Indian Institute of Technology Madras

Department of Aerospace Engineering

AS5545

Dynamics and Control of Spacecraft

Sliding-Mode Controller Design for Spacecraft Attitude Tracking Maneuvers

Submitted by:

AE21B056 Shreeya Padte

Contents

List of Figures

1 Abstract

This paper presents a robust sliding-mode control law for the spacecraft attitude tracking problem. By utilizing the two significant natural features connected to the spacecraft motion and the second approach of Lyapunov theory, it is easy to achieve system stability in the sliding mode. Multiaxial attitude tracking maneuvers serve as an illustration of the sliding-mode controller's effectiveness and resilience to uncertainty.

2 Introduction

The sliding mode controller is designed based on defining an error signal and a sliding vector to ensure convergence of tracking error to zero. A Lyapunov function is proposed to confirm the system's stability when in the sliding mode. The sliding vector incorporates a constant matrix Kp related to the convergence rate of the error signal. The control law is modified using a saturation function to mitigate chattering, which often arises in practical applications of sliding-mode control. The paper presents simulations of multiaxial attitude tracking maneuvers, showing the control laws' success and the sliding-mode controller's robustness to disturbances and parameter variations.

3 Spacecraft Model Description

A spacecraft's motion can be described by the kinematic and dynamic equations.

3.1 Kinematic equations

The kinematic equation describes the relation between the spacecraft's orientation (represented by a Gibbs vector ρ) and its angular velocity ω.

- **Gibbs vector**
- The Gibbs vector ρ (a 3-element vector) is one way to represent the orientation of a rigid body in three-dimensional space.
- Let $\rho \in R^3$ represent the orientation related to the rotation quaternion. The components of ρ are derived from the rotation quaternion q as follows:

$$\rho = (\frac{q_1}{q_4}, \frac{q_2}{q_4}, \frac{q_3}{q_4})^T \tag{1}$$

  where q $= [q_1, q_2, q_3, q_4]^T$ is the quaternion representation of the attitude.

- **Kinematic relationship:**

$$\dot{\rho} = T(\rho)\omega \tag{2}$$

  where $T(\rho)$ is a matrix that depends on ρ. This matrix is used to convert angular velocity to the rate of change of the Gibbs vector.

- The matrix $T(\rho)$ is a function of ρ and is designed to satisfy certain properties, including the condition

$$T(\rho) \geq \frac{1}{2}I$$

  , which ensures the stability and accuracy of the kinematic equation.

- $T(\rho)$ can be expressed as:

$$T(\rho) = \frac{1}{2}[(1 + \rho^T \rho)I + 2\rho\rho^T - \rho\times]$$

- To show that
$$T(\rho) \geq \frac{1}{2}I$$
, we have to show for any non-zero vector x $\in R^3$,
$$x^T(T(\rho) - \frac{1}{2}I)x \geq 0$$
$$x^T(T(\rho) - \frac{1}{2}I)x = \frac{1}{2}x^T(I + \rho\rho^T + \rho\times)x = \frac{1}{2}(x^Tx + x^Tpp^Tx + x^T(p\times)x)$$
Here, x^Tx is the squared norm, hence non-negative,
x^Tpp^Tx is $(x^Tp)^2$ which is a scalar representing the square of the projection of x onto ρ. This term is also always non-negative,
$x^T(p\times)x$ is 0. Skew-symmetric matrices have zero on the main diagonal and thus contribute zero when multiplied by the same vector from both sides. Thus,
$$x^TT(\rho)x =\geq \frac{1}{2}x^Tx$$
which implies,
$$x^T(T(\rho) - \frac{1}{2}I)x \geq 0$$
This means, $T(\rho) - \frac{1}{2}I$ is positive semi-definite. We conclude that,
$$T(\rho) \geq \frac{1}{2}I$$

Lyapunov function

Lyapunov functions are scalar functions that may be used to prove the stability of an equilibrium of an ODE. If the Lyapunov candidate function V is globally positive definite and its time derivative is globally negative definite, then the equilibrium is theoretically asymptotically stable. [**2**]
This property of $T(\rho) \geq \frac{1}{2}I$ is useful for stability analysis using Lyapunov theory, which is essential for proving the stability of the sliding-mode controller. Ensuring that $T(\rho)$ maintains a lower bound guarantees certain stability characteristics of the kinematic model.

