Contents


```
% In the name of Allah the beneficent the merciful
% Code by NVE Team, Sharif University, Tehran, Iran
% Date : 1395/12/24
```

Cleaning

```
clc; clear; close all;
```

Variables

```
syms alpha beta gamma
syms ax ay az
```

Parameters

```
global m I T_Amp T_Freq T_phase
m = 150;
I =[5.8200   -0.8200    0.1900
   -0.8200    3.4100   -0.2100
    0.1900   -0.2100    5.5000];

M = [m*eye(3)  zeros(3,3); zeros(3,3) I];

T_Amp = 100;     % N/m
T_Freq = 50;     % Rad/s
T_phase = 0;     % Radian
```

Stiffness Matrix

Mount Positions

```
r = [ax ay az]';
r_1 = [-93 417.4 172.0]'*1e-3;
r_2 = [-56 -433.1 116.5]'*1e-3;
r_3 = [262 36.9 -68]'*1e-3;

% Cross product Matrix
B = [0 -az ay; az 0 -ax; -ay ax 0];
B_1 = double(subs(B,[ax ay az],r_1'));
B_2 = double(subs(B,[ax ay az],r_2'));
B_3 = double(subs(B,[ax ay az],r_3'));

% Mount Inclinations
o_1 = [180 10 180]'*pi/180;
o_2 = [0 0 0]'*pi/180;
o_3 = [180 0 0]'*pi/180;

% Mount Rotation Matrices
A  =  [ cos(alpha)*cos(beta), cos(alpha)*sin(beta)*sin(gamma) - cos(gamma)*sin(alpha),
sin(alpha)*sin(gamma) + cos(alpha)*cos(gamma)*sin(beta)
        cos(beta)*sin(alpha), cos(alpha)*cos(gamma) + sin(alpha)*sin(beta)*sin(gamma),
cos(gamma)*sin(alpha)*sin(beta) - cos(alpha)*sin(gamma)
                  -sin(beta),                                    cos(beta)*sin(gamma),
cos(beta)*cos(gamma)];
A_1 = double (subs(A,[alpha beta gamma],o_1'));
A_2 = double(subs(A,[alpha beta gamma],o_2'));
A_3 = double(subs(A,[alpha beta gamma],o_3'));

% Mo  unt Stiffness
k_l_1 = diag([94.6 111.3 92.4])*1e3;
k_l_2 = diag([72.8 72.8 84.4])*1e3;
k_l_3 = diag([203.9 41.7 82.0])*1e3;
k_1 = A_1*k_l_1*A_1';
k_2 = A_2*k_l_2*A_2';
k_3 = A_3*k_l_3*A_3';

% Finally! The Stiffness Matrix
K_F1 = [k_1 k_1*B_1' ; (k_1*B_1')' B_1*k_1*B_1'];
K_F2 = [k_2 k_2*B_2' ; (k_2*B_2')' B_2*k_2*B_2'];
K_F3 = [k_3 k_3*B_3' ; (k_3*B_3')' B_3*k_3*B_3'];
K = K_F1 + K_F2 + K_F3;
```

Damping Matrix

Mount Damping Coefficients

```
c_l_1 = diag([94.6 111.3 92.4]);
```

```
c_l_2 = diag([72.8 72.8 84.4]);
c_l_3 = diag([203.9 41.7 82.0]);
c_1 = A_1*c_l_1*A_1';
c_2 = A_2*c_l_2*A_2';
c_3 = A_3*c_l_3*A_3';

% Damping Matrix
C_F1 = [c_1 c_1*B_1' ; (c_1*B_1')' B_1*c_1*B_1'];
C_F2 = [c_2 c_2*B_2' ; (c_2*B_2')' B_2*c_2*B_2'];
C_F3 = [c_3 c_3*B_3' ; (c_3*B_3')' B_3*c_3*B_3'];
C = C_F1 + C_F2 + C_F3;
```

