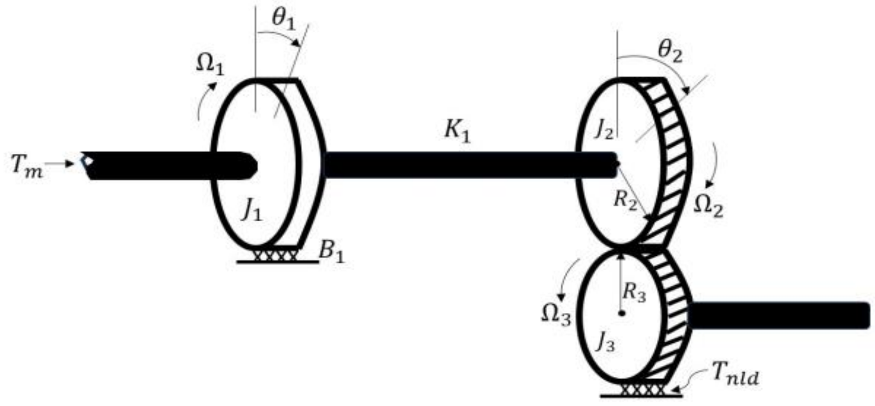
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

AE233M: INTRODUCTION TO VIBRATIONS

PROJECT TITLE:

VIBRATION ANALYSIS OF A TILTROTOR GEARBOX



SUBMITTED BY:

Ambadas Joshi (220485)

C Disha (220302)

Pallavi Anil Undre (220748)

PROFESSOR'S NAME:

Tanmay Mathur

Date of Submission: 15 Feb 2025

Computer Project: AE 233M Introduction to Vibrations

Question 1: Nonlinear Equations of Motion

Problem Statement

The system consists of a tiltrotor gearbox with the following parameters:

- Rotational inertias: $J_1 = 2\,\text{N-m-sec}^2/\text{rad}$, $J_2 = 1\,\text{N-m-sec}^2/\text{rad}$, $J_3 = 0.5\,\text{N-m-sec}^2/\text{rad}$
- Torsional stiffness: $K_1 = 1000\,\text{Nm/rad}$
- Damping coefficients: $B_1 = 10\,\text{N-m-sec/rad}$, $\beta = 5\,\text{N-m-sec}^3/\text{rad}^3$
- Gear radii: $R_2 = 0.5\,\text{m}$, $R_3 = 0.25\,\text{m}$

The nonlinear friction between Gear 3 and the housing is:

$$T_{\text{nld}} = 1.5\Omega_3 + \beta\Omega_3^3 \tag{1}$$

Degrees of Freedom

The system has three degrees of freedom:

- $q_1 = \Omega_1$ (Angular velocity of J_1)
- $q_2 = \theta_1 - \theta_2$ (Relative angular displacement)
- $q_3 = \Omega_3$ (Angular velocity of J_3)

Equations of Motion

Using Newton's mechanics, the equations of motion are:

$$J_1\dot{\Omega}_1 = T_m - B_1\Omega_1 - K_1(\theta_1 - \theta_2) \tag{2}$$
$$J_2\dot{\Omega}_2 = K_1(\theta_1 - \theta_2) - T_{\text{gear}} \tag{3}$$
$$J_3\dot{\Omega}_3 = T_{\text{gear}} - T_{\text{nld}} \tag{4}$$

where T_{gear} is the torque transmitted between Gear 1 and Gear 2.
Using the gear relationship:

$$\Omega_2 = \frac{R_3}{R_2}\Omega_3 = 0.5\Omega_3 \tag{5}$$

State-Space Representation

Defining the state variables:

$$q_1 = \Omega_1, \quad q_2 = \theta_1 - \theta_2, \quad q_3 = \Omega_3 \tag{6}$$

The state-space equations become:

$$\dot{q}_1 = \frac{1}{J_1}\left(T_m - B_1 q_1 - K_1 q_2\right) \tag{7}$$
$$\dot{q}_2 = q_1 - 0.5q_3 \tag{8}$$
$$\dot{q}_3 = \frac{1}{J_3 + 0.5J_2}\left(K_1 q_2 - 1.5q_3 - \beta q_3^3\right) \tag{9}$$

