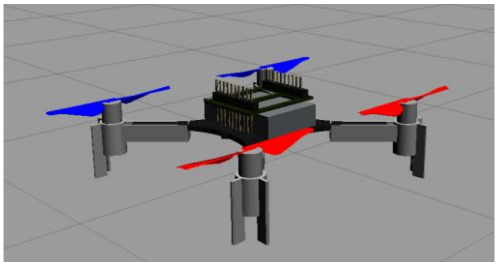
Robust Trajectory Tracking for Quadrotor UAVs using Sliding Mode Control

Problem Statement:



The objective of this project is to develop a robust controller to enable a quadrotor to track desired trajectories in the presence of external disturbances. The quadrotor being used is the Crazyflie 2.0. It has 4 DC motors so we will have motor speeds as inputs to the simulation.

Dynamic Model

$$\mathrm{q} = \begin{pmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{pmatrix} \qquad \dot{q} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \qquad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}$$

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

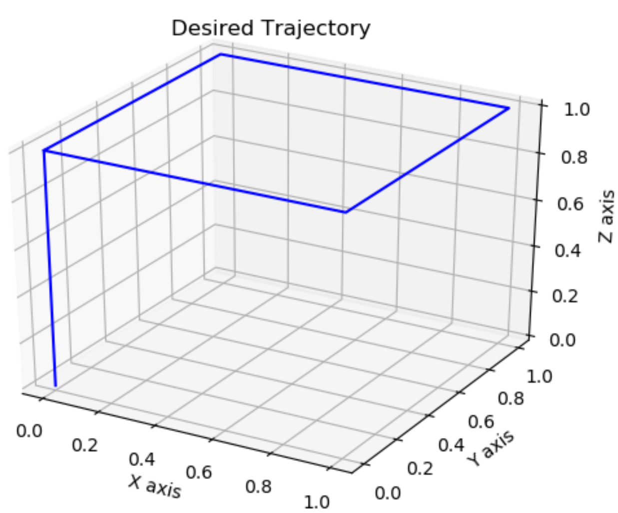
$$\ddot{q} = f(q,\dot{q}) + g(q,\dot{q})u$$

$$f(q,\dot q) = \begin{pmatrix} 0 \\ 0 \\ -g \\ \dfrac{\dot\psi\,\dot\theta\,(I_y - I_z)}{I_x} - \dfrac{I_p\,\Omega\,\dot\theta}{I_x} \\ \dfrac{I_p\,\Omega\,\dot\phi}{I_y} - \dfrac{\dot\phi\,\dot\psi\,(I_x - I_z)}{I_y} \\ \dfrac{\dot\phi\,\dot\theta\,(I_x - I_y)}{I_z} \end{pmatrix} \qquad g(q,\dot q) = \begin{pmatrix} \dfrac{\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)}{m} & 0 & 0 & 0 \\ -\dfrac{\cos(\psi)\sin(\phi) - \cos(\phi)\sin(\psi)\sin(\theta)}{m} & 0 & 0 & 0 \\ \dfrac{\cos(\phi)\cos(\theta)}{m} & 0 & 0 & 0 \\ 0 & \dfrac{1}{I_x} & 0 & 0 \\ 0 & 0 & \dfrac{1}{I_y} & 0 \\ 0 & 0 & 0 & \dfrac{1}{I_z} \end{pmatrix}$$

Physical Parameters

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^{-6} 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

Part 1 - Trajectory Generation



Part 2 - Sliding Mode Control

Our objective is to design a boundary layer-based sliding model controller for the altitude and attitude of the quadrotor. Hence, we will deal with the cartesian coordinate z and the three spherical coordinates representing roll, pitch and yaw.

