RBE502 - Final Project
Robust Trajectory Tracking for Quadrotor UAVs using Sliding Mode Control

Gaddipati, Chaitanya Sriram
cgaddipati@wpi.edu

Baidya, Anoushka
abaidya@wpi.edu

May 3, 2023

1 Development and Implementation

1.1 Part 1: Trajectory Generation

The quintic polynomial equation for each joint angle is defined as

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{1}$$

where t is the time variable and $q_d(t)$ is the desired trajectory.
Taking the first derivative of the quintic polynomial equation to get the desired velocity trajectory. The first derivative of $q_d(t)$ is given as

$$\dot{q_d}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{2}$$

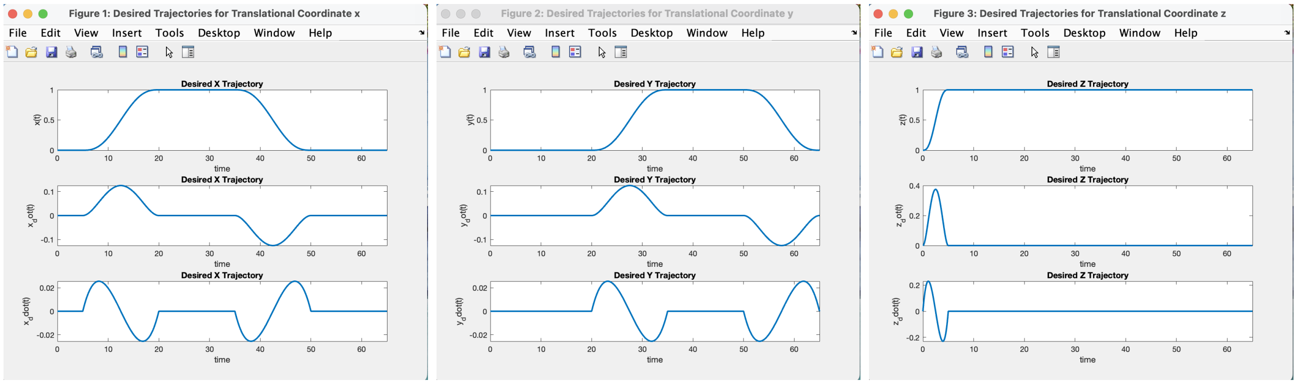
Taking the second derivative of the quintic polynomial equation to get the desired acceleration trajectory for the joint angles. The second derivative of $q_d(t)$ is given by

$$\ddot{q_d}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{3}$$

Solving the matrices to find the parameter coefficients $a_0, a_1, a_2, a_3, a_4, a_5$
A fifth order quintic polynomial defined for positions x,y and z. Substituting x, y and z final and initial as q_d. Linear equations were formulated by substituting the initial and final conditions for each section, followed by solving the equations for the coefficients of the polynomials. This is carried out for each of the 5 sections.

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix} \tag{4}$$

The coefficients can be solved by using equation $A = inv(T)q$.The final plots for the trajectories are shown in figure 1.



(a) Desired Trajectories -X (b) Desired Trajectories -Y (c) Desired Trajectories -Z

Figure 1: Desired Trajectories for Translational Coordinates

1.2 Part 2: Sliding Mode Control Law Formulation

For the Quadrotor, simplified equations of motion (assuming small angles) are given below:

$$\ddot{x} = \frac{1}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)u_1 \tag{5}$$

$$\ddot{y} = \frac{1}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)u_1 \tag{6}$$

$$\ddot{z} = \frac{1}{m}(\cos\phi\cos\theta)u_1 - g \tag{7}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\Omega\dot{\theta} + \frac{1}{I_x}u_2 \tag{8}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi} + \frac{1}{I_y}u_3 \tag{9}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \frac{1}{I_z}u_4 \tag{10}$$

$$\tag{11}$$

The Quadrotor is controlled through altitude z and the attitude containing roll, pitch and yaw angles ϕ, θ and ψ. So the sliding mode control law for the system is split into individual control laws for altitude and attitude variables.