Question 2: System response to Step Torque

MATLAB CODE

The following MATLAB code is to simulate the response of the system (non linear) due to step torque (Tm) which changes from magnitude (case1) 1 N-m to 2 N-m at t = 0 and (case2) 1 N-m to 30 N-m at t = 0.

```
    function tiltrotor_gearbox2()
    % Parameters
    J1 = 2; J2 = 1; J3 = 0.5;
    K1 = 1000; B1 = 10; beta = 5;

    % Initial conditions
    x0 = [0; 0; 0]; % [Omega1; theta1-theta2; Omega3]

    % Time span
    tspan = [0 10];

    % Step input 1: Tm changes from 1 N-m to 2 N-m
    Tm1 = @(t) 1 + (t >= 0);
    [t1, x1] = ode45(@(t, x) dynamics(t, x, Tm1(t), J1, J3, J2, K1, B1, beta), tspan, x0);

    % Step input 2: Tm changes from 1 N-m to 30 N-m
    Tm2 = @(t) 1 + 29 * (t >= 0);
    [t2, x2] = ode45(@(t, x) dynamics(t, x, Tm2(t), J1, J3, J2, K1, B1, beta), tspan, x0);

    % Plot results
    figure;
    subplot(2,1,1);
    plot(t1, x1(:,2)); % theta1 - theta2 for Tm = 2 N-m
    title('StepResponse:␣Tm␣=␣2␣N-m');
    xlabel('Time(s)'); ylabel('\theta_1-\theta_2(rad)');

    subplot(2,1,2);
    plot(t2, x2(:,2)); % theta1 - theta2 for Tm = 30 N-m
    title('StepResponse:␣Tm␣=␣30␣N-m');
    xlabel('Time(s)'); ylabel('\theta_1-\theta_2(rad)');

    % Analyze results
    analyze_response(t1, x1(:,2), 'Tm␣=␣2␣N-m');
    analyze_response(t2, x2(:,2), 'Tm␣=␣30␣N-m');
end

function dxdt = dynamics(~, x, Tm, J1, J3, J2, K1, B1, beta)
    % State variables
    Omega1 = x(1);
    theta_diff = x(2);
    Omega3 = x(3);

    % Equations of motion
    dOmega1 = (Tm - B1 * Omega1 - K1 * theta_diff) / J1;
    dtheta_diff = Omega1 - 0.5 * Omega3;
    dOmega3 = (K1 * theta_diff - 1.5 * Omega3 - beta * Omega3^3) / (J3 + 0.5 * J2);

    % Return derivatives
    dxdt = [dOmega1; dtheta_diff; dOmega3];
end

function analyze_response(t, theta_diff, title_str)
    % Steady-state value
    steady_state = theta_diff(end);
    fprintf('%s␣-␣Steady-state␣value:␣%.4f␣rad\n', title_str, steady_state);

    % Maximum overshoot
    max_overshoot = max(theta_diff) - steady_state;
    fprintf('%s␣-␣Maximum␣overshoot:␣%.4f␣rad\n', title_str, max_overshoot);

    % Period of oscillations
    [~, locs] = findpeaks(theta_diff);
    if length(locs) >= 2
        period = mean(diff(t(locs)));
        fprintf('%s␣-␣Period␣of␣oscillations:␣%.4f␣s\n', title_str, period);
    else
        fprintf('%s␣-␣Not␣enough␣peaks␣to␣determine␣period.\n', title_str);
    end
end
```

Simulation Output

Torque T_m (N-m)	Steady-State Value (rad)	Max Overshoot (rad)	Period (s)
2	0.0006	0.0012	0.1975
30	0.0221	0.0042	0.1996

