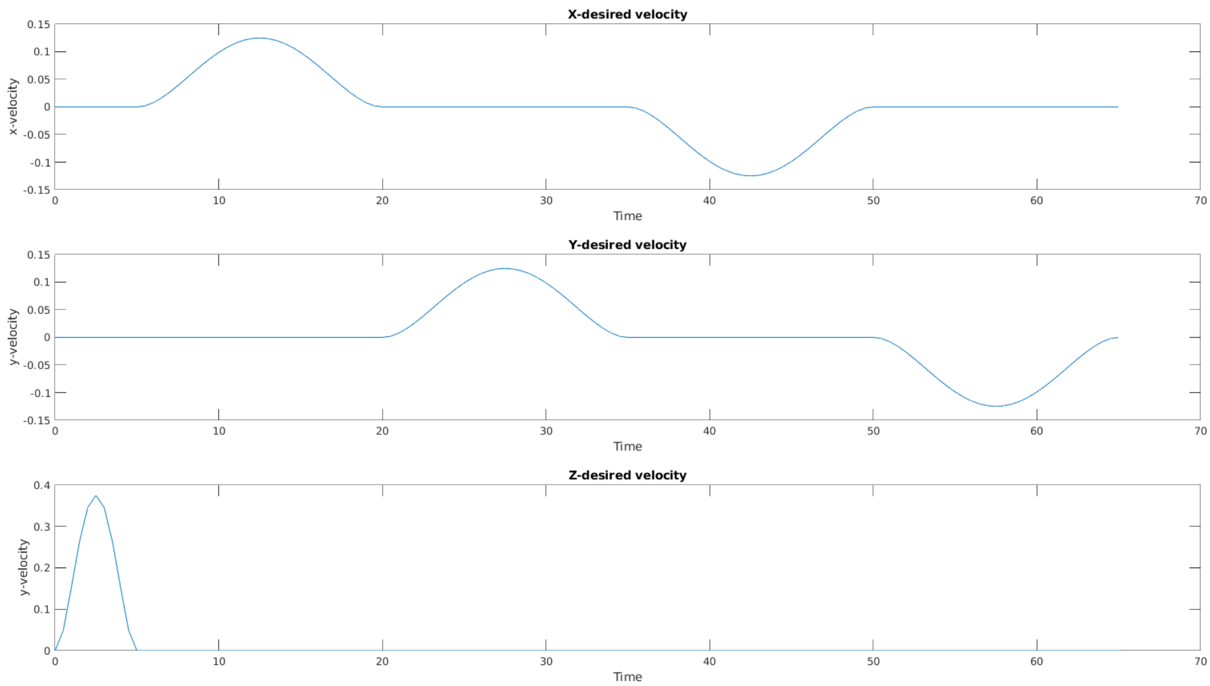

Figure 4: Desired Velocity Trajectories

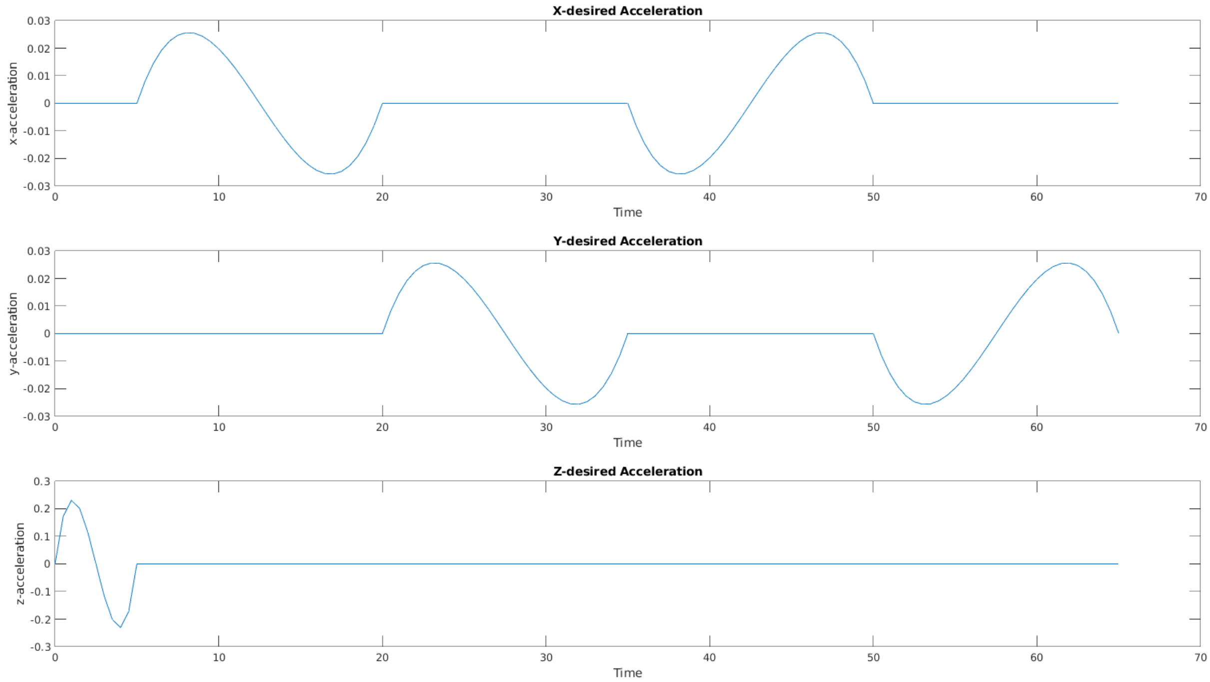

Figure 5: Desired Acceleration Trajectories

Sliding Mode control design

Equation of Motion:

Generalised coordinates for the quadrotor model is defined as follows:

$$q = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}$$

where, x, y, z are the co-ordinates, and ϕ, θ, ψ are the roll , pitch and yaw angles. The control inputs on the system are simply considered as: $u = \begin{bmatrix} u1 & u2 & u3 & u4 \end{bmatrix}$

Equation of Motion of the quadrotor:

$$
\begin{aligned}
\ddot{x} &= \frac{1}{m}\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)u_1 \\
\ddot{y} &= \frac{1}{m}\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right)u_1 \\
\ddot{z} &= \frac{1}{m}\left(\cos\phi\cos\theta\right)u_1 - g \\
\ddot{\phi} &= \dot{\theta}\,\dot{\psi}\,\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\,\Omega\,\dot{\theta} + \frac{1}{I_x}\,u_2 \\
\ddot{\theta} &= \dot{\phi}\,\dot{\psi}\,\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\,\Omega\,\dot{\phi} + \frac{1}{I_y}\,u_3 \\
\ddot{\psi} &= \dot{\phi}\,\dot{\theta}\,\frac{I_x - I_y}{I_z} + \frac{1}{I_z}\,u_4
\end{aligned}
$$

Figure 6: Equation of motion for Crazyflie Quadroter

The four decoupled motions are translation along Z-axis, rolling about X-axis, pitching about Y-axis and yawing about Z-axis. Thus we model the system with four control inputs u_1 , u_2 , u_3 , u_4 for controlling the the altitude and attitude of the quadrotor.