3.2 Dynamic equations

The dynamic equation describes the rotational motion of the spacecraft, governed by Newton's second law for rotational systems. It considers the spacecraft's moment of inertia and external control torques.

- J is the spacecraft's inertia matrix (assumed to be symmetric and positive definite). The angular momentum h in the body frame is given by, $h = J\omega$
- **Euler's equation of motion:** The rotational dynamics are described by Euler's equation, which states that the time derivative of the angular momentum h is equal to the sum of applied torques and other forces: $\dot{h} = \tau$
- $h = J\omega$ and assuming J is constant,
$$J\dot{\omega} = \tau - \omega \times (J\omega)$$
Taking H $= -\omega \times J$, external disturbances d, the equation can be modified to,
$$J\dot{\omega} = H\omega + \tau + d \tag{3}$$

4 Sliding mode controller design

Sliding Mode Control (SMC) is a nonlinear control method widely used for its robustness against system uncertainties and external disturbances. The key idea of SMC is to design a sliding surface where the system's error dynamics exhibit desirable behavior and to develop a control law that forces the system to remain on this surface. In this report, we focus on designing an SMC for tracking the desired trajectory of a nonlinear dynamic system.

4.1 Sliding Mode Dynamics

The sliding vector $s(t)$ is defined to combine position and velocity errors into a single term:

$$s(t) = (\omega(t) - \omega_d(t)) + K_p(\rho(t) - \rho_d(t)),$$

where:

- $\omega(t)$: Actual angular velocity,
- $\omega_d(t)$: Desired angular velocity,
- $\rho(t) - \rho_d(t) = \epsilon(t)$: Position tracking error,
- K_p: Positive definite gain matrix.

The dynamics of $s(t)$ are derived by differentiating:

$$\dot{s}(t) = \dot{\omega}(t) - \dot{\omega}_d(t) + K_p\dot{\epsilon}(t).$$

Using the system's kinematic equations and premultiplying by $T(\rho)$, the sliding dynamics are expressed as:

$$\dot{\epsilon} + T(\rho)K_p\epsilon = 0.$$

4.2 Stability Analysis

To ensure stability, we propose a Lyapunov candidate function:

$$V_\epsilon = \frac{1}{2}\epsilon^T K_p \epsilon.$$

Differentiating V_ϵ, we obtain:

$$\dot{V}_\epsilon = \epsilon^T K_p \dot{\epsilon}.$$

Substituting $\dot{\epsilon} = -T(\rho)K_p\epsilon$, the derivative becomes:

$$\dot{V}_\epsilon = -\epsilon^T K_p T(\rho) K_p \epsilon.$$

Rewriting in terms of $K_p\epsilon$, we have:

$$\dot{V}_\epsilon = -(K_p\epsilon)^T T(\rho)(K_p\epsilon).$$

From Property 2 (positive definiteness of $T(\rho)$), we can further derive:

$$\dot{V}_\epsilon \leq -\frac{1}{2}(K_p\epsilon)^T(K_p\epsilon) \leq 0.$$

This shows that $\dot{V}_\epsilon$ is always non-positive, ensuring that V_ϵ is a Lyapunov function. Therefore, V_ϵ will converge to zero, and the error vector $\epsilon(t)$ will decay to zero over time.

4.3 Convergence Analysis

To analyze the convergence of the tracking errors, we consider the Lyapunov candidate function V_ϵ:

$$V_\epsilon = \frac{1}{2}\epsilon^T K_p \epsilon,$$

where K_p is symmetric and positive definite.

Derivative of V_ϵ

The time derivative of V_ϵ is given by:

$$\dot{V}_\epsilon = \epsilon^T K_p \dot{\epsilon}.$$

From the sliding mode condition, substituting $\dot{\epsilon} = -T(\rho)K_p\epsilon$, we have:

$$\dot{V}_\epsilon = -\epsilon^T K_p T(\rho) K_p \epsilon.$$

Since $T(\rho)$ is symmetric and positive definite, it follows that:

$$\dot{V}_\epsilon \leq -\frac{1}{2}(K_p\epsilon)^T(K_p\epsilon) \leq 0.$$

This confirms that $\dot{V}_\epsilon$ is non-positive, ensuring V_ϵ decreases monotonically.

