AE426HON Project 2 – Dynamics and Stability of Spacecraft Attitude Systems

Jon Fueston

Embry-Riddle Aeronautical University

Professor Crespo Cutillas

20 February 2026

Introduction

The purpose of this project is to investigate the attitude dynamics and stability of rigid bodies through analytical derivation and numerical simulation. The project is divided into two sections: the analysis of free rigid body motion and the kinematics of a symmetric spinning top. Using MATLAB's numerical integrators, this report explores the stability of principal axis rotation and models the nutational evolution of a symmetric top using 3-1-3 Euler angle sequences.

Free Rigid Body Dynamics

Analytical Equilibrium Analysis

The motion of a torque-free rigid body is described by Euler's equations. For a body with principal moments of inertia $I_1 < I_2 < I_3$, these equations are given by:

$$I_1\dot{\omega}_1 = (I_2 - I_3)\omega_2\omega_3$$

$$I_2\dot{\omega}_2 = (I_3 - I_1)\omega_3\omega_1$$

$$I_3\dot{\omega}_3 = (I_1 - I_2)\omega_1\omega_2$$

To determine equilibrium solutions, the angular acceleration values are set to zero. As the principal moments of inertia are distinct, the inertia difference terms are non-zero and the products of the angular velocities must equal zero. To satisfy these condistion, the body must rotate purely about one of its principal axes. This results in three equilibrium solutions, which represent constant-rate pure spin about each of the axes.

Numerical Setup and Conservation Laws

To model these conditions from a general initial condition ω_0, MATLAB's ode45 numerical integrator was used. In torque-free motion, both kinetic energy and the magnitude of the angular momentum remain constant throughout the simulation. These quantities are defiined as:

$$T = \frac{1}{2}(I_1\omega_1^2 + I_2\omega_2^2 + I_3\omega_3^2)$$

$$H^2 = I_1^2\omega_1^2 + I_2^2\omega_2^2 + I_3^2\omega_3^2$$

As demonstrated in Figure 1, the simulation yields constant T and H^2, validating the accuracy of the implementation.

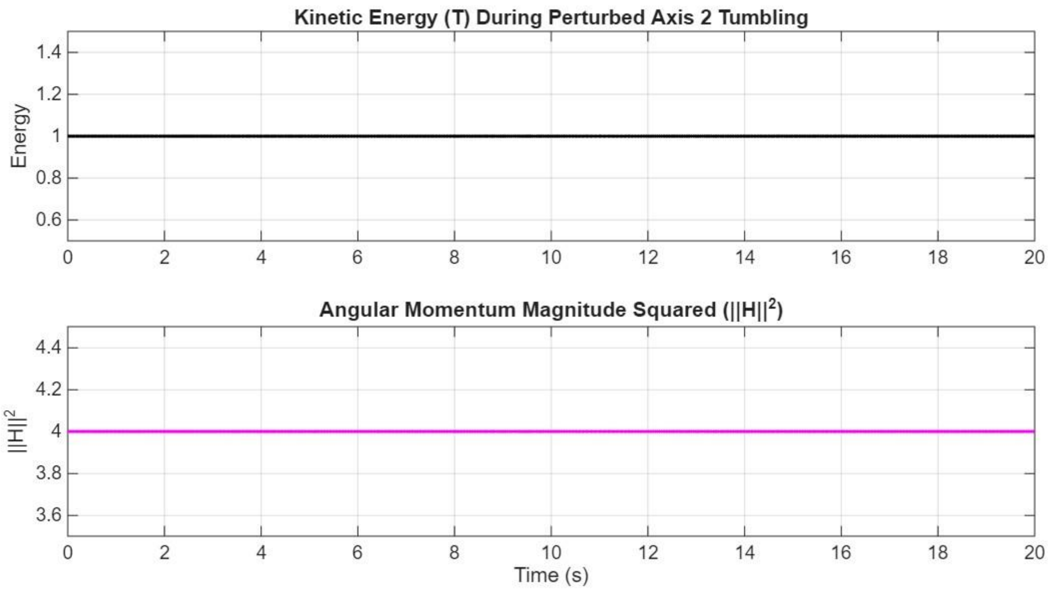

Figure 1. Constant kinetic energy and angular momentum despite tumbling body.