Execution : Linear

```
stp = 0.01;
F_T = 5;
% options  = [];
options = odeset('maxstep',0.001);
x0 = [0;0;0;0;0;0;0;0;0;0;0;0];     % The initial condition
[t,x] = ode45(@eng_mount, 0:stp:F_T, x0, options, M, C, K);      % solving the ODE
with the duration of 5 seconds
```

Execution : Linear without damper

```
options = odeset('maxstep',0.001);
x0 = [0;0;0;0;0;0;0;0;0;0;0;0];     % The initial condition
[~,x_wd] = ode45(@eng_mount_without_damper, 0:stp:F_T, x0, options, M, K);      %
solving the ODE with the duration of 5 seconds
```

Execution : NonLinear

options = [];

```
options = odeset('maxstep',0.001);
x0 = [0;0;0;0;0;0;0;0;0;0;0;0];     % The initial condition
[~,x_non] = ode45(@nonlinear_eng_mount, 0:stp:F_T, x0, options);      % solving the
ODE with the duration of 5 seconds
```

Execution : NonLinear without damper

options = [];

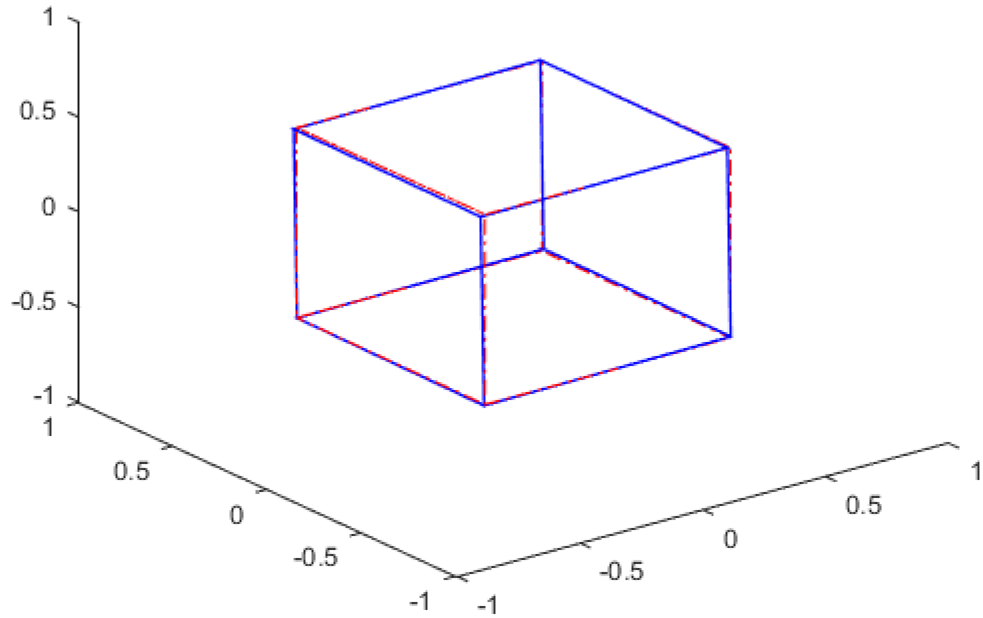
```
options = odeset('maxstep',0.001);
x0 = [0;0;0;0;0;0;0;0;0;0;0;0];     % The initial condition
[t_non,x_non_wd] = ode45(@nonlinear_eng_mount_without_damper, 0:stp:F_T, x0, options);
% solving the ODE with the duration of 5 seconds
```

FFT

```
Fs = 1/stp;                  % Sampling frequency
T = 1/Fs;                    % Sample time
L = length(t)-1;             % Length of signal
t1 = (0:L-1)*T;              % Time vector

% Signal
y = x';

NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;

f = Fs/2*linspace(0,1,NFFT/2+1);
```