Eigenvalue Representation

To analyze the decay rate, we decompose K_p as:

$$K_p = U\Sigma U^T,$$

where U is a unitary matrix, and $\Sigma = \text{diag}[\lambda_1, \lambda_2, \lambda_3]$, with $\lambda_i > 0$ being the eigenvalues of K_p. Substituting this decomposition, V_ϵ becomes:

$$V_\epsilon = \frac{1}{2}\Sigma(\epsilon^T U)^T(\epsilon^T U) = \frac{1}{2}(\lambda_1\epsilon_1^2 + \lambda_2\epsilon_2^2 + \lambda_3\epsilon_3^2),$$

where ϵ_i are components of the transformed error vector $\epsilon^T U$.

The derivative of V_ϵ is then:

$$\dot{V}_\epsilon \leq -\frac{1}{2}(\lambda_1^2\epsilon_1^2 + \lambda_2^2\epsilon_2^2 + \lambda_3^2\epsilon_3^2).$$

Exponential Convergence

Defining $\lambda_{\min} = \min(\lambda_1, \lambda_2, \lambda_3)$ and $\lambda_{\max} = \max(\lambda_1, \lambda_2, \lambda_3)$, the bounds for V_ϵ are:

$$\frac{1}{2}\lambda_{\min}\|\epsilon\|^2 \leq V_\epsilon \leq \frac{1}{2}\lambda_{\max}\|\epsilon\|^2.$$

The time evolution of V_ϵ satisfies:

$$\dot{V}_\epsilon \leq -\lambda_{\min}V_\epsilon.$$

Solving this differential inequality, we obtain:

$$V_\epsilon(t) \leq V_\epsilon(t_0)e^{-\lambda_{\min}(t-t_0)}.$$

Hence, the tracking errors satisfy:

$$\|\epsilon(t)\| \leq \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}\|\epsilon(t_0)\|e^{-\frac{1}{2}\lambda_{\min}(t-t_0)}.$$

The exponential convergence rate of the tracking errors is directly related to the smallest eigenvalue $\lambda_{\min}$ of K_p. By appropriately selecting K_p, the convergence rate can be controlled.

4.4 Control Law Design

The control law is designed such that the reaching and sliding conditions are satisfied:

$$V_s = \frac{1}{2} s^T J s \geq 0 \tag{4}$$

V_s is 0 only when s=0. We find first derivative:

$$\begin{aligned}
\dot{V}_s &= s^T J \dot{s} && (5)\\
&= s^T [J\dot{\omega} - J\dot{\hat{\omega}} + \lambda J(\dot{\rho} - \dot{\rho}_d)] && (6)\\
&= s^T [H\omega + \tau + d - J\dot{\hat{\omega}} + \lambda J T(\rho)\omega - \lambda J \dot{\rho}_d] && (7)
\end{aligned}$$

From above equations, we have:

$$\dot{V}_s = \boldsymbol{s}^T [\boldsymbol{\delta} + \boldsymbol{\tau}'] \tag{8}$$

Expanding this, it becomes:

$$\begin{aligned}
\dot{V}_s &= \sum_{i=1}^{3} s_i(\delta_i + \tau_i) && (9)\\
&= \sum_{i=1}^{3} -k_i |s_i| \left[1 - \frac{\delta_i}{k_i} \mathrm{sgn}(s_i)\right] && (10)
\end{aligned}$$

where

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \Delta H \boldsymbol{\omega} + \Delta \dot{J} \boldsymbol{\omega} + \lambda \Delta J [T(\boldsymbol{\rho})\boldsymbol{\omega} - \dot{\boldsymbol{\rho}}_d] + \boldsymbol{d}. \tag{11}$$