Stability Analysis

To evaluate the stability of the axes, each equilibrium solution was perturbed by a small value ϵ such that $\omega(0) = \omega_{eq} + \epsilon\delta\omega$ with $0 < \epsilon \ll 1$.

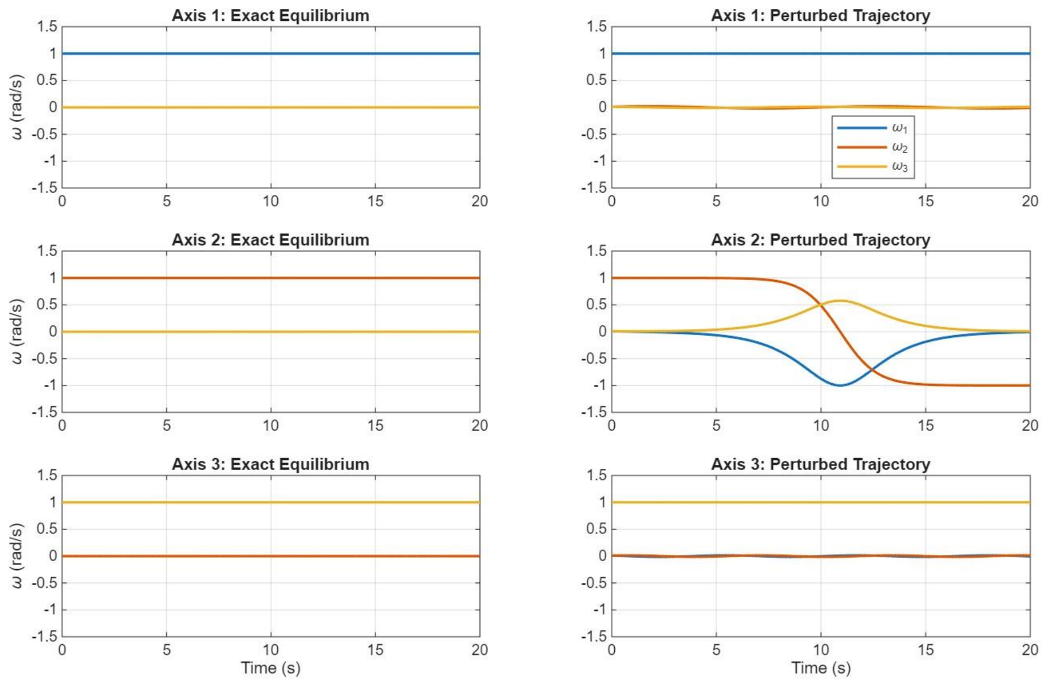

Figure 2. Perturbation of system on each axis.

Figure 2 shows the evolution of each perturbed condition. For rotations about the minor and major axes, the spin rate about the primary axes remains near the nominal equilibrium value, whereas the transverse axes harmonically oscillate, indicating stable equilibrium. Pertubation of the intermediate axis causes ω_2 to tumble, flipping between positive and negative spin. This is a demonstration of the Tennis Racket Theorem (Dzhanibekov effect), which states that rotation about the intermediate axis is inherently unstable in exactly the way shown in the simulation.

Axis-Symmetric Case

The simulation was additionally run for an axis-symmetric body with $I_1 = I_2 < I_3$. Because $I_1 - I_2 = 0$, the third Euler equation becomes $\dot{\omega_3} = 0$.