Sliding surface 's' is defined such that $s = \dot{e} + e\,\lambda$;

Here 'e' denotes the error between the actual state and the desired state of the system and λ is a positive constant (a tuning parameter). The condition for a valid sliding surface is: $s.\dot{s} \leq -k.|s(t)|$. Where 'k' is a positive constant(a tuning variable). Each SMC is designed for all four control inputs u_1, u_2, u_3 and u_4.

Control input u_1 design:

To design the control input u_1, we work on the equation of translational acceleration in the Z direction.

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

From the previously developed quintic trajectories, we can obtain the desired translational position z_d, velocity $\dot{z}_d$ and acceleration $\ddot{z}_d$. The error between the quadrotor's actual position and the desired position in the z direction is denoted by an error term $e_z = z - z_d$. Taking time derivatives of this error term gives the errors in velocity and acceleration along Z axis as follows:

$$
\begin{aligned}
\dot{e}_z &= \dot{z} - \dot{z}_d \\
\ddot{e}_z &= \ddot{z} - \ddot{z}_d
\end{aligned}
$$

Sliding surface:

$$
\begin{aligned}
s_z &= \dot{e}_z + \lambda_z\,e_z = (\dot{z} - \dot{z}_d) + \lambda_z\,e_z \\
\dot{s}_z &= (\ddot{z} - \ddot{z}_d + \lambda_z\,(\dot{z} - \dot{z}_d))
\end{aligned}
$$