1.2.1 Control law for altitude:

The equation of motion for altitude can be written in control affine form as shown below:

$$\ddot{z} = -g + \left[\frac{1}{m}(\cos\phi\cos\theta)\right]u_1 \tag{12}$$

The error is $e_1 = z - z_d$. Let the sliding surface be s_1.

$$s_1 = \dot{e_1} + \lambda_1 e_1 \tag{13}$$

This is a valid sliding surface since it satisfies the following conditions:

- $\dot{s_1} = \ddot{z} - \ddot{z}_d + \lambda_1\dot{e_1}$ contains the control input u_1
- As $s_1 \to 0$, the solution to the sliding surface is $e_1(t) = e^{-\lambda_1 t}$, which implies $e_1 \to 0$ and $\dot{e_1} \to 0$.

After choosing the sliding surface, the control law is formulated such that the sliding condition is satisfied.

$$s_1\dot{s_1} = s_1\Big[\ddot{z} - \ddot{z}_d + \lambda_1\dot{e_1}\Big] s_1\dot{s_1} = s_1\Big[-g + \Big[\frac{1}{m}(\cos\phi\cos\theta)\Big]u_1 - \ddot{z}_d + \lambda_1\dot{e_1}\Big] \tag{14}$$

Choose a u_1 such that the terms in the above equations are eliminated and also such that it contains an additional control variable u_r.

$$u_1 = \frac{m}{\cos\phi\cos\theta}(g + \ddot{z}_d - \lambda_1\dot{e_1} + u_r) \tag{15}$$

Substituting the above u_1 in the equation for $s_1\dot{s_1}$.

$$s_1\dot{s_1} = s_1 u_r \tag{16}$$

The above equation should satisfy the sliding condition.

$$s_1\dot{s_1} = s_1 u_r \leq -k_1|s_1| \tag{17}$$

To satisfy the condition u_r becomes:

$$u_r = -k_1 sat(\frac{s_1}{\phi_{b1}}) \tag{18}$$

Here $k_1 > 0$. The overall control law is:

$$u_1 = \frac{m}{\cos\phi\cos\theta}(g + \ddot{z}_d - \lambda_1\dot{e_1} - k_1 sat(\frac{s_1}{\phi_{b1}})) \tag{19}$$

1.2.2 Control laws for attitude:

The attitude contains the roll, pitch and yaw terms. Before proceeding with the control law design, the desired roll, pitch and yaw terms are calculated.

$$F_x = m(-k_p(x - x_d) - k_d(\dot{x} - \dot{x_d} + \ddot{x_d}) \tag{20}$$

$$F_y = F_x = m(-k_p(y - y_d) - k_d(\dot{y} - \dot{y_d} + \ddot{y_d}) \tag{21}$$

$$\theta_d = \arcsin\frac{F_x}{u_1} \tag{22}$$

$$\phi_d = \arcsin\frac{-F_y}{u_1} \tag{23}$$

For the problem, it is assumed that the desired yaw, desired angular velocity and angular accelerations are zero.

$$\psi_d = 0 \text{ and } \dot{\phi_d} = \dot{\theta_d} = \dot{\psi_d} = 0 \text{ and } \ddot{\phi_d} = \ddot{\theta_d} = \ddot{\psi_d} = 0 \tag{24}$$

With these values the control laws are designed. The equation of motion for the roll is as follows:

$$\ddot{\phi} = \Big[\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\Omega\dot{\theta}\Big] + [\frac{1}{I_x}]u_2 \tag{25}$$

In the above equation the Ω term is calculated using the angular velocities of the propeller from the previous time instance.
The error is $e_2 = \phi - \phi_d$. Let the sliding surface be s_2.

$$s_2 = \dot{e_2} + \lambda_2 e_2 \tag{26}$$

This is a valid sliding surface since it satisfies the following conditions:

- $\dot{s_2} = \ddot{\phi} - \ddot{\phi_d} + \lambda_2\dot{e_2}$ contains the control input u_2
- As $s_2 \to 0$, the solution to the sliding surface is $e_2(t) = e^{-\lambda_2 t}$, which implies $e_2 \to 0$ and $\dot{e_2} \to 0$.