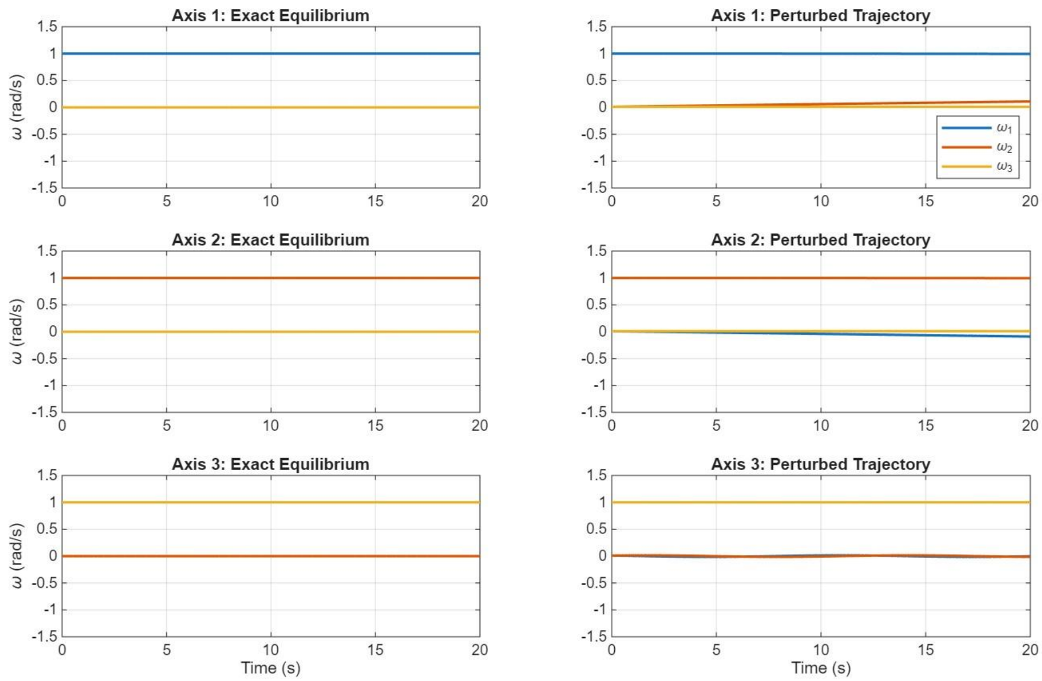

Figure 3. Perturbation of axis-symmetric body

As seen in Figure 3, this change results in a different stability profile. Major axis perturbations remain stable, whereas perturbations to axes 1 and 2 result in energy exchange between the symmetric axes without the tumbling instability shown in the asymmetric case.

Spinning Top in Euler Angles

Mathematical Derivation

The kinematics of a symmetric top can be described using a 3-1-3 Euler angle sequence representing precession ϕ , nutation θ , and spin ψ . To integrate the equations of motion, the angular velocity vector must be derived in terms of these angles and rates.

The sequence is described by three successive rotations from the inertial frame to the body-fixed frame.

1. Precession ($\dot{\phi}$) about the inertial Z-axis
2. Nutation ($\dot{\theta}$) about the X' axis
3. Spin ($\dot{\psi}$) about the body Z-axis.

Transformation into the final reference frame gives the final angular velocity vector:

$$\omega_1 = \dot{\phi} sin\theta sin\psi + \dot{\theta} cos\psi$$

$$\omega_2 = \dot{\phi} sin\theta cos\psi + \dot{\theta} sin\psi$$

$$\omega_3 = \dot{\phi} cos\theta + \dot{\psi}$$

Numerical Simulation

As the requested physical parameters could not be found, analysis was performed with selected values to demonstrate the influence of gyroscopic stiffness on nutation stability. The system was integrated with an initial spin rate of $\dot{\psi}(0) = 20\pi$ [rad/s] and initial nutation angles $\theta(0) = 0.001$ [rad] and $\theta(0) = \frac{\pi}{2}$ [rad].

In the near-vertical case, the high spin rate generated enough angular momentum to constrain the top to small, harmonic change in nutation. As shown in Figure 4, the nutation angle remains mostly constant, and the top does not reach the 95-degree limit in the 10-second simulation window.