Table 1: Simulation results for different torque values

Plots for the Response of the System for Change in Torque

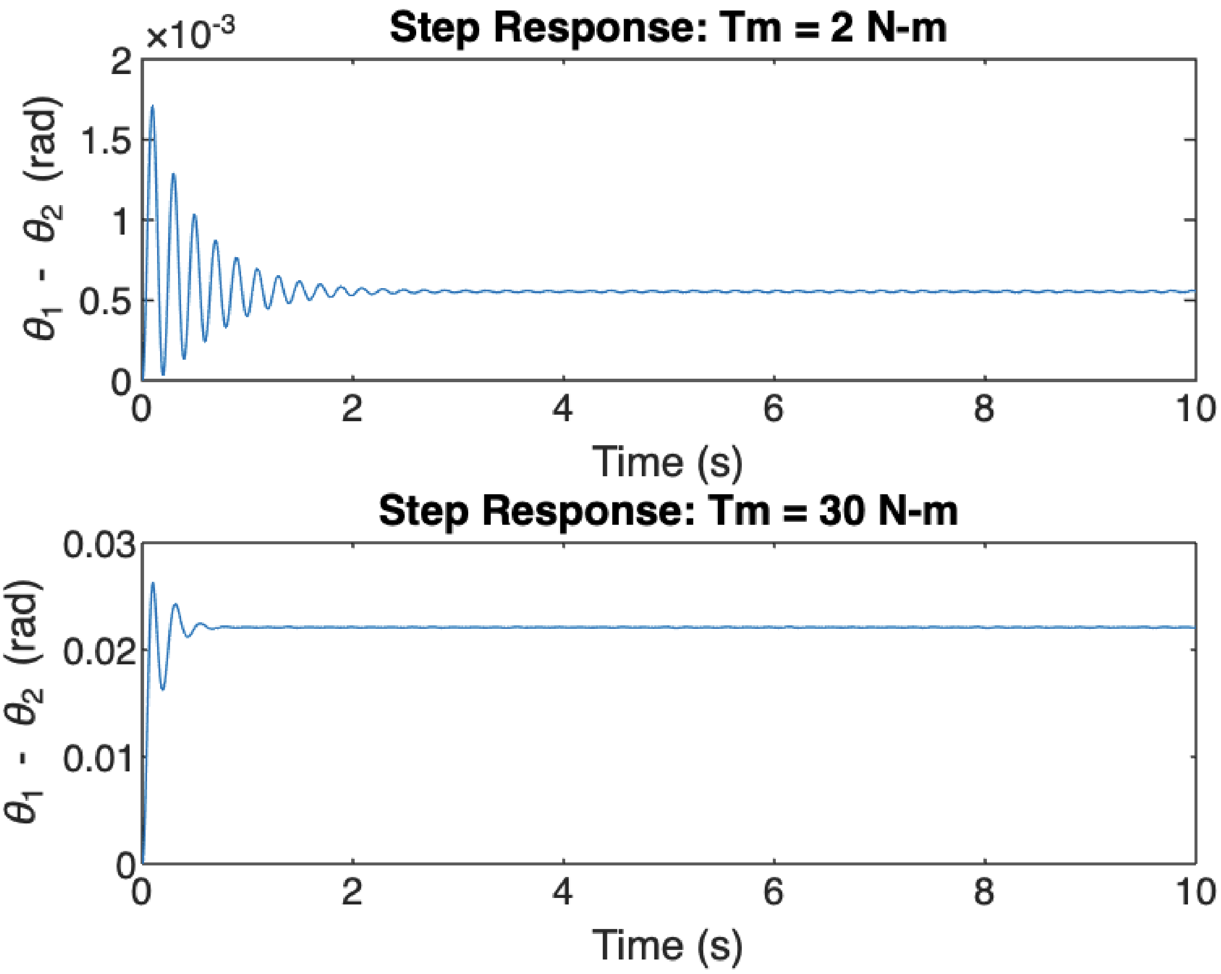


Figure 1: Plot of the response of the nonlinear system

Question 3: Linearize the system model in the vicinity of the equilibrium operating point, Tm = 1 Nm.

State-Space Representation

The state variables are defined as:

$$q_1 = \Omega_1, \quad q_2 = \theta_1 - \theta_2, \quad q_3 = \Omega_3$$

The nonlinear state-space equations of motion are:

$$\dot{q}_1 = \frac{1}{J_1}\left(T_m - B_1 q_1 - K_1 q_2\right)$$

$$\dot{q}_2 = q_1 - 0.5 q_3$$

$$\dot{q}_3 = \frac{1}{J_3 + 0.5 J_2}\left(K_1 q_2 - 1.5 q_3 - \beta q_3^3\right)$$

Equilibrium point

At equilibrium, all derivatives are zero ($\dot{q}_1 = \dot{q}_2 = \dot{q}_3 = 0$), and $T_m = 1\,\text{N-m}$.

1. From $\dot{q}_1 = 0$:

$$0 = \frac{1}{J_1}\left(T_m - B_1 q_1 - K_1 q_2\right)$$

$$T_m = B_1 q_1 + K_1 q_2$$

At equilibrium, $T_m = 1$, so:

$$1 = B_1 q_1 + K_1 q_2$$

2. From $\dot{q}_2 = 0$:

$$0 = q_1 - 0.5 q_3$$

$$q_1 = 0.5 q_3$$

3. From $\dot{q}_3 = 0$:

$$0 = \frac{1}{J_3 + 0.5 J_2}\left(K_1 q_2 - 1.5 q_3 - \beta q_3^3\right)$$

$$K_1 q_2 = 1.5 q_3 + \beta q_3^3$$

Substitute $q_1 = 0.5 q_3$ into the equation for q_2:

$$1 = B_1(0.5 q_3) + K_1 q_2$$

$$1 = 0.5 B_1 q_3 + K_1 q_2$$

From $K_1 q_2 = 1.5 q_3 + \beta q_3^3$, substitute q_2:

$$1 = 0.5 B_1 q_3 + (1.5 q_3 + \beta q_3^3)$$

$$1 = (0.5 B_1 + 1.5) q_3 + \beta q_3^3$$

Substitute $B_1 = 10$:

$$1 = (0.5 \cdot 10 + 1.5) q_3 + \beta q_3^3$$

$$1 = 6.5 q_3 + \beta q_3^3$$

For small q_3, the cubic term βq_3^3 is negligible, so:

$$q_3 \approx \frac{1}{6.5} \approx 0.1538\,\text{rad/s}$$

Then:

$$q_1 = 0.5 q_3 \approx 0.0769\,\text{rad/s}$$

$$q_2 = \frac{1 - B_1 q_1}{K_1} \approx \frac{1 - 10 \cdot 0.0769}{1000} \approx 0.000231\,\text{rad}$$