Since the external disturbances $\boldsymbol{d}$ and uncertain parameters ΔJ and ΔH are all bounded, the upper bound of $|\delta_i|$ can be found and is denoted as:

$$\delta_i^{\max} = \delta_i^{\max}(\boldsymbol{\omega}, \boldsymbol{\rho}, \dot{\boldsymbol{\rho}}_d, \ddot{\boldsymbol{\rho}}_d, t). \tag{12}$$

if we choose:

$$k_i = \delta_i^{\max}(\boldsymbol{\omega}, \boldsymbol{\rho}, \dot{\boldsymbol{\rho}}_d, \ddot{\boldsymbol{\rho}}_d, t), \quad \text{for } i = 1, 2, 3, \tag{13}$$

then equation becomes:

$$\begin{aligned}
\dot{V}_s &= -\sum_{i=1}^{3} \delta_i^{\max} |s_i| \left[1 - \frac{\delta_i}{\delta_i^{\max}} \mathrm{sgn}(s_i)\right] && (14)\\
&< 0, \quad \text{for } \boldsymbol{s} \neq 0. && (15)
\end{aligned}$$

This implies that V_s is really a Lyapunov function. Therefore, the reaching and sliding of the sliding mode $\boldsymbol{s} = 0$ is guaranteed.

Control law: $\boxed{\tau = -H_0\omega + J_0\dot{\hat{\omega}} - \lambda J_0[T(\rho)\omega - \dot{\rho}_d] + \tau',}$

where:

- $H_0 = h_0 \times \boldsymbol{\omega}, \quad h_0 = J_0\boldsymbol{\omega},$
- τ' is defined as:

$$\tau_i' = -k_i \cdot \mathrm{sgn}(s_i), \quad i = 1, 2, 3,$$

- $$\mathrm{sgn}(s_i) = \begin{cases} 1 & \text{if } s_i > 0, \\ -1 & \text{if } s_i < 0. \end{cases}$$

- $K = \begin{bmatrix} k_1 \cdot \text{sgn}(s_1) \\ k_2 \cdot \text{sgn}(s_2) \\ k_3 \cdot \text{sgn}(s_3) \end{bmatrix}$,

  The coefficients k_1, k_2, and k_3 are determined by the upper bounds of $\boldsymbol{\delta}$ and are computed as:

$$k_1 = |(\Delta J_{22} - \Delta J_{33})\omega_2\omega_3| + |\Delta J_{11}\dot{\hat{\omega}}_1| + \lambda \Delta J_{11}|T_{\omega_1} - \dot{\rho}_{d1}| + |d_1|_{\max} + 1, \tag{16}$$

$$k_2 = |(\Delta J_{33} - \Delta J_{11})\omega_3\omega_1| + |\Delta J_{22}\dot{\hat{\omega}}_2| + \lambda \Delta J_{22}|T_{\omega_2} - \dot{\rho}_{d2}| + |d_2|_{\max} + 1, \tag{17}$$

$$k_3 = |(\Delta J_{11} - \Delta J_{22})\omega_1\omega_2| + |\Delta J_{33}\dot{\hat{\omega}}_3| + \lambda \Delta J_{33}|T_{\omega_3} - \dot{\rho}_{d3}| + |d_3|_{\max} + 1, \tag{18}$$

  where T_{ω_i} represents the i-th element of $T(\boldsymbol{\rho})\omega$, i=1,2,3.

- $\boldsymbol{s}(t) = \boldsymbol{\omega}(t) - \hat{\boldsymbol{\omega}}(t) + \lambda(\boldsymbol{\rho}(t) - \boldsymbol{\rho}_d(t))$ is the sliding vector,

- $\text{sat}(s_i, \epsilon) = \begin{cases} 1, & \text{if } s_i > \epsilon, \\ \frac{s_i}{\epsilon}, & \text{if } |s_i| \leq \epsilon, \\ -1, & \text{if } s_i < -\epsilon, \end{cases}$

The control law ensures stability in the sliding mode, with the convergence rate of the error signal $\boldsymbol{\epsilon}(t)$ being $\lambda/2$. To mitigate chattering, the sign function is replaced with the saturation function $\text{sat}(s_i, \epsilon)$, where $\epsilon = 0.05$.

5 Results

Initial Conditions

The simulation uses the following initial conditions:

- Initial angular velocity: $\boldsymbol{\omega}(0) = [0.001, -0.005, 0.001]^T$ rad/s,
- Initial attitude vector: $\boldsymbol{\rho} = [1, 1, -1]^T$.