Sliding condition :

$(s_z\,\dot{s}_z = s_z\,(\ddot{z} - \ddot{z}_d + \lambda_z\,(\dot{z} - \dot{z}_d))) \leq -k_z.|s_z(t)|$

$$s_z\,\dot{s}_z = s_z\left(\frac{u_1\,\cos(\phi)\,\cos(\theta)}{m} - g - \ddot{z}_d + \lambda_z\,(\dot{z} - \dot{z}_d)\right)$$

Control u_1 is designed as:

$$u_1 = \frac{(g + \ddot{z}_d - \lambda_z\, \dot{e}_z)}{\frac{\cos(\phi)\, \cos(\theta)}{m}} - \frac{k_z\, sign(s_z)}{\frac{\cos\phi\, \cos\theta}{m}}$$

In order to tackle the issue of chattering, $sign(s_z)$ is replaced with a saturation function $sat(\frac{s_z}{\phi_z})$. Where ϕ_z is the boundary layer defined around the sliding surface and it is a tuning variable.

$$\textbf{\textit{Control Input :}}\ u_1 = \frac{m\,(g + \ddot{z}_d - \lambda_z\, \dot{e}_z - k_z\, sat(\frac{s_z}{\phi_z})}{\cos\phi\, \cos\theta}$$

Control input u_2 design:

To design the control input u_2, we work on the equation of angular acceleration about X-axis ($\ddot{\phi}$).

$$\ddot{\phi} = \frac{\dot{\theta}\, \dot{\psi}\, (I_y - I_z)}{I_x} - \frac{I_p\, \Omega\, \dot{\theta}}{I_x} + \frac{u_2}{I_x}$$

The desired position trajectories in X and Y direction needs to be converted into desired pitch and roll angles respectively. Term $\Omega = \omega_1 - \omega_2 + \omega_3 - \omega_4$, where ω terms represent the respective rotor RPMs. The desired roll (ϕ) and pitch (θ) trajectories are derived as follows:

1. We first calculate the force required at each time to achieve the desired trajectory in the X direction.

$$F_x = m(-k_p(x - x_d) - k_d(\dot{x} - \dot{x}_d) + \ddot{x}_d)$$

2. Similarly, force required to track the trajectory in Y-direction is given as:

$$F_y = m(-k_p(y - y_d) - k_d(\dot{y} - \dot{y}_d) + \ddot{y}_d)$$

   Here k_p and k_d are tuning parameters.

3. The desired roll and pitch angles are then determined using these forces.

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

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

Now, we have the desired roll (ϕ) angle, comparing this with the actual roll angle of the drone we can form an error term $e_\phi = \phi - \phi_d$. Taking time derivatives of this error term gives the errors in the angular velocity ($\dot{\phi}$)and angular acceleration ($\ddot{\phi}$) about the X-axis.

$$\dot{e}_\phi = \dot{\phi} - \dot{\phi}_d$$
$$\ddot{e}_\phi = \ddot{\phi} - \ddot{\phi}_d$$

Sliding surface:

$$s_\phi = \dot{e}_\phi + \lambda_\phi\, e_\phi = (\dot{\phi} - \dot{\phi}_d) + \lambda_\phi\, e_\phi$$
$$\dot{s}_\phi = \left(\ddot{\phi} - \ddot{\phi}_d + \lambda_\phi \left(\dot{\phi} - \dot{\phi}_d\right)\right)$$

Sliding Condition :

$s_\phi\,\dot{s}_\phi = s_\phi\left(\ddot{\phi} - \ddot{\phi}_d + \lambda_\phi\left(\dot{\phi} - \dot{\phi}_d\right)\right) \leq -k_\phi.|s_\phi(t)|$

$$s_\phi\,\dot{s}_\phi = s_\phi\left(\frac{\dot{\theta}\,\dot{\psi}\,(I_y - I_z)}{I_x} - \frac{I_p\,\Omega\,\dot{\theta}}{I_x} + \frac{u_2}{I_x} - \ddot{\phi}_d + \lambda_\phi\left(\dot{\phi} - \dot{\phi}_d\right)\right)$$