After choosing the sliding surface, the control law is formulated such that the sliding condition is satisfied.

$$s_2\dot{s_2} = s_2\Big[\ddot{\phi} - \ddot{\phi_d} + \lambda_2\dot{e_2}\Big]s_2\dot{s_2} = s_2\Big[\Big[\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} - \frac{I_p}{I_x}\Omega\dot{\theta}\Big] + [\frac{1}{I_x}]u_2 - \ddot{\phi_d} + \lambda_2\dot{e_2}\Big] \tag{27}$$

Choose a u_2 such that the terms in the above equations are eliminated and also such that it contains an additional control variable u_r.

$$u_2 = I_x(-\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} + \frac{I_p}{I_x}\Omega\dot{\theta} + \ddot{\phi_d} - \lambda_2\dot{e_2} + u_r) \tag{28}$$

Substituting the above u_2 in the equation for $s_2\dot{s_2}$.

$$s_2\dot{s_2} = s_2u_r \tag{29}$$

The above equation should satisfy the sliding condition.

$$s_2\dot{s_2} = s_2u_r \leq -k_2|s_2| \tag{30}$$

To satisfy the condition u_r becomes:

$$u_r = -k_2 sat(\frac{s_2}{\phi_{b2}}) \tag{31}$$

Here $k_2 > 0$. The overall control law is:

$$u_2 = I_x(-\dot{\theta}\dot{\psi}\frac{I_y - I_z}{I_x} + \frac{I_p}{I_x}\Omega\dot{\theta} + \ddot{\phi}_d - \lambda_2\dot{e_2} - k_2 sat(\frac{s_2}{\phi_{b2}})) \tag{32}$$

Similarly calculating the control law for pitch. The equation of motion for the pitch is as follows:

$$\ddot{\theta} = \Big[\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi}\Big] + [\frac{1}{I_y}]u_3 \tag{33}$$

In the above equation the Ω term is calculated using the angular velocities of the propeller from the previous time instance.
The error is $e_3 = \theta - \theta_d$. Let the sliding surface be s_3.

$$s_3 = \dot{e_3} + \lambda_3 e_3 \tag{34}$$

This is a valid sliding surface since it satisfies the following conditions:

- $\dot{s_3} = \ddot{\theta} - \ddot{\theta_d} + \lambda_3\dot{e_3}$ contains the control input u_3
- As $s_3 \to 0$, the solution to the sliding surface is $e_3(t) = e^{-\lambda_3 t}$, which implies $e_3 \to 0$ and $\dot{e_3} \to 0$.

After choosing the sliding surface, the control law is formulated such that the sliding condition is satisfied.

$$s_3\dot{s_3} = s_3\Big[\ddot{\theta} - \ddot{\theta}_d + \lambda_3\dot{e_3}\Big] s_3\dot{s_3} = s_3\Big[\Big[\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} + \frac{I_p}{I_y}\Omega\dot{\phi}\Big] + [\frac{1}{I_y}]u_3 - \ddot{\theta}_d + \lambda_3\dot{e_3}\Big] \tag{35}$$

Choose a u_3 such that the terms in the above equations are eliminated and also such that it contains an additional control variable u_r.

$$u_3 = I_y(-\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} - \frac{I_p}{I_y}\Omega\dot{\phi} + \ddot{\theta}_d - \lambda_3\dot{e_3} + u_r) \tag{36}$$

Substituting the above u_3 in the equation for $s_3\dot{s_3}$.

$$s_3\dot{s_3} = s_3u_r \tag{37}$$

The above equation should satisfy the sliding condition.

$$s_3\dot{s_3} = s_3u_r \leq -k_3|s_3| \tag{38}$$

To satisfy the condition u_r becomes:

$$u_r = -k_3 sat(\frac{s_3}{\phi_{b3}}) \tag{39}$$

Here $k_3 > 0$. The overall control law is:

$$u_3 = I_y(-\dot{\phi}\dot{\psi}\frac{I_z - I_x}{I_y} - \frac{I_p}{I_y}\Omega\dot{\phi} + \ddot{\theta}_d - \lambda_3\dot{e_3} - k_3 sat(\frac{s_3}{\phi_{b3}})) \tag{40}$$

Next the control law for the yaw is formed. The equation of motion for yaw is as shown below.

$$\ddot{\psi} = \Big[\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z}\Big] + [\frac{1}{I_z}]u_4 \tag{41}$$

The error is $e_4 = \psi - \psi_d$. Let the sliding surface be s_4.

$$s_4 = \dot{e_4} + \lambda_4 e_4 \tag{42}$$

This is a valid sliding surface since it satisfies the following conditions:

- $\dot{s_4} = \ddot{\psi} - \ddot{\psi_d} + \lambda_4 \dot{e_4}$ contains the control input u_4
- As $s_4 \to 0$, the solution to the sliding surface is $e_4(t) = e^{-\lambda_4 t}$, which implies $e_4 \to 0$ and $\dot{e_4} \to 0$.