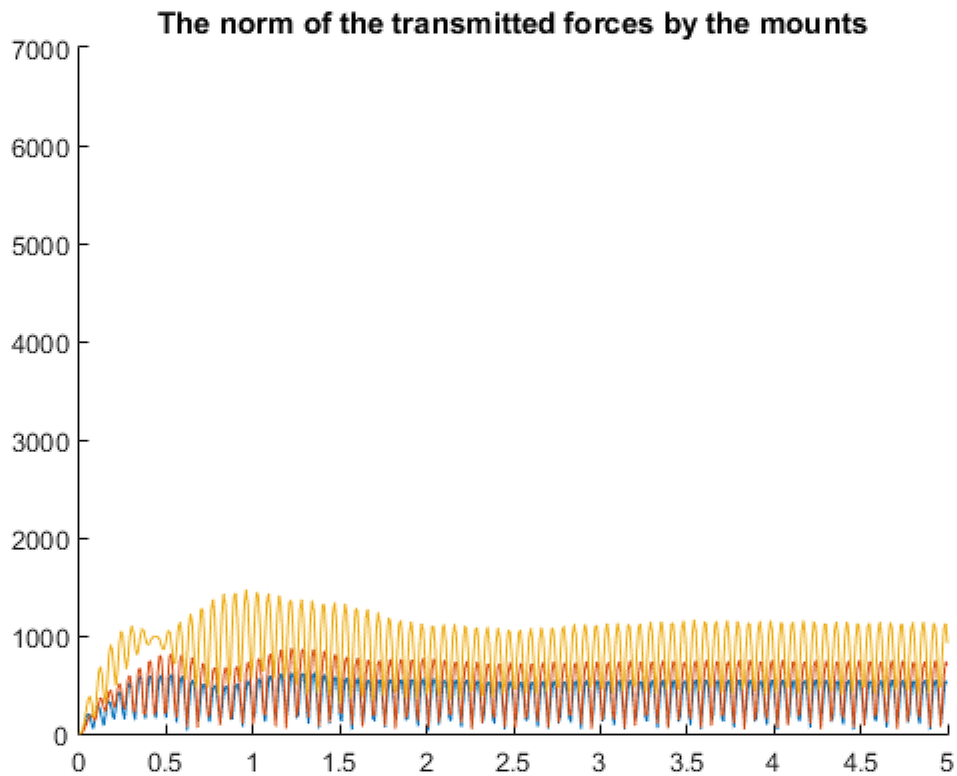
Animation

```
P1 = cube([0 0 0],[0 0 0]);     % plotting the engine in its equilibrium position

for i = 1:length(t)
    T = x(i,1:3);
    eul = x(i,4:6);
    P = cube(T,eul);
    figure(1);clf;
    title('The position and orientation of the engine')
    plot3(P1(:,1),P1(:,2),P1(:,3),'color','r','linestyle','-.','linewidth',.5)
    hold on
    plot3(P(:,1),P(:,2),P(:,3),'color','b','linewidth',1)
    axis([-1 1 -1 1 -1 1])
    pause(0.001)
end
```



Force Results

```
F_1 = zeros(length(t),3);   F_2 = F_1;  F_3 = F_1;
F_1_n = t';  F_2_n = F_1_n;   F_3_n = F_1_n;

for i = 1:length(t)

    F_1(i,:) = (-k_1*[eye(3) B_1']*x(i,1:6)' - c_1*[eye(3) B_1']*x(i,7:12)')';
    F_1_n(i) = norm(F_1(i,:));
    F_2(i,:) = (-k_2*[eye(3) B_2']*x(i,1:6)' - c_2*[eye(3) B_2']*x(i,7:12)')';
    F_2_n(i) = norm(F_2(i,:));
    F_3(i,:) = (-k_3*[eye(3) B_3']*x(i,1:6)' - c_3*[eye(3) B_3']*x(i,7:12)')';
    F_3_n(i) = norm(F_3(i,:));
    figure(2);clf;hold on;
    title('The norm of the transmitted forces by the mounts')
    plot(t,F_1_n)
    plot(t,F_2_n)
    plot(t,F_3_n)
    axis([0 t(end) 0 7000])
%     legend('Mount 1','Mount 2','Mount 3');
    pause(0.0005)
end
```