The desired reference trajectory for the attitude vector is given by:

$$\boldsymbol{\rho}_d(t) = \begin{bmatrix} \sin\left(\frac{\pi}{50}t\right) \\ -\sin\left(\frac{\pi}{50}t\right) \\ 0.5\cos\left(\frac{\pi}{50}t\right) \end{bmatrix}. \tag{19}$$

$$J_0 = \begin{bmatrix} 87.212 & 0 & 0 \\ 0 & 86.067 & 0 \\ 0 & 0 & 114.562 \end{bmatrix}. \tag{20}$$

The variation in the inertial matrix ΔJ satisfies:

$$|\Delta J_{011}| < 8.7212 \text{ (10\% of } J_{11}), \tag{21}$$

$$|\Delta J_{022}| < 4.3034 \text{ (5\% of } J_{22}), \tag{22}$$

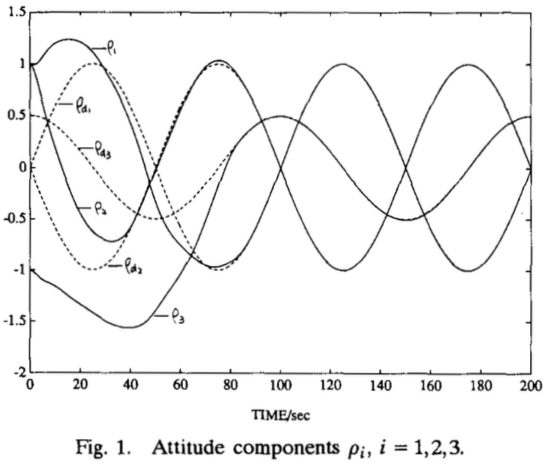
$$|\Delta J_{033}| < 17.1843 \text{ (15\% of } J_{33}). \tag{23}$$

The inertial matrix J is given as: J = J_0 + ΔJ

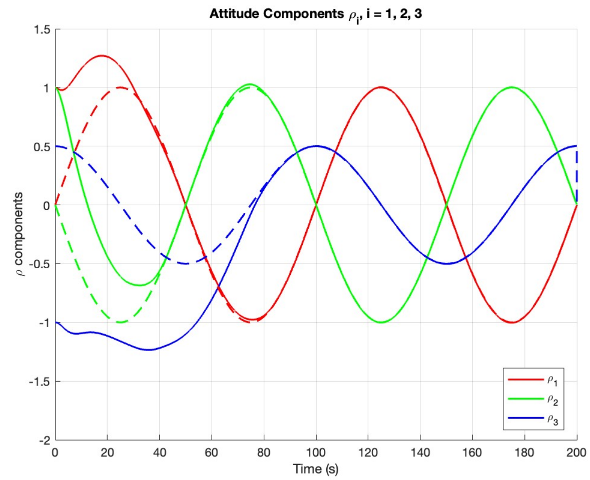
$$J = \begin{bmatrix} 95.933 & 0 & 0 \\ 0 & 81.763 & 0 \\ 0 & 0 & 131.746 \end{bmatrix}. \tag{24}$$

The external disturbance $\boldsymbol{d}$ is defined as:

$$\boldsymbol{d} = \begin{bmatrix} -0.005\sin(t) \\ +0.005\sin(t) \\ -0.005\sin(t) \end{bmatrix} \text{ (N·m)}. \tag{25}$$