After choosing the sliding surface, the control law is formulated such that the sliding condition is satisfied.

$$s_4 \dot{s_4} = s_4 \Big[\ddot{\psi} - \ddot{\psi_d} + \lambda_4 \dot{e_4}\Big] s_4 \dot{s_4} = s_4 \Big[\Big[\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z}\Big] + [\frac{1}{I_z}]u_4 - \ddot{\psi_d} + \lambda_4 \dot{e_4}\Big] \tag{43}$$

Choose a u_4 such that the terms in the above equations are eliminated and also such that it contains an additional control variable u_r.

$$u_4 = I_z(-\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \ddot{\psi_d} - \lambda_4 \dot{e_4} + u_r) \tag{44}$$

Substituting the above u_4 in the equation for $s_4 \dot{s_4}$.

$$s_4 \dot{s_4} = s_4 u_r \tag{45}$$

The above equation should satisfy the sliding condition.

$$s_4 \dot{s_4} = s_4 u_r \leq -k_4 |s_4| \tag{46}$$

To satisfy the condition u_r becomes:

$$u_r = -k_4 sat(\frac{s_4}{\phi_{b4}}) \tag{47}$$

Here $k_4 > 0$. The overall control law is:

$$u_4 = I_z(-\dot{\phi}\dot{\theta}\frac{I_x - I_y}{I_z} + \ddot{\psi_d} - \lambda_4 \dot{e_4} - k_4 sat(\frac{s_4}{\phi_{b4}})) \tag{48}$$

1.3 Part 3: Code and Tuning

The simulation is done entirely on ROS and Gazebo with python. In the workspace setup, a ROS package is created with the title 'project'. In the package the main script for sliding control is written. In the script file a 'Quadrotor' class is created. In the constructor of the class the system parameters are created as attributes. The desired trajectory parameters and the propeller velocity terms are also initialized in the constructor. A 'traj_evaluate' function returns the desired trajectory positions, velocities and accelerations at the given time instance. This function generates the quintic polynomial trajectories for the quadrotor using the boundary conditions.
Next the 'smc_control' function calls the trajectory function to get the desired trajectory parameters at a given time instance. This function uses these values along with the current state variables that are published to get the new control inputs for the new time step. The formulae for the control inputs are discussed in the previous section. During this process the error variables for roll, pitch and yaw are ensured to be between $-\pi$ and π by using an inverse tangent function. The control inputs are then converted to propeller angular velocities using the allocation matrix and these values are kept within the desired range for the speed. The motor velocities are then published to each motor. This process is repeated for every time step.
The list of all the tuning parameters for the controller are given below:

1. **Kp = 95 and Kd =6**
   The proportional and derivative gains, Kp and Kd, impact the value of θ_d and ϕ_d, which indirectly represents the error in the x and y direction. When Kp and Kd are set at higher values, the resulting force is also greater, leading to a larger value of θ_d and ϕ_d and a faster convergence rate. Adjusting Kp and Kd to optimize performance requires careful consideration to prevent overshooting and oscillations.

2. **$\boldsymbol{\lambda_1 = 8}$, $\boldsymbol{\lambda_2 = 15}$, $\boldsymbol{\lambda_3 = 17}$, $\boldsymbol{\lambda_4 = 6}$**
   Increasing the value of lambda results in a stronger influence of the sliding surface on the system dynamics. This leads to faster convergence to the origin on the sliding surface, which is a desirable characteristic in sliding mode control. However, it is important to carefully choose the value of lambda, as an excessively high value can lead to undesirable chattering and oscillations in the system. Therefore, it is crucial to strike a balance between the convergence rate and the stability of the system while selecting the value of lambda.