Thus, the equilibrium point is:

$$q_1 \approx 0.0769, \quad q_2 \approx 0.000231, \quad q_3 \approx 0.1538$$

Linearization of the System

Compute the Jacobian Matrix

The system dynamics are:

$$f_1 = \frac{1}{J_1}\left(T_m - B_1 q_1 - K_1 q_2\right)$$
$$f_2 = q_1 - 0.5 q_3$$
$$f_3 = \frac{1}{J_3 + 0.5 J_2}\left(K_1 q_2 - 1.5 q_3 - \beta q_3^3\right)$$

The Jacobian matrix A is:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\ \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \end{bmatrix}$$

Compute each partial derivative:
1. For f_1:

$$\frac{\partial f_1}{\partial q_1} = -\frac{B_1}{J_1}, \quad \frac{\partial f_1}{\partial q_2} = -\frac{K_1}{J_1}, \quad \frac{\partial f_1}{\partial q_3} = 0$$

2. For f_2:

$$\frac{\partial f_2}{\partial q_1} = 1, \quad \frac{\partial f_2}{\partial q_2} = 0, \quad \frac{\partial f_2}{\partial q_3} = -0.5$$

3. For f_3:

$$\frac{\partial f_3}{\partial q_1} = 0, \quad \frac{\partial f_3}{\partial q_2} = \frac{K_1}{J_3 + 0.5 J_2}, \quad \frac{\partial f_3}{\partial q_3} = \frac{1}{J_3 + 0.5 J_2}\left(-1.5 - 3\beta q_3^2\right)$$

At the equilibrium point, substitute $q_3 \approx 0.1538$:

$$\frac{\partial f_3}{\partial q_3} = \frac{1}{J_3 + 0.5 J_2}\left(-1.5 - 3\beta (0.1538)^2\right)$$

Substitute $J_3 = 0.5$, $J_2 = 1$, and $\beta = 5$:

$$\frac{\partial f_3}{\partial q_3} = -1.8555$$

Substitute the partial derivatives into the Jacobian matrix:

$$A = \begin{bmatrix} -\frac{B_1}{J_1} & -\frac{K_1}{J_1} & 0 \\ 1 & 0 & -0.5 \\ 0 & \frac{K_1}{J_3 + 0.5 J_2} & -1.8555 \end{bmatrix}$$

Substitute the given parameter values: - $J_1 = 2$, $J_2 = 1$, $J_3 = 0.5$ - $B_1 = 10$, $K_1 = 1000$
The Jacobian matrix becomes:

$$A = \begin{bmatrix} -5 & -500 & 0 \\ 1 & 0 & -0.5 \\ 0 & 1000 & -1.8555 \end{bmatrix}$$

Linearized State-Space Model

The linearized state-space model is:

$$\dot{\mathbf{q}} = A\mathbf{q} + B\mathbf{u}$$

where: $\mathbf{q} = [q_1, q_2, q_3]^T$ is the state vector, $\mathbf{u} = T_m$ is the input, $B = [\frac{1}{J_1}, 0, 0]^T = [0.5, 0, 0]^T$.