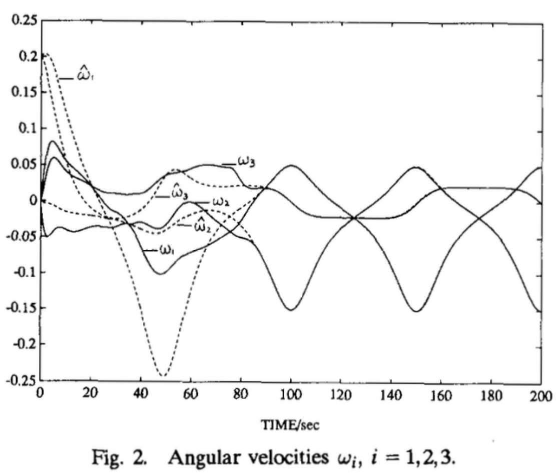


(a) Paper results

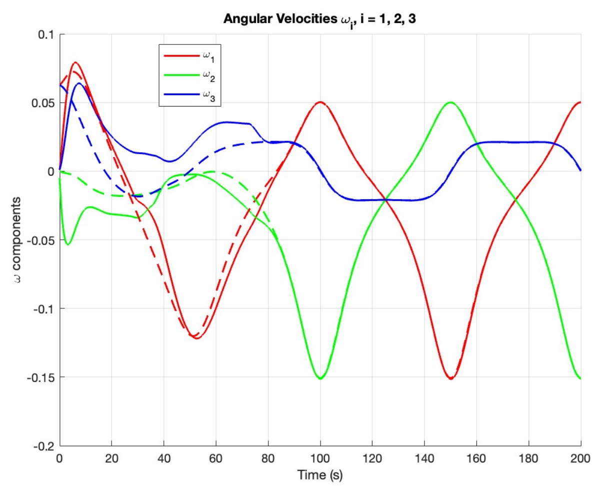


(b) Simulation results

Figure 1: Attitude components ρ_i, i = 1,2,3.



(a) Paper results



(b) Simulation results

Figure 2: Angular velocities ω_i, i = 1,2,3.

- The convergence of the attitude vector indicates that the sliding mode control law ensures stability and tracking accuracy.
- The angular velocity smoothly approaches the desired trajectory, showing the effectiveness of the control strategy in regulating the rotational motion.
- The torques remain smooth, with minimal evidence of chattering. This validates the design modification where a saturation function replaces the sign function.
- It takes approximately 80 seconds for the spacecraft to reach the desired attitude and angular velocity.
- The control torque requirement is low: around -4 to 2 Nm.

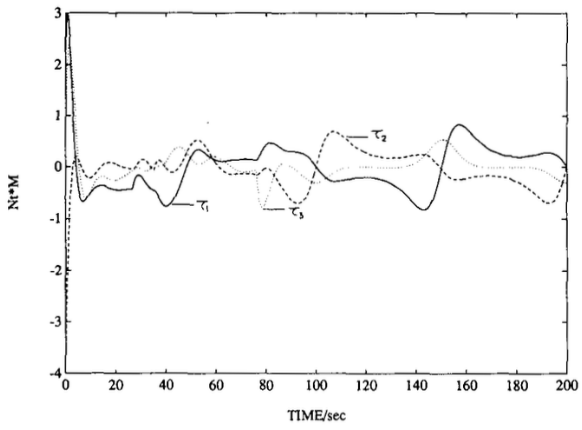
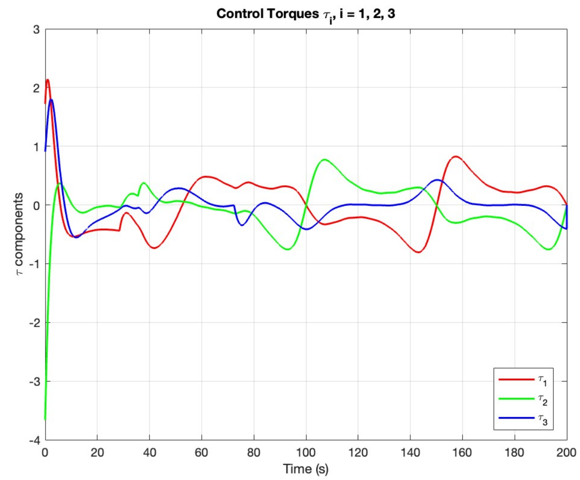


Fig. 3. Control torques τ_i, $i = 1,2,3$.

(a) Paper results



(b) Simulation results

Figure 3: Control torques τ_i, i = 1,2,3.