3. **K1 = 12**
By increasing K1, the control input u1 also increases, resulting in a more aggressive controller. As a consequence, the quadrotor converges more rapidly to the desired trajectory.Aggressive control input may crash the quadrotor.

4. **K2 = 155**
By increasing K2, the control input u2 also increases, resulting in a more aggressive controller. As a consequence, the quadrotor converges more rapidly to the desired trajectory. Aggressive control input may crash the quadrotor.

5. **K3 = 115**
By increasing K3, the control input u3 also increases, resulting in a more aggressive controller. As a consequence, the quadrotor converges more rapidly to the desired trajectory.Aggressive control input may crash the quadrotor.

6. **K4 = 23**
By increasing K4, the control input u4 also increases, resulting in a more aggressive controller. As a consequence, the quadrotor converges more rapidly to the desired trajectory.Aggressive control input may crash the quadrotor.

7. $\boldsymbol{\phi_b = 0.9}$
We have found that the boundary layer is a crucial parameter for effective trajectory tracking. Our experiments showed that the quadrotor was unable to track desired trajectories without the implementation of a saturation function and boundary layer, which helps to eliminate discontinuity in the control law and reduces chattering. Increasing the boundary layer allows for smoother control input and improved performance. Additionally, we utilized the same boundary layer for designing all control laws in our study to ensure consistency and comparability of results.

2 Results

The performance of the quadrotor was evaluated based on the 3D plot of the actual trajectory over the desired trajectory, as shown in Figure 2. Upon careful observation, several findings were obtained. Firstly, in the x direction, the quadrotor exhibited slight wobbling near the waypoint. Although there was a slight deviation from the trajectory, it was negligible and is expected from the sliding mode control due to the boundary layer term used. Secondly, in the y direction, the motion was smooth with slight overshoot from the trajectory. Nevertheless, the tracking error was also negligible. Finally, in the z direction, the controller demonstrated good performance with no significant issues. These findings indicate that the controller is effective in ensuring satisfactory trajectory tracking in the x, y, and z directions, with only minor deviations or overshoots.

3 Discussion

In this project we developed a sliding mode control scheme to enable a quadrotor to track a desired trajectory in the presence of external disturbances. The controller was able to achieve convergence to the desired trajectory while keeping input torques to a in the required range, thereby preventing any unwanted damage to the motors. The controller included the boundary layer term to deal with the chattering in the control input. Similar to robust control this controller also exhibited a small error in the trajectory. But this error is negligible. The challenging aspect of the project was the tuning of the controller parameters. Especially in this controller there are many tuning variables and finding the right combination was difficult. Due to this, we can see a slight wobble in the quadrotor during some sections of the trajectory.
In the future, we can investigate other methods of enhancing the controller. It is worth exploring other methods that can improve the robustness of the controller without leading to chattering. For example, we could consider using adaptive control techniques, which allow the controller to adapt to changes in the system parameters over time. We can also explore the model predictive control technique for the project.

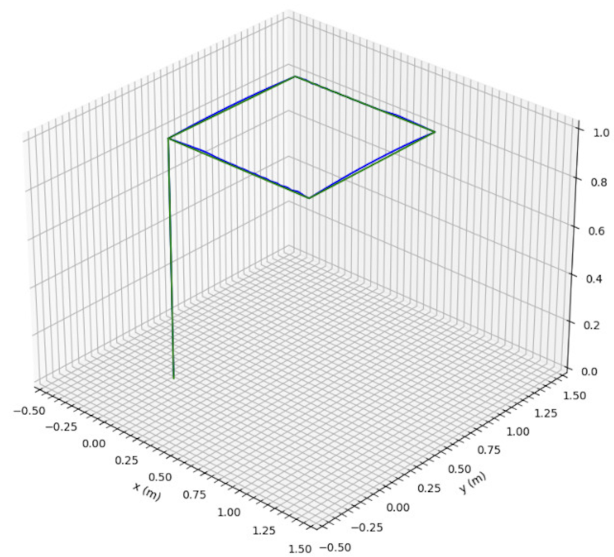


Figure 2: 3D plot of Desired and Actual Trajectory