Control u_2 is designed as:

$$u_2 = -\left(\frac{\dot{\theta}\,\dot{\psi}\,(I_y - I_z)}{I_x} - \frac{I_p\,\Omega\,\dot{\theta}}{I_x} - \ddot{\phi}_d + \lambda_\phi\,(\dot{e}_\phi)\right)I_x - k_\phi\,sign(s_\phi)\,I_x$$

In order to tackle the issue of chattering, $sign(s_\phi)$is replaced with a saturation function $sat(\frac{s_\phi}{\phi_\phi})$. Where ϕ_ϕ is the boundary layer defined around the sliding surface and it is a tuning variable.

$$\textbf{\textit{Control Input :}}\ u_2 = -\left(\frac{\dot{\theta}\,\dot{\psi}\,(I_y - I_z)}{I_x} - \frac{I_p\,\Omega\,\dot{\theta}}{I_x} - \ddot{\phi}_d + \lambda_\phi\,(\dot{e}_\phi)\right)I_x - k_\phi\,sat(\frac{s_\phi}{\phi_\phi})\,I_x$$

The values for $\dot{\phi}_d$, $\ddot{\phi}_d$ are equal to zero.

Control input u_3 design :

To design the control input u_3, we work on the equation of angular acceleration about Y-axis ($\ddot{\theta}$).

$$\ddot{\theta} = \frac{\dot{\phi}\,\dot{\psi}\,(I_z - I_x)}{I_y} + \frac{I_p\,\Omega\,\dot{\phi}}{I_y} + \frac{u_3}{I_y}$$

The desired values for θ are derived using the given relation:

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

Now, we have the desired pitch (θ), comparing this with the actual pitch of the drone we can form an error term $e_\theta = \theta - \theta_d$. Taking time derivatives of this error term gives the errors in the angular velocity ($\dot{\theta}$)and angular acceleration ($\ddot{\theta}$) about the Y-axis as follows:

$$\dot{e}_\theta = \dot{\theta} - \dot{\theta}_d$$
$$\ddot{e}_\theta = \ddot{\theta} - \ddot{\theta}_d$$

Sliding surface:

$$s_\theta = \dot{e}_\theta + \lambda_\theta\,e_\theta = (\dot{\theta} - \dot{\theta}_d) + \lambda_\theta\,e_\theta$$
$$\dot{s}_\theta = \left(\ddot{\theta} - \ddot{\theta}_d + \lambda_\theta\left(\dot{\theta} - \dot{\theta}_d\right)\right)$$

Sliding Condition :

$s_\theta\,\dot{s}_\theta = s_\theta\left(\ddot{\theta} - \ddot{\theta}_d + \lambda_\theta\left(\dot{\theta} - \dot{\theta}_d\right)\right) \leq -k_\theta.|s_\theta(t)|$

$$s_\theta\,\dot{s}_\theta = s_\theta\left(\frac{\dot{\phi}\,\dot{\psi}\,(I_z - I_x)}{I_y} + \frac{I_p\,\Omega\,\dot{\phi}}{I_y} + \frac{u_3}{I_y} - \ddot{\theta}_d + \lambda_\theta\left(\dot{\theta} - \dot{\theta}_d\right)\right)$$

Control u_3 is designed as:

$$u_3 = -\left(\frac{\dot{\phi}\,\dot{\psi}\,(I_z - I_x)}{I_y} + \frac{I_p\,\Omega\,\dot{\phi}}{I_y} - \ddot{\theta}_d + \lambda_\theta\,\dot{e}_\theta\right) I_y - k_\theta\,sign(s_\theta)\,I_y$$

In order to tackle the issue of chattering, $sign(s_\theta)$ is replaced with a saturation function $sat(\frac{s_\theta}{\phi_\theta})$. Where ϕ_θ is the boundary layer defined around the sliding surface and it is a tuning variable.

Control Input :

$$u_3 = -\left(\frac{\dot{\phi}\,\dot{\psi}\,(I_z - I_x)}{I_y} + \frac{I_p\,\Omega\,\dot{\phi}}{I_y} - \ddot{\theta}_d + \lambda_\theta\,\dot{e}_\theta\right) I_y - k_\theta\,sat(\frac{s_\theta}{\phi_\theta})\,I_y$$

The values for $\dot{\theta}_d$, $\ddot{\theta}_d$ are given to be zero.