MATLAB CODE

The following MATLAB code simulates the response of the linearized system due to the torque (Tm) that varies from magnitude (case1) 1 Nm to 2 Nm and (case2) 1 Nm to 30 Nm.

```
function tiltrotor_gearbox3()
    % Parameters
    J1 = 2; J2 = 1; J3 = 0.5;
    K1 = 1000; B1 = 10; beta = 5;

    % Linearized system matrices (Jacobian)
    A = [-B1/J1, -K1/J1, 0;
          1,    0,     -0.5;
          0, K1/(J3 + 0.5 * J2), (-1.5 - 3 * beta * 0.1538^2) / (J3 + 0.5 * J2)];
    B = [1/J1; 0; 0];
    x0 = [0; 0; 0]; % Initial conditions [Omega1; theta1 - theta2; Omega3]
    tspan = [0 10];

    % Step input 1: Tm changes from 1 N-m to 2 N-m
    Tm1 = 1; % Step size 1 Nm
    [t1, x1] = ode45(@(t, x) A * x + B * Tm1, tspan, x0);
    theta1_theta2_1 = x1(:, 2);

    % Step input 2: Tm changes from 1 N-m to 30 N-m
    Tm2 = 29; % Step size 29 Nm
    [t2, x2] = ode45(@(t, x) A * x + B * Tm2, tspan, x0);
    theta1_theta2_2 = x2(:, 2);

    % Plot results
    figure;
    subplot(2, 1, 1);
    plot(t1, theta1_theta2_1, 'LineWidth', 1.5);
    title('Linearized␣Response:␣Tm␣=␣2␣N-m');
    xlabel('Time␣(s)'); ylabel('\theta_1␣-␣\theta_2␣(rad)');

    subplot(2, 1, 2);
    plot(t2, theta1_theta2_2, 'LineWidth', 1.5);
    title('Linearized␣Response:␣Tm␣=␣30␣N-m');
    xlabel('Time␣(s)'); ylabel('\theta_1␣-␣\theta_2␣(rad)');

    % Analyze results
    analyze_response(t1, theta1_theta2_1, 'Tm␣=␣2␣N-m');
    analyze_response(t2, theta1_theta2_2, 'Tm␣=␣30␣N-m');

    % Nested function for analysis (kept inside tiltrotor_gearbox3)
    function analyze_response(t, theta_diff, title_str)
        % Steady-state value
        steady_state = theta_diff(end);
        fprintf('%s␣-␣Steady-state␣value:␣%.4f␣rad\n', title_str, steady_state);

        % Maximum overshoot
        max_overshoot = max(theta_diff) - steady_state;
        fprintf('%s␣-␣Maximum␣overshoot:␣%.4f␣rad\n', title_str, max_overshoot);

        % Period of oscillations
        [~, locs] = findpeaks(theta_diff);
        if length(locs) >= 2
            period = mean(diff(t(locs)));
            fprintf('%s␣-␣Period␣of␣oscillations:␣%.4f␣s\n', title_str, period);
        else
            fprintf('%s␣-␣Not␣enough␣peaks␣to␣determine␣period.\n', title_str);
        end
    end

end
```

Simulation Output

Torque T_m (N-m)	Steady-State Value (rad)	Max Overshoot (rad)	Period (s)
2	0.0003	0.0006	0.1972
30	0.0079	0.0170	0.1977