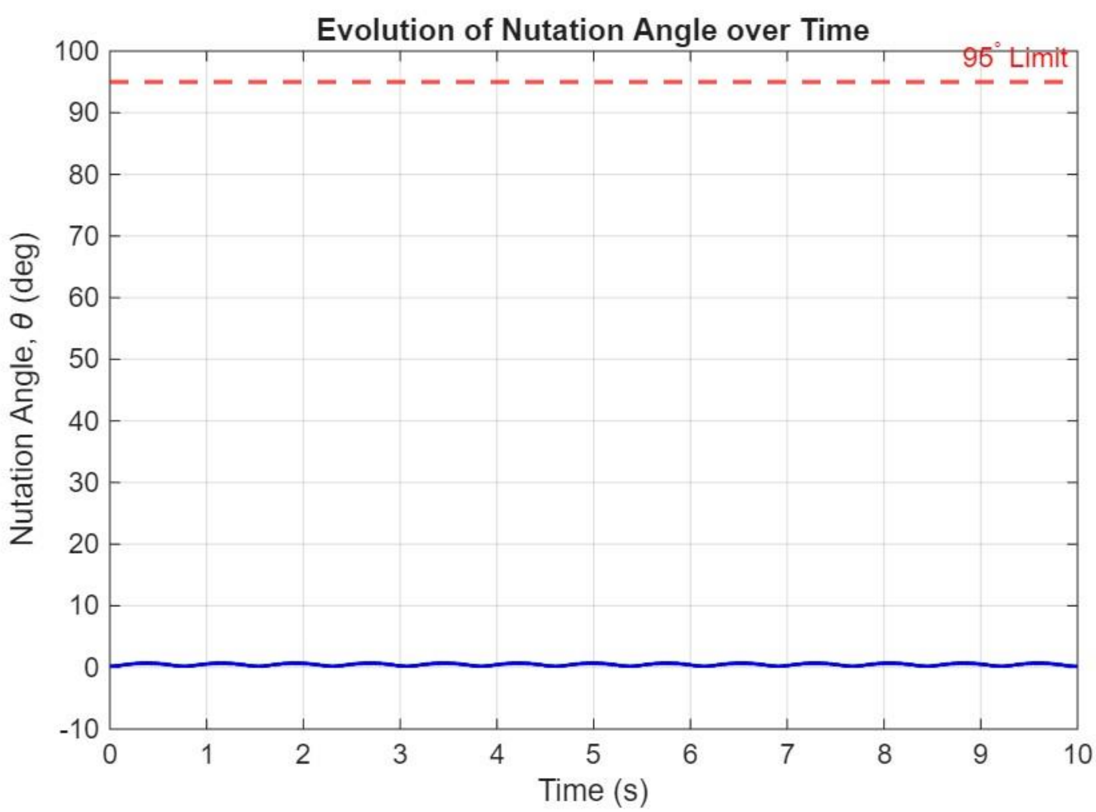

Figure 4. Small oscillation of nutation angle.

When the spin rate is decreased to 3.14 [rad/s], the gyroscopic effect is not strong enough to overcome even the small 0.001 [rad] perturbation and the system becomes unstable, as shown in Figure 5 below.