Control input u_4 design :

To design the control input u_4 we work on the equation of angular acceleration about Z-axis ($\ddot{\psi}$).

$$\ddot{\psi} = \frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z} + \frac{u_4}{I_z}$$

The desired yaw angle (ψ_d), velocity ($\dot{\psi}_d$) and acceleration ($\ddot{\psi}_d$) are all zero, i.e. no desired yaw motion. Now, we have the desired yaw (ψ_d), comparing this with the actual yaw of the drone we can form an error term $e_\psi = \psi - \psi_d$. Taking time derivatives of this error term gives the errors in the angular velocity ($\dot{\psi}$) and angular acceleration ($\ddot{\psi}$) about the Z-axis as follows:

$$\begin{aligned} \dot{e}_\psi &= \dot{\psi} - \dot{\psi}_d = \dot{\psi} - 0 = \dot{\psi} \\ \ddot{e}_\psi &= \ddot{\psi} - \ddot{\psi}_d = \ddot{\psi} - 0 = \ddot{\psi} \end{aligned}$$

Sliding surface:

$$\begin{aligned} s_\psi &= \dot{e}_\psi + \lambda_\psi\,e_\psi = (\dot{\psi} - \dot{\psi}_d) + \lambda_\psi\,e_\psi \\ \dot{s}_\psi &= \ddot{\psi} - \ddot{\psi}_d + \lambda_\psi\left(\dot{\psi} - \dot{\psi}_d\right) \end{aligned}$$

Sliding Condition :

$$s_\psi\,\dot{s}_\psi = s_\psi\left(\ddot{\psi} - \ddot{\psi}_d + \lambda_\psi\left(\dot{\psi} - \dot{\psi}_d\right)\right) \leq -k_\psi.|s_\psi(t)|$$

$$s_\psi\,\dot{s}_\psi = s_\psi\left(\frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z} + \frac{u_4}{I_z} - \ddot{\psi}_d + \lambda_\psi\left(\dot{\psi} - \dot{\psi}_d\right)\right)$$

Control u_4 is designed as:

$$u_4 = -\left(\frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z} - \ddot{\psi}_d + \lambda_\psi\,\dot{e}_\psi\right) I_z - k_\psi\,sign(s_\psi)\,I_z$$

In order to tackle the issue of chattering, The $sign(s_\psi)$ is replaced with a saturation function $sat(\frac{s_\psi}{\phi_\psi})$. Where ϕ_ψ is the boundary layer defined around the sliding surface and it is a tuning variable.

$$\textbf{\textit{Control Input :}}\ u_4 = -\left(\frac{\dot\phi\,\dot\theta\,(I_x - I_y)}{I_z} - \ddot\psi_d + \lambda_\psi\,\dot e_\psi\right) I_z - k_\psi\,sat(\frac{s_\psi}{\phi_\psi})\,I_z$$

The desired values for ψ_d, $\dot\psi_d$ and $\ddot\psi_d$ are zero.

Tuning Parameters:

Sr. No.	K	Value
1	k_p	60
2	k_d	5
3	k_z	15
4	k_ϕ	150
5	k_θ	150
6	k_ψ	25

Sr. No.	Lambda	Value
1	λ_z	5
2	λ_ϕ	12
3	λ_θ	12
4	λ_ψ	5

Sr. No.	Boundary	Value
1	Boundary ϕ_z	1
2	Boundary ϕ_ϕ	1
3	Boundary ϕ_θ	1
4	Boundary ϕ_ψ	1

Discussion on effects of tuning parameters on System Response

It affects the characteristics of the controller and thus directly impact the system response. We want the quadrotor to trace a desired trajectory by keeping errors between the actual and desired states within acceptable bounds.

All the tuning parameter and it's characterstics are explained below:

1. **K :** Depending on K system takes times to converge to the sliding surface.Lower the value more time to convrse to sliding surface. Higher the value lesser the time but it might cause an overshoot.
2. **Lambda :** This value decides the time required converse to the desired trajectory. Lower the lambda more time it will take to converse. Higher value takes less time but it might cause overshoot.
3. **Boundary :** To avoid the chattering issue in the controller input we use this boundary. but higher the boundary higher the deviation from the desired trajectory and vise versa.
4. As discussed above each value contribute to the relative action. like λ_θ contribute to the time to converse to desire pitch. K_p and K_d contribute to altitude of the quadrotor.