Table 2: Simulation results for different torque values

Plots for the Response of the System for Change in Torque

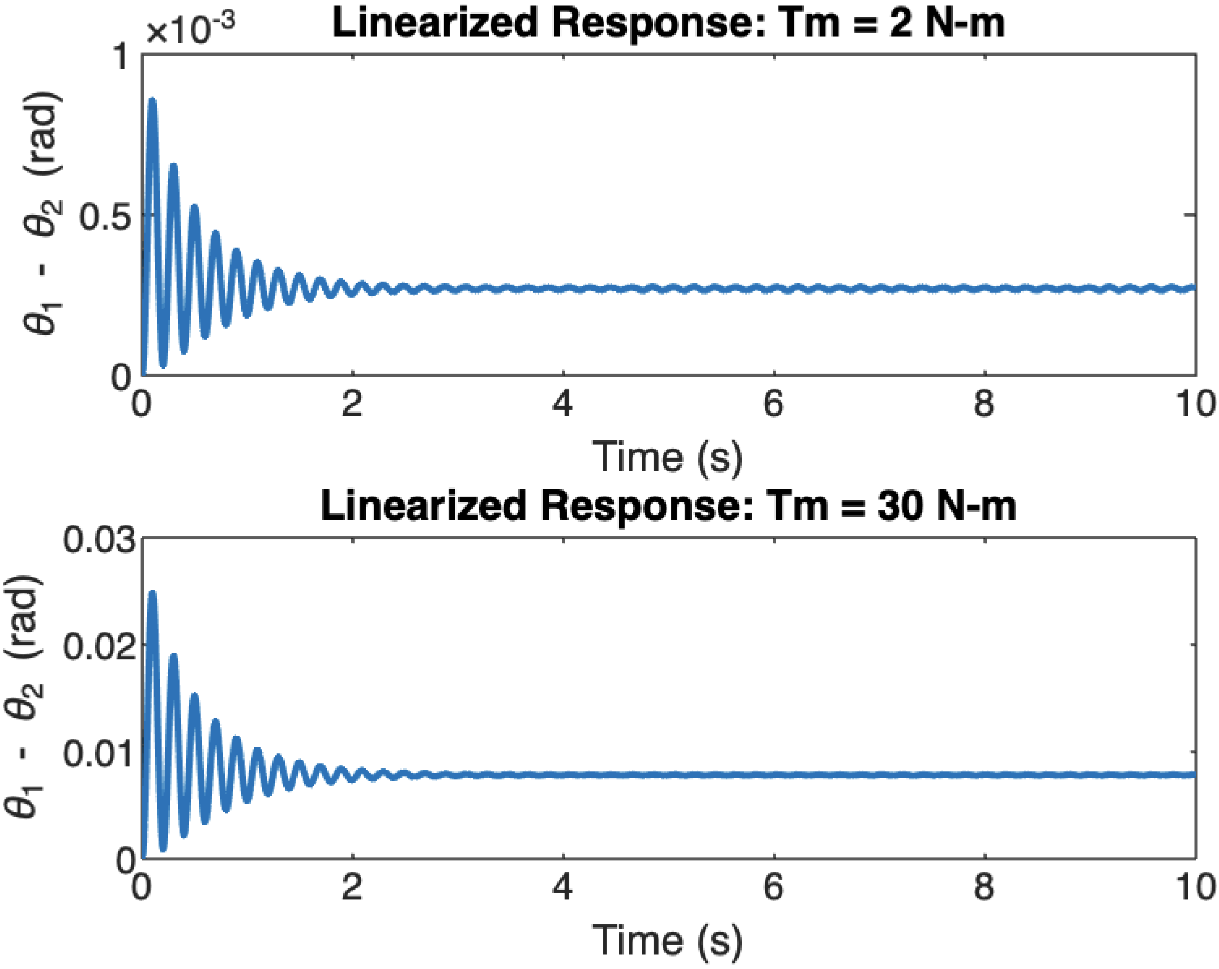


Figure 2: Plot of the response of the linear system

Error Analysis

The accuracy of the linearized model is evaluated by comparing its response with that of the nonlinear model for different values of T_m. The error in q_2 between the linear and nonlinear models is calculated as follows:

- **For $T_m = 2$ N·m:**

$$\text{Error} = \left| \frac{0.000549041 - 0.000513502}{0.000513502} \right| \times 100 \approx 6.92\%$$

  - The linearized model, developed around $T_m = 1$ N·m, shows relatively small error (6.92%) in frequency response and time-domain behavior.
  - The deviation from the nonlinear model is minor, indicating that linearization is reasonably accurate for small deviations in T_m.

- **For $T_m = 30$ N·m:**

$$\text{Error} = \left| \frac{0.00805447 - 0.0220893}{0.0220893} \right| \times 100 \approx 63.52\%$$

  - The error between the nonlinear and linear models increases significantly(63.52%).
  - The linear model fails to capture the nonlinear effects, leading to discrepancies in predicted system behavior.
  - This suggests that the linearized model is only valid for small perturbations around $T_m = 1$ N·m and does not generalize well to higher values of T_m.

Validity of Linearized Model About $T_m = 1$ N·m

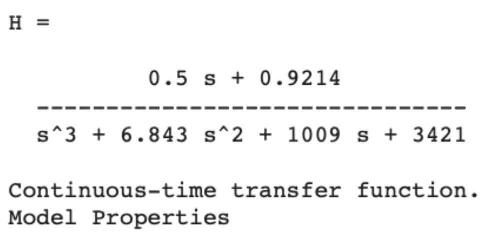
- The linearized model is only valid in a small neighborhood around $T_m = 1$ N·m.
- For $T_m = 2$ N·m, the linear approximation is still reasonable.
- For larger values such as $T_m = 30$ N·m, nonlinear effects dominate, making the linear model unreliable.

Question 4: Transfer Function Analysis: Derivation, Poles, Zeros, and Damping Characteristics using MATLAB

```
    % Given system parameters
J1 = 2; % N-m-sec^2/rad
J2 = 1; % N-m-sec^2/rad
J3 = 0.5; % N-m-sec^2/rad
K1 = 1000; % Nm/rad
B1 = 10; % N-m-sec/rad
beta = 5; % N-m-sec^3/rad^3
R2 = 0.5; % m
R3 = 0.25; % m

tspan = [0, 10]; % Simulation time span
Xini = [0, 0, 0]; % Initial conditions for [q1, q2, q3]

% Solve ODEs using ode45
[tv, Xvec] = ode45(@(t, X) systemEquations(t, X, J1, J2, J3, K1, B1, beta,2), tspan, Xini);
[tv2, Xvec2] = ode45(@(t, X) systemEquations(t, X, J1, J2, J3, K1, B1, beta,30), tspan, Xini);

%Linerization about T = 1N-m
% Initial guess for equilibrium
X0 = [0.0756, 0, 0.1512];

% Solve for equilibrium using fsolve
options = optimoptions('fsolve', 'Display', 'iter', 'TolFun', 1e-10, 'TolX', 1e-10);
Xsol = fsolve(@(X) equilibriumEquations(X, J1, J2, J3, K1, B1, beta), X0, options);

disp('Equilibrium␣Point:');
disp(Xsol);

syms omega_1 diff omega_3 Ti real
syms q1 q2 q3 Th real

% Define system equations
Tnld = 1.5 * omega_3 + beta * omega_3^3;

dq1dt = (1/J1) * (Ti - B1*omega_1 - K1*diff);
dq2dt = omega_1 - 0.5*omega_3;
dq3dt = (1 / (J3 + 0.5*J2)) * (K1*diff - Tnld);


Eq1Lin = vpa(expand(subs(dq1dt, [Ti, omega_1, diff, omega_3], [1 + Th, q1 + Xsol(1), q2+ Xsol(2), q3 + Xsol(3)])),6);
Eq3Lin = vpa(expand(subs(dq3dt, [Ti, omega_1, diff, omega_3], [1 + Th, q1 + Xsol(1), q2+ Xsol(2), q3 + Xsol(3)])),6);
Amat = double(A);
BMat = double(-B/Th);
disp(Amat);
disp(BMat);
C1 = [1 0 0];
C2 = [0 1 0];
C3 = [0 0 1];
Dmat = [0];
sys1 = ss(Amat, BMat, C1, Dmat);
sys2 = ss(Amat, BMat, C2, Dmat);
sys3 = ss(Amat, BMat, C3, Dmat);
```