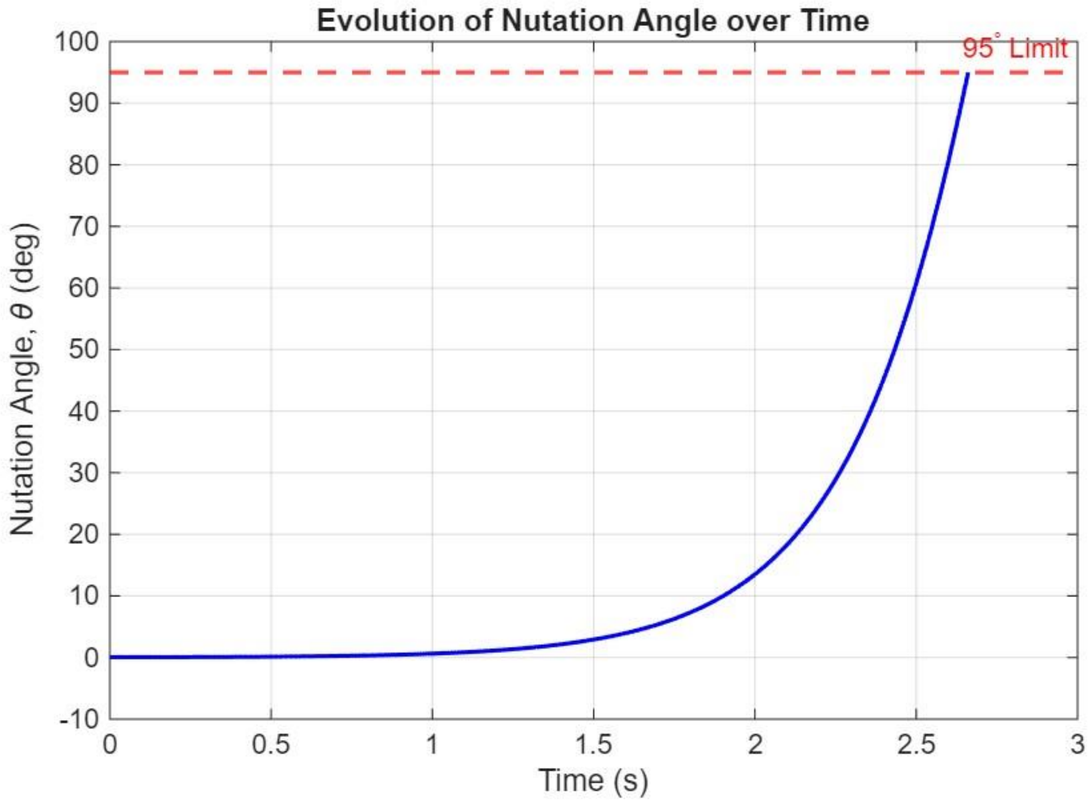

Figure 5. Low spin rate simulation.

To further understand the instability of the system, a second simulation was run with an initial nutation angle of 45 degrees. By increasing this angle, the gravitational torque $M = mgl \sin\theta$ increases, overcoming the gyroscopic stabilization provided by the high spin rate. The top decays rapidly, reaching the 95-degree mark within the simulation timeframe. Figure 6 demonstrates that while high spin rates can provide stability, there is a threshold where the gravity torque "wins" and the system destabilizes.

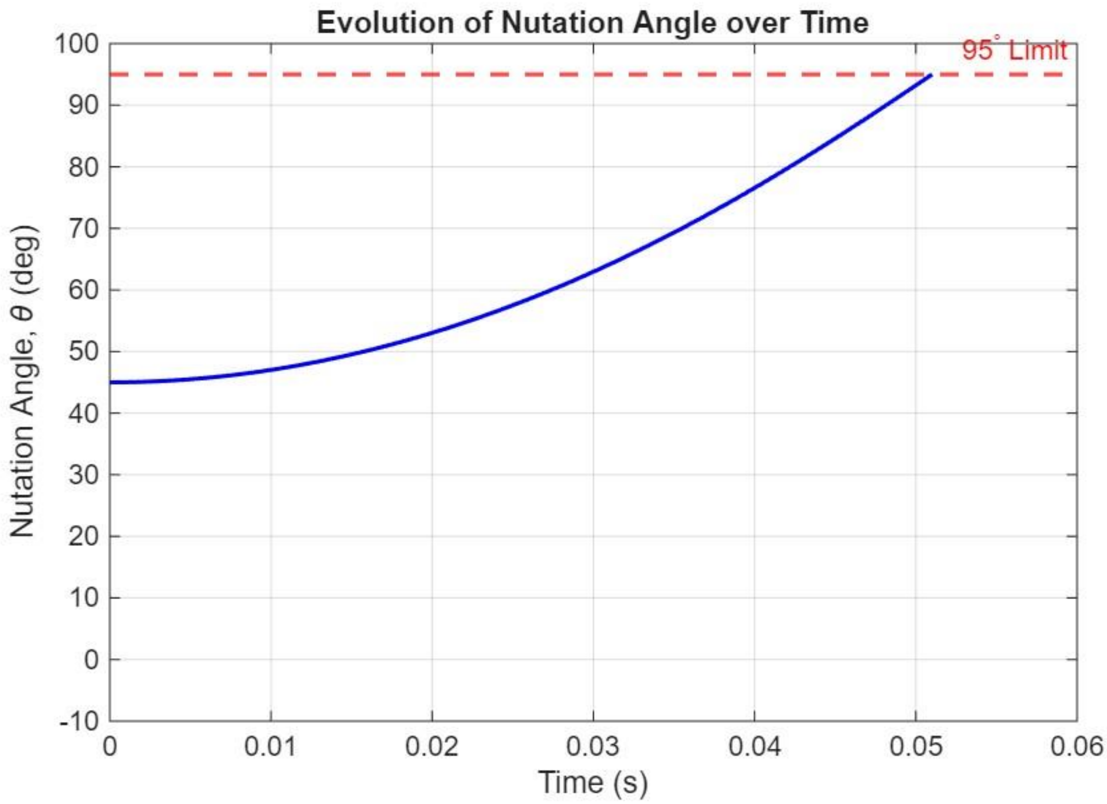

Figure 6. 45-degree initial nutation angle.

Conclusion

This project successfully simulated the attitude dynamics of both free rigid bodies and symmetric tops using numerical and analytical methods.