Discussion on Code Developed in Part 3:

1. We first designed the desired quintic trajectories separately in matlab file named $traj_gen.m$ in terms of the time t. we hard coded desired position, velocity and acceleration in X, Y and Z direction.
2. Next for designing the controller we derived the $S\dot{S}$, then to satisfy the sliding condition we added $kSgns$ to the equation. we equate this equation equal to zero and solved for u. We designed all u's in matlab file named *smc.m* in symbolic form and used them in the python script.
3. When we run the *code.py* file all the states in x,y and z direction saved in the file named *log.pkl* to plot that states.
4. To compare desired trajectory and actual trajectory, we run *visualize.py* file which reads *log.pkl* file and plot desired and actual trajectories.

Discussion on Performance of Controller

The quadrotor was tuned with gains listed in the above 'Tuning Parameters' section. The performance 3D plot is shown below. From the graph we can infer the following:

3D Plot of Actual vs. Desired Trajectory

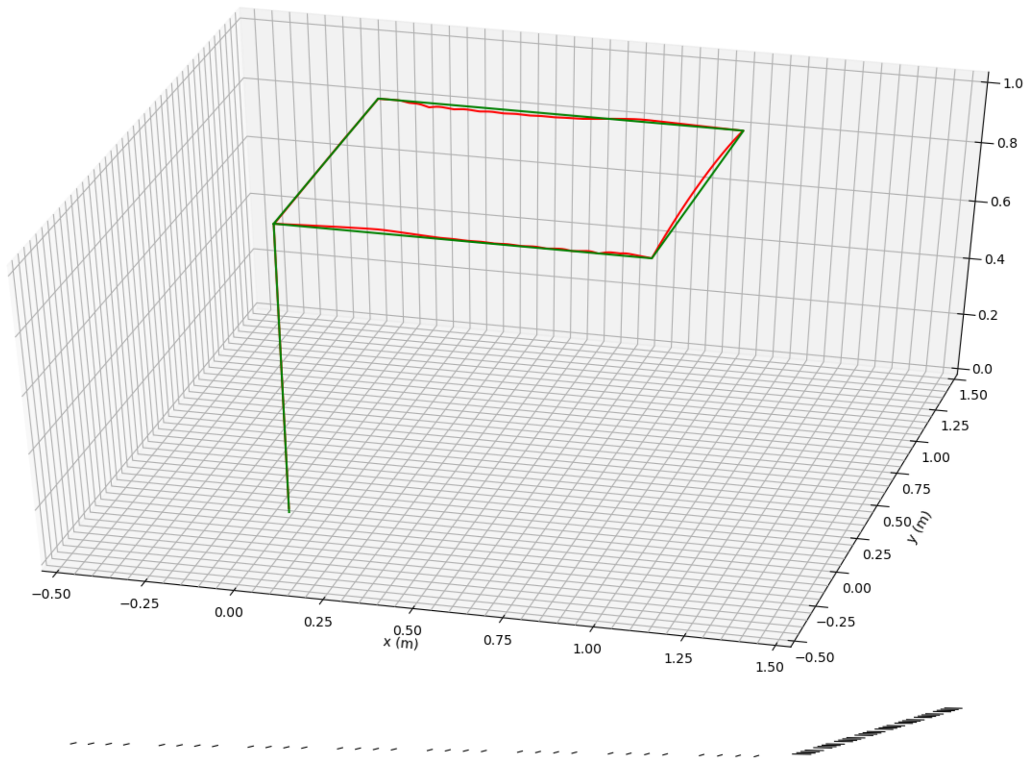

Figure 7: 3D Plot of the actual(Red) and Desired(Green) trajectories

a) The quadrotor was able to trace all the 5 paths successfully.
b) There is some deviation in the trajectory this is due the external disturbance in the gazebo, as it create the simulation environment as real world.
c) The deviation might further reduced by tuning the parameters. but the small deviation is acceptable.

Submitted files

a) *trajgen.m* File used to generate and plot desired trajectory.
b) *smc.m* File used to solve the controller
c) *code.py* This is the main python file to simulate the environment
d) *visualize.py* To plot the desired and actual trajectories.