Transfer Function Analysis

```
[num , den ] = ss2tf(Amat, BMat, C2, Dmat);
H = tf(num,den);
disp(H);
```



```
H =

            0.5 s + 0.9214
  --------------------------------
  s^3 + 6.843 s^2 + 1009 s + 3421

Continuous-time transfer function.
Model Properties
```

Poles and Zeros of Transfer Function

```
[p,z] = pzmap(sys2);
format longG
disp(p);
disp(z);
```

Poles =

$$\begin{gathered} -3.42998180950368 + 0i \\ -1.70644409524816 + 31.5372201396781i \\ -1.70644409524816 - 31.5372201396781i \end{gathered}$$

Zeros =

$$-1.84278$$

Dominant poles =

```
P1 = -1.70644409524816 + 31.5372201396781i;
P2 = -1.70644409524816 - 31.5372201396781i;
```

```
% natural frequency
wn = abs(P1);
disp(wn)
```

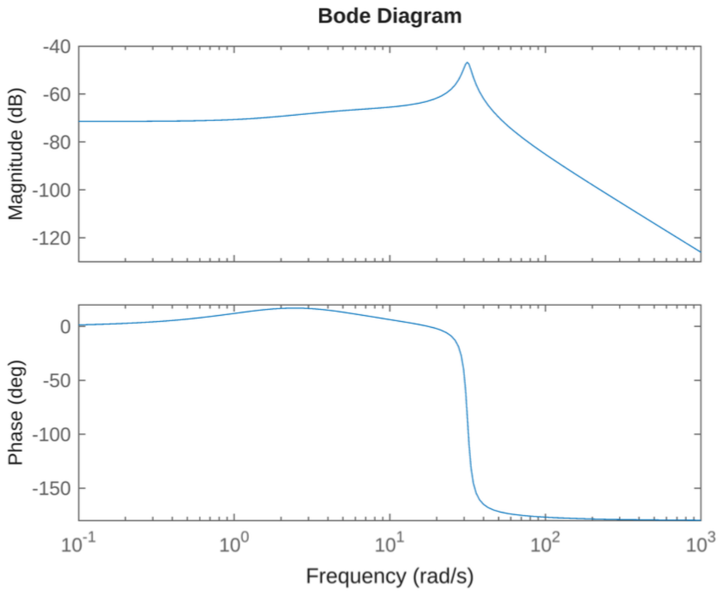
Natural Frequency = 31.5833532986718

```
%damping ratio
zeta = -real(P1) / wn;
disp(zeta)
```