The investigation in Part 1 confirmed that for asymmetric rigid bodies, stable rotation is only possible about the major and minor axes. The numerical simulation provided results in agreement with the Dzhanibekov effect, where small perturbations lead to dramatic tumbling about the intermediate axis. Additional simulation of the axis-symmetric case provided evidence that symmetry eliminates this instability.

In Part 2, the derivation of the Euler angle kinematics allowed for the modeling of a symmetric top. The results show the competition between gyroscopic stability and gravitational torque. Even without specific textbook parameters, the simulation shows that high spin rates can keep the nutation angle at low values, but that high initial tilt will lead to the collapse of the system. All computation evidence supports the underlying principles of rigid body dynamics.

Appendix A – MATLAB Code

```
% JB Fueston - 2627845 | AE426HON Final Project | 29 April 2026
```

```
% References: MATLAB documentation, Stack Overflow, Curtis, AI
% All logic is handwritten. Any AI generated code is flagged as such

% NOTE: Parameters It, Ia, and mgl are placeholders to demonstrate the
% behavior of the top. References values could not be found in my copy of Curtis'
book

clear; clc; close all;

%%  PART I

% Moments of inertia
I = [2.0; 2.0; 3.0];

% spin rate and small perturbation (0 < epsilon << 1)
Omega = 1;        % rad/s
epsilon = 0.01;   % Perturbation value

% Unperturbed equilibrium
w_eq1 = [Omega; 0; 0];
w_eq2 = [0; Omega; 0];
w_eq3 = [0; 0; Omega];

% Perturbed IC
w0_p1 = w_eq1 + [0; epsilon; epsilon];
w0_p2 = w_eq2 + [epsilon; 0; epsilon];
w0_p3 = w_eq3 + [epsilon; epsilon; 0];

tspan_part1 = [0 20]; % 20s
options_part1 = odeset('RelTol', 1e-8, 'AbsTol', 1e-8);

% Simulate and Plot
% Simulation code hand-written. 6-pack sublot logic AI generated.
figure('Name', 'Stability of Principal Axes', 'Position', [100, 100, 1000, 600]);

% Minor
[t_unpert1, w_unpert1] = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w_eq1,
options_part1);
[t_pert1, w_pert1]  = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w0_p1,
options_part1);

subplot(3, 2, 1);
plot(t_unpert1, w_unpert1, 'LineWidth', 1.5);
title('Axis 1: Exact Equilibrium'); ylabel('\omega (rad/s)'); grid on;
ylim([-1.5, 1.5]); % Adds space around the lines

subplot(3, 2, 2);
plot(t_pert1, w_pert1, 'LineWidth', 1.5);
title('Axis 1: Perturbed Trajectory'); grid on;
legend('\omega_1', '\omega_2', '\omega_3', 'Location', 'best');
ylim([-1.5, 1.5]);

% Intermediate
[t_unpert2, w_unpert2] = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w_eq2,
options_part1);
```

```
[t_pert2, w_pert2]  = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w0_p2,
options_part1);

subplot(3, 2, 3);
plot(t_unpert2, w_unpert2, 'LineWidth', 1.5);
title('Axis 2: Exact Equilibrium'); ylabel('\omega (rad/s)'); grid on;
ylim([-1.5, 1.5]);

subplot(3, 2, 4);
plot(t_pert2, w_pert2, 'LineWidth', 1.5);
title('Axis 2: Perturbed Trajectory'); grid on;
ylim([-1.5, 1.5]);

% Major
[t_unpert3, w_unpert3] = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w_eq3,
options_part1);
[t_pert3, w_pert3]  = ode45(@(t, w) euler_equations(t, w, I), tspan_part1, w0_p3,
options_part1);

subplot(3, 2, 5);
plot(t_unpert3, w_unpert3, 'LineWidth', 1.5);