6 Conclusions

- We designed a sliding mode controller for spacecraft attitude tracking maneuvers. Sliding Mode Control provides a powerful framework for trajectory tracking with robust performance. By designing an appropriate sliding vector and employing Lyapunov stability theory, we ensure exponential convergence of the tracking error. This method is particularly suitable for systems with uncertainties and disturbances.
- There exist two important natural properties of the spacecraft model. First, the inertia matrix J is symmetric positive definite, and second, the matrix $T(\rho)$ in the kinematic equation satisfies $T(\rho) \geq \frac{1}{2}I$. With these properties and based on the direct method of the Lyapunov stability theory, a new sliding vector and two significant Lyapunov functions are introduced in the controller design and system stability analysis.
- The sliding surface S(x) is often chosen as a linear combination of the system states or errors. *S(x)* = *Ce*, C is the matrix that determines the dynamics of the sliding surface. $\dot{S}x = 0$, $\dot{e} = -C^{-1}S$, The eigenvalues of C control how fast the error decays to zero. A higher magnitude of eigenvalues implies faster convergence but introduces chattering in the system. Hence, the convergent rate of the error signal can be determined by choosing the sliding vector suitably.

References

[1] YON-PING CHEN, SHIH-CHE LO, Sliding-Mode Controller Design for Spacecraft Attitude Tracking Maneuvers, *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, VOL. 29, NO. 4, OCTOBER 1993.

[2] Wikipedia, *Lyapunov Function*, https://en.wikipedia.org/wiki/Lyapunovfunction.