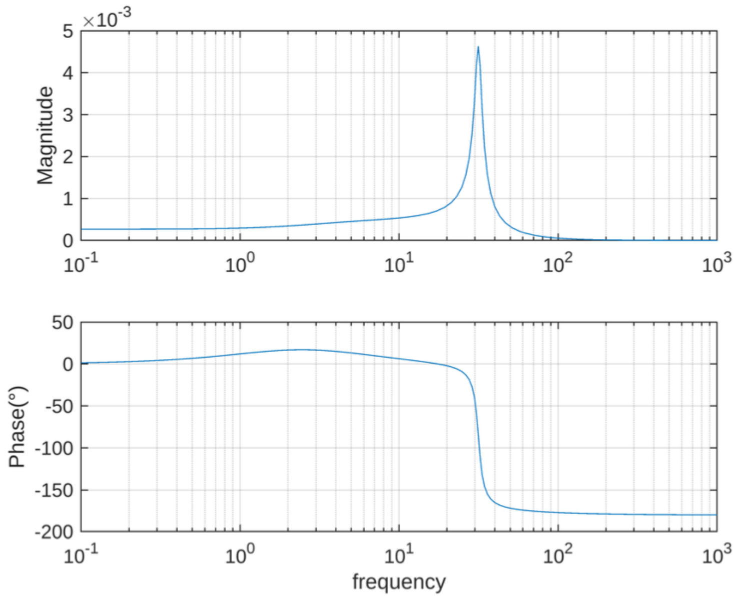
Damping ratio = 0.0540298580429685

Bode Plot

```
    bp = bodeplot(sys2);
```



```
[mag,phase,wout] = bode(sys2);
figure()
hold on
subplot(2,1,1)
semilogx(wout, squeeze(mag))
grid on
ylabel('Magnitude')
subplot(2,1,2)
semilogx(wout, squeeze(phase))
ylabel('Phase(deg)')
xlabel('frequency')
grid on
```



Function Definitions

```
function F = equilibriumEquations(X, J1, J2, J3, K1, B1, beta)
q1 = X(1);
q2 = X(2);
q3 = X(3);
Tm = 1; % Assume zero external torque at equilibrium
Tnld = 1.5 * q3 + beta * q3^3;
% Equilibrium equations (setting derivatives to zero)
F(1) = (1/J1) * (Tm - B1*q1 - K1*q2); % dq1/dt = 0
F(2) = q1 - 0.5*q3; % dq2/dt = 0
F(3) = (1 / (J3 + 0.5*J2)) * (K1*q2 - Tnld); % dq3/dt = 0
F = F’; % Return as column vector
end
function dXdt = systemEquations(t, X, J1, J2, J3, K1, B1, beta , Tm)
q1 = X(1);
q2 = X(2);
q3 = X(3);
% Nonlinear friction torque
Tnld = 1.5 * q3 + beta * q3^3;
% System of equations
dq1dt = (1/J1) * (Tm - B1*q1 - K1*q2);
dq2dt = q1 - 0.5*q3;
dq3dt = (1 / (J3 + 0.5*J2)) * (K1*q2 - Tnld);
dXdt = [dq1dt; dq2dt; dq3dt];
end
```

Explanation for Single Natural Frequency in Bode Plot

- The system has two poles: one near the imaginary axis (dominant) and another real, negative, and located far from the imaginary axis.
- Since the second pole is far from the imaginary axis, its influence on system dynamics is minimal.
- As a result, the frequency response exhibits only **one prominent peak**, corresponding to the dominant pole.
- The Bode plot reflects this by showing a single resonance peak instead of two.

Effect of Contact Stiffness in Gears on the Analysis

If the gear teeth are no longer rigid and can deform due to contact stiffness, the system dynamics will change significantly

- A new stiffness term Kc (contact stiffness) must be introduced in the model.
- Instead of assuming rigid gear meshing, the system now behaves like a spring-mass system where the teeth store and release elastic energy.
- Gear 1 is connected to Gear 2 via a shaft (no direct contact, just rotational coupling).
- Gear 2 is in contact with Gear 3 (direct contact), and this is where the contact stiffness comes into play.
- The gear meshing force is no longer instantaneously transmitted but instead follows Hooke's Law-like behavior.

  $$T_{gear} = K_c(\theta_2 - \theta_3)$$

- So the equations of motions would change to :

  $$J_1\dot{\Omega}_1 = T_m - B_1\Omega_1 - K_1(\theta_1 - \theta_2)$$

  $$J_2\dot{\Omega}_2 = K_1(\theta_1 - \theta_2) - K_c(\theta_2 - \theta_3)$$

  $$J_3\dot{\Omega}_3 = K_c(\theta_2 - \theta_3) - T_{\text{nld}}$$

- Effect on Step Response
  - Increased Damping and Reduced Overshoot
  - Lower Natural Frequency and Longer Settling Time
  - Potential gear backlash effects which cause additional oscillations or instabilities in torque transmission.
- Changes in Stability Validity of Linearization
  - Linearization Validity Becomes Weaker for Large Inputs
  - Resonance Risk in High-Speed Systems

Individual contributions in the project :

- Ambadas Joshi : Transfer Function Analysis: Derivation, Poles, Zeros, and Damping Characteristics using MATLAB and Error Analysis.
- C Disha : Nonlinear Equations of Motion , System response to Step Torque(non linear) using MATLAB and Effect of Contact Stiffness in Gears on the Analysis.
- Pallavi Undre : Linearize the system model in the vicinity of the equilibrium operating point, Tm = 1 Nm and system response to step torque (linear) using MATLAB.