title('Axis 3: Exact Equilibrium'); ylabel('\omega (rad/s)'); xlabel('Time (s)');
grid on;
ylim([-1.5, 1.5]);

subplot(3, 2, 6);
plot(t_pert3, w_pert3, 'LineWidth', 1.5);
title('Axis 3: Perturbed Trajectory'); xlabel('Time (s)'); grid on;
ylim([-1.5, 1.5]);

% Conservation
T = 0.5 * (I(1).*w_pert2(:,1).^2 + I(2).*w_pert2(:,2).^2 + I(3).*w_pert2(:,3).^2);
H2 = (I(1)^2).*w_pert2(:,1).^2 + (I(2)^2).*w_pert2(:,2).^2 +
(I(3)^2).*w_pert2(:,3).^2;

figure('Name', 'Conservation of Energy and Momentum', 'Position', [150, 150, 800,
400]);

subplot(2, 1, 1);
plot(t_pert2, T, 'k', 'LineWidth', 1.5);
title('Kinetic Energy (T) During Perturbed Axis 2 Tumbling');
ylabel('Energy'); grid on;
ylim([T(1) - 0.5, T(1) + 0.5]);

subplot(2, 1, 2);
plot(t_pert2, H2, 'm', 'LineWidth', 1.5);
title('Angular Momentum Magnitude Squared (||H||^2)');
ylabel('||H||^2'); xlabel('Time (s)'); grid on;
ylim([H2(1) - 0.5, H2(1) + 0.5]);


%%  PART II: SPINNING TOP IN EULER ANGLES

% Parameters (Dummy variables - I cannot find the top referenced in the problem
statement)
```

```
It  = 0.1;    % I1 = I2
Ia  = 0.05;   % I3
mgl = 100;    % Gravity torque parameter

% ICs
phi0 = 0;           % precession (rad)
psi0 = 0;           % spin (rad)
theta0 = 0.25*pi;      % nutation (rad) - Given

phi_dot0 = 0;       % Initial precession rate (rad/s)
theta_dot0 = 0;     % Initial nutation rate (rad/s)
psi_dot0 = 20*pi;   % Initial spin rate (rad/s) - Given in prompt

X0 = [phi0; theta0; psi0; phi_dot0; theta_dot0; psi_dot0];

tspan_part2 = [0 10]; %10s

% Stop @ 95 deg
options_part2 = odeset('RelTol', 1e-8, 'AbsTol', 1e-8, 'Events', @nutation_event);
[t2, X] = ode45(@(t,X) top_dynamics(t, X, It, Ia, mgl), tspan_part2, X0,
options_part2);

theta_deg = rad2deg(X(:, 2));

% Plot nutation
figure('Name', 'Top Nutation', 'Position', [150, 150, 600, 400]);
plot(t2, theta_deg, 'b', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Nutation Angle, \theta (deg)');
title('Evolution of Nutation Angle over Time');
yline(95, 'r--', '95^\circ Limit', 'LineWidth', 1.5);
ylim([-10, 100])
grid on;


%% Local functions
% FRB Euler eqns
function dwdt = euler_equations(~, w, I)
    I1 = I(1); I2 = I(2); I3 = I(3);
    w1 = w(1); w2 = w(2); w3 = w(3);

    dwdt = zeros(3,1);
    dwdt(1) = ((I2 - I3) / I1) * w2 * w3;
    dwdt(2) = ((I3 - I1) / I2) * w3 * w1;
    dwdt(3) = ((I1 - I2) / I3) * w1 * w2;
end

% Top eqns of motion
function dXdt = top_dynamics(~, X, It, Ia, mgl)
    theta = X(2);
    phi_d = X(4);
    th_d  = X(5);
    psi_d = X(6);

    % accelerations
```

```
    th_dd = (It * phi_d^2 * sin(theta) * cos(theta) - Ia * phi_d * sin(theta) *
(phi_d * cos(theta) + psi_d) + mgl * sin(theta)) / It;

    if abs(sin(theta)) < 1e-6
        phi_dd = 0;
    else
        phi_dd = (-2 * It * phi_d * th_d * cos(theta) + Ia * th_d * (phi_d *
cos(theta) + psi_d)) / (It * sin(theta));
    end

    psi_dd = phi_d * th_d * sin(theta) - phi_dd * cos(theta);

    dXdt = [phi_d; th_d; psi_d; phi_dd; th_dd; psi_dd];
end

% Stopping integration at 95 deg - This method is AI-generated
function [value, isterminal, direction] = nutation_event(~, X)
    % Triggers when Nutation (theta) reaches 95 degrees (in radians)
    target_theta = deg2rad(95);
    theta = X(2);

    value = theta - target_theta;
    isterminal = 1;               % 1 = Stop the integration
    direction = 0;
end
```