Step 1 - Define the sliding surface

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

$$e = \begin{pmatrix} z - z_d \\ \phi - \phi_d \\ \theta - \theta_d \\ \psi - \psi_d \end{pmatrix} = \begin{pmatrix} z - z_d \\ \phi - \phi_d \\ \theta - \theta_d \\ \psi \end{pmatrix} \qquad \dot{e} = \begin{pmatrix} \dot{z} - \dot{z}_d \\ \dot{\phi} - \dot{\phi}_d \\ \dot{\theta} - \dot{\theta}_d \\ \dot{\psi} - \dot{\psi}_d \end{pmatrix} = \begin{pmatrix} \dot{z} - \dot{z}_d \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} \dot{z} - \dot{z}_d + \lambda\left(z - z_d\right) \\ \dot{\phi} + \lambda\left(\phi - \phi_d\right) \\ \dot{\theta} + \lambda\left(\theta - \theta_d\right) \\ \dot{\psi} + \lambda\,\psi \end{pmatrix}$$

Step 2 - Define the control input

In equation of sliding surface, we measure the values of z, phi, theta, psi, z_dot, phi_dot, theta_dot, and psi_dot as states. Lambda is a tuning parameter which is generally set to 1. The values of z_d, and z_dot_d are obtained from the generated trajectory.

The control input can be calculated using

$$u = \frac{-f(q, \dot{q}) + \ddot{q}_d - \lambda \dot{e} + u_r}{g(q, \dot{q})}$$

We consider only 4 states for controllers so the terms to be used in the above equation are as follows

$$f(q,\dot{q}) = \begin{pmatrix} -g \\ \frac{\dot{\psi}\,\dot{\theta}\,(I_y - I_z)}{I_x} - \frac{I_p\,\Omega\,\dot{\theta}}{I_x} \\ \frac{I_p\,\Omega\,\dot{\phi}}{I_y} - \frac{\dot{\phi}\,\dot{\psi}\,(I_x - I_z)}{I_y} \\ \frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z} \end{pmatrix} \qquad \ddot{q} = \begin{pmatrix} \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix}$$

$$g(q,\dot{q}) = \begin{pmatrix} \frac{\cos(\phi)\,\cos(\theta)}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{pmatrix}$$

The robust control term is calculates as $u_r = -(\rho + k)sat(\frac{s}{\phi_s})$ where rho = 0 as all model parameters are known. The value of phi_s defines the boundary and is set to 0.1 while k is a tuning parameter.

In this system, the values of phi_d and theta_d (desired values of roll and pitch) are not readily available. They need to be calculated by using the value of u_1.

Hence, we calculate the u_1 value using the above given equation (we need solution to the first row of u). Then we use the resulting u_1 value with given equations to get desired roll and pitch which can then be used to calculate the other control inputs.

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

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

Note: k_p and k_d are tuning parameters for PD control

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

The final control input equations are given as

$$u = \begin{pmatrix} \frac{\cos(\phi)\,\cos(\theta)}{m}\left(g + \ddot{z}_d - \lambda(\dot{z} - \dot{z}_d) - \mathrm{sat}\left(\frac{s_1}{\phi_s}\right)(k+\rho)\right) \\ -\frac{1}{I_x}\left(\lambda\,\dot{\phi} - \ddot{\phi}_d + \mathrm{sat}\left(\frac{s_2}{\phi_s}\right)(k+\rho) - \frac{I_p\,\Omega\,\dot{\theta}}{I_x} + \frac{\dot{\psi}\,\dot{\theta}\,(I_y - I_z)}{I_x}\right) \\ -\frac{1}{I_y}\left(\lambda\,\dot{\theta} - \ddot{\theta}_d + \mathrm{sat}\left(\frac{s_3}{\phi_s}\right)(k+\rho) + \frac{I_p\,\Omega\,\dot{\phi}}{I_y} - \frac{\dot{\phi}\,\dot{\psi}\,(I_x - I_z)}{I_y}\right) \\ -\frac{1}{I_z}\left(\lambda\,\dot{\psi} - \ddot{\psi}_d + \mathrm{sat}\left(\frac{s_4}{\phi_s}\right)(k+\rho) + \frac{\dot{\phi}\,\dot{\theta}\,(I_x - I_y)}{I_z}\right) \end{pmatrix}$$

where,

rho = 0, phi_ddot_d = theta_ddot_d = psi_ddot_d = 0

Using these equations, we calculate the value for u for each time step and use the allocation matrix to derive the final motor speeds.

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