7 Appendix

7.1 Code to get the plots:

```
clc;
clear;
close all;

% MULTI-ATTITUDE TRACKING MANEUVERS

% Define initial conditions and parameters based on the paper
omega0 = [0.001; -0.005; 0.001];   % Initial angular velocity
rho0 = [1; 1; -1];                  % Initial attitude components
rho_d = @(t) [sin(pi*t/50); -sin(pi*t/50); 0.5*cos(pi*t/50)]; %
    Reference vector

% Parameters for J0 and Delta J (variation in inertia)
J0 = diag([87.212, 86.067, 114.562]);  % Nominal inertia matrix
deltaJ_max = diag([8.7212, 4.3034, 17.1843]); % Maximum allowable
    variation
J = J0 + deltaJ_max;  % Perturbed inertia matrix

% Control parameters
lambda = 0.5; % Convergence rate of sliding mode control
epsilon = 0.05; % Saturation function parameter for SMC

% Simulation time
T = 200; % Total time in seconds (adjust as needed)
dt = 0.01; % Time step in seconds (adjust as needed)
time = 0:dt:T; % Time vector
N = length(time); % Number of time steps

d = zeros(3, N);
d(1, :) = -0.005 * sin(time);
d(2, :) = 0.005 * sin(time);
d(3, :) = -0.005 * sin(time);


% Initialize state variables for storing results
rho = zeros(3, length(time));
omega = zeros(3, length(time));
tau_control = zeros(3, length(time));

rho_d_values = zeros(3, length(time));
omega_cap = zeros(3, length(time));
T_rho_omega = zeros(3, length(time));

% Set initial conditions
rho(:,1) = rho0;
omega(:,1) = omega0;
rho_d_dot0 = [pi/50; -pi/50; 0];
omega_cap(:,1) = 0.5 * (eye(3) - skew_symmetric(rho0)) * rho_d_dot0
    ;

for i = 1:length(time)-1
    t = time(i);

    % Desired reference and its derivative
```

```
    rho_d_t = rho_d(t);
    rho_d_dot = (rho_d(t + dt) - rho_d(t)) / dt; % Approximate derivative
    rho_d_values(:, i) = rho_d(t);

    % Compute the transformation matrix T(rho)
    rho_t = rho(:,i);
    rho_cross = skew_symmetric(rho_t);
    T_rho = 0.5 * (eye(3) + (rho_t * rho_t') + rho_cross);
    rho_dot = T_rho * omega(:, i);
    T_rho_inv = 2 / (1 + rho_t' * rho_t) * (eye(3) - rho_cross);

    % Sliding vector s
    s = omega(:,i) - omega_cap(:,i) + lambda * (rho_t - rho_d_t);

    rho_d_2dot = (rho_d(t + 2*dt) - 2*rho_d(t + dt) + rho_d(t)) / (dt^2);

    rho_T_rho = rho_t' * rho_t;                         % Scalar: rho^T * rho
    T_rho_omega(:,i) = T_rho * omega(:,i);                   % Vector: T(rho) * omega

    term1 = -2 / (1 + rho_T_rho) * skew_symmetric(T_rho_omega(:,i));

    rho_T_rho_omega = rho_t' * T_rho_omega(:,i);

    term2 = -4 * rho_T_rho_omega / (1 + rho_T_rho)^2 * (eye(3) - skew_symmetric(rho_t));

    dT_inv_dt = term1 + term2;

    omega_cap_dot = dT_inv_dt * rho_d_dot + T_rho_inv * rho_d_2dot;

    k1 = abs((deltaJ_max(2,2) - deltaJ_max(3,3)) * omega(2,i) * omega(3,i)) + deltaJ_max(1,1) * abs(omega_cap_dot(1)) ...
         + lambda * deltaJ_max(1,1) * abs(T_rho_omega(1,i) - rho_d_dot(1)) + abs(0.005) + 1;

    k2 = abs((deltaJ_max(3,3) - deltaJ_max(1,1)) * omega(3,i) * omega(1,i)) + deltaJ_max(2,2) * abs(omega_cap_dot(2)) ...
         + lambda * deltaJ_max(2,2) * abs(T_rho_omega(2,i) - rho_d_dot(2)) + abs(0.005) + 1;

    k3 = abs((deltaJ_max(1,1) - deltaJ_max(2,2)) * omega(1,i) * omega(2,i)) + deltaJ_max(3,3) * abs(omega_cap_dot(3)) ...
         + lambda * deltaJ_max(3,3) * abs(T_rho_omega(3,i) - rho_d_dot(3)) + abs(0.005) + 1;

    h0 = J0 * omega(:, i);

    % control torque computation
    tau = - skew_symmetric(h0) * omega(:,i) + J0 * omega_cap_dot ...
          - J0 * lambda * (T_rho * omega(:,i) - rho_d_dot) ...
```

```
            - K_sign(s, epsilon, k1, k2, k3);
    tau_control(:,i) = tau;

    % current disturbance
    d_current = d(:, i);                     % 3x1 vector for the current time step

    H_omega_term = skew_symmetric(h0) * omega(:, i);  % H(omega)*omega
    tau_plus_d = tau + d_current;                 % Control + disturbance

    omega_dot = J \ (tau_plus_d + H_omega_term);

    omega(:, i+1) = omega(:, i) + omega_dot * dt;

    rho(:,i+1) = rho(:,i) + rho_dot * dt;

    omega_cap(:,i+1) = omega_cap(:,i) + omega_cap_dot * dt;
end


figure(1);
hold on;
plot(time, rho(1,:), 'r', time, rho(2,:), 'g', time, rho(3,:), 'b', 'LineWidth',1.5);
plot(time, rho_d_values(1,:), 'r--', time, rho_d_values(2,:), 'g--', time, rho_d_values(3,:), 'b--', 'LineWidth',1.5);
hold off;
grid on;
ylim([-2, 1.5]);
xlabel('Time (s)');
ylabel('\rho components');
title('Attitude Components \rho_i, i = 1, 2, 3');
legend('\rho_1', '\rho_2', '\rho_3','Location','best');

figure(2);
hold on;
plot(time, omega(1,:), 'r', time, omega(2,:), 'g', time, omega(3,:), 'b','LineWidth',1.5);
plot(time, omega_cap(1,:), 'r--', time, omega_cap(2,:), 'g--', time, omega_cap(3,:), 'b--','LineWidth',1.5);
hold off;
grid on;
xlabel('Time (s)');
ylabel('\omega components');
title('Angular Velocities \omega_i, i = 1, 2, 3');
legend('\omega_1', '\omega_2', '\omega_3','Location','best');

figure(3);
plot(time, tau_control(1,:), 'r', time, tau_control(2,:), 'g', time, tau_control(3,:), 'b','LineWidth',1.5);
grid on;
xlabel('Time (s)');
ylabel('\tau components');
title('Control Torques \tau_i, i = 1, 2, 3');
```

```
legend('\tau_1', '\tau_2', '\tau_3','Location','best');


function K_s = K_sign(s, epsilon, k1, k2, k3)
    K_s = [k1; k2; k3] .* sat(s, epsilon);
end

function sat_val = sat(s, epsilon)
    sat_val = min(1, abs(s / epsilon)) .* sign(s);
end

function S = skew_symmetric(v)
    S = [   0     -v(3)    v(2);
           v(3)     0     -v(1);
          -v(2)    v(1)     0   ];
end
```