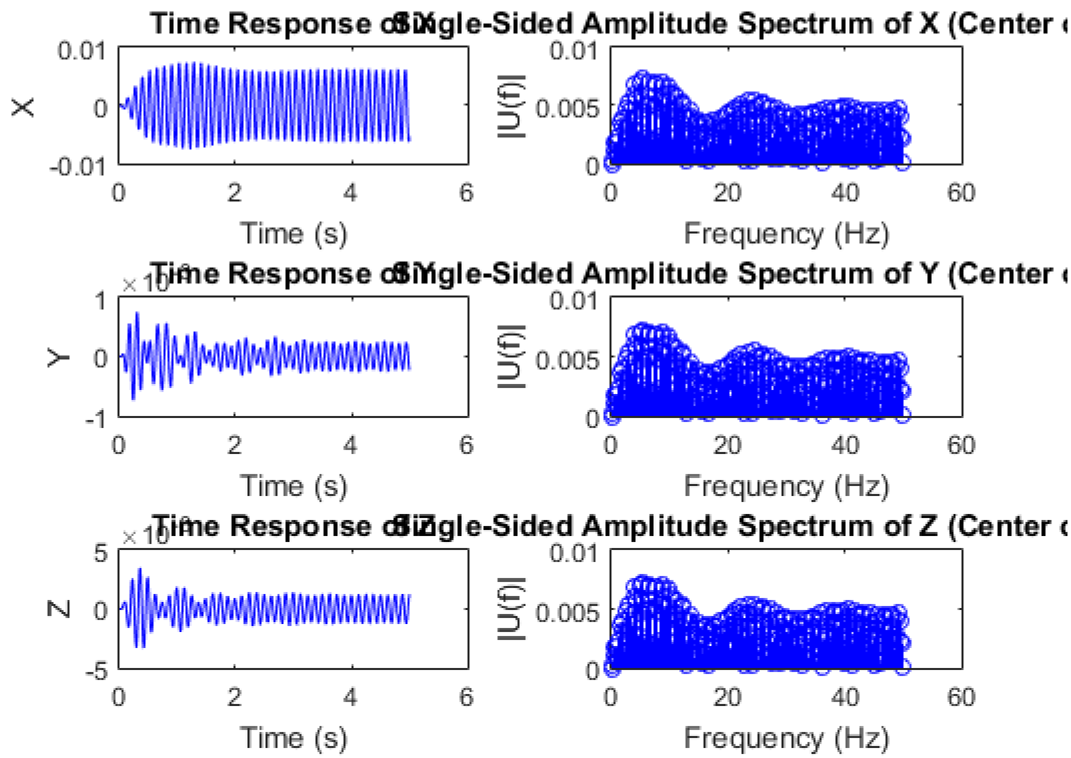
Results

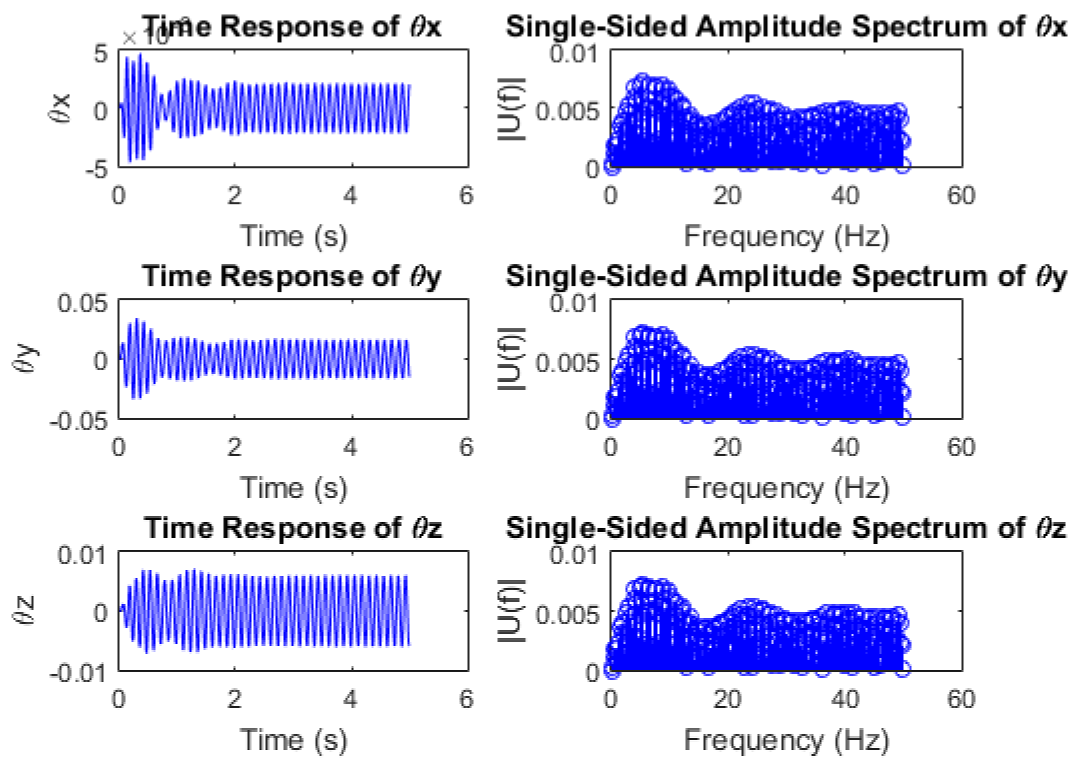
Plot Time Response of position (left column) of center of mass and their FFT (Right column)

```
figure;
subplot(3,2,1), plot(t,y(1,:),'b')
title('Time Response of X')
xlabel('Time (s)')
ylabel('X')

subplot(3,2,3), plot(t,y(2,:),'b')
title('Time Response of Y')
xlabel('Time (s)')
ylabel('Y')

subplot(3,2,5), plot(t,y(3,:),'b')
title('Time Response of Z')
xlabel('Time (s)')
ylabel('Z')

% Plot single-sided amplitude spectrum of positions in right column.
subplot(3,2,2), stem(f,2*abs(Y(1,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of X (Center of Mass)')
xlabel('Frequency (Hz)')
```

```
ylabel('|U(f)|')

subplot(3,2,4), stem(f,2*abs(Y(2,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of Y (Center of Mass)')
xlabel('Frequency (Hz)')
ylabel('|U(f)|')

subplot(3,2,6), stem(f,2*abs(Y(3,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of Z (Center of Mass)')
xlabel('Frequency (Hz)')
ylabel('|U(f)|')

figure;

% Plot Time Response of position (left column)
% of center of mass and their FFT (Right column)
subplot(3,2,1), plot(t,y(4,:),'b')
title('Time Response of \thetax')
xlabel('Time (s)')
ylabel('\thetax')

subplot(3,2,3), plot(t,y(5,:),'b')
title('Time Response of \thetay')
xlabel('Time (s)')
ylabel('\thetay')

subplot(3,2,5), plot(t,y(6,:),'b')
title('Time Response of \thetaz')
xlabel('Time (s)')
ylabel('\thetaz')

% Plot single-sided amplitude spectrum of orientations in right column.
subplot(3,2,2), stem(f,2*abs(Y(4,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of \thetax')
xlabel('Frequency (Hz)')
ylabel('|U(f)|')

subplot(3,2,4), stem(f,2*abs(Y(5,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of \thetay')
xlabel('Frequency (Hz)')
ylabel('|U(f)|')

subplot(3,2,6), stem(f,2*abs(Y(6,1:NFFT/2+1)),'b')
title('Single-Sided Amplitude Spectrum of \thetaz')
```

```
xlabel('Frequency (Hz)')
ylabel('|U(f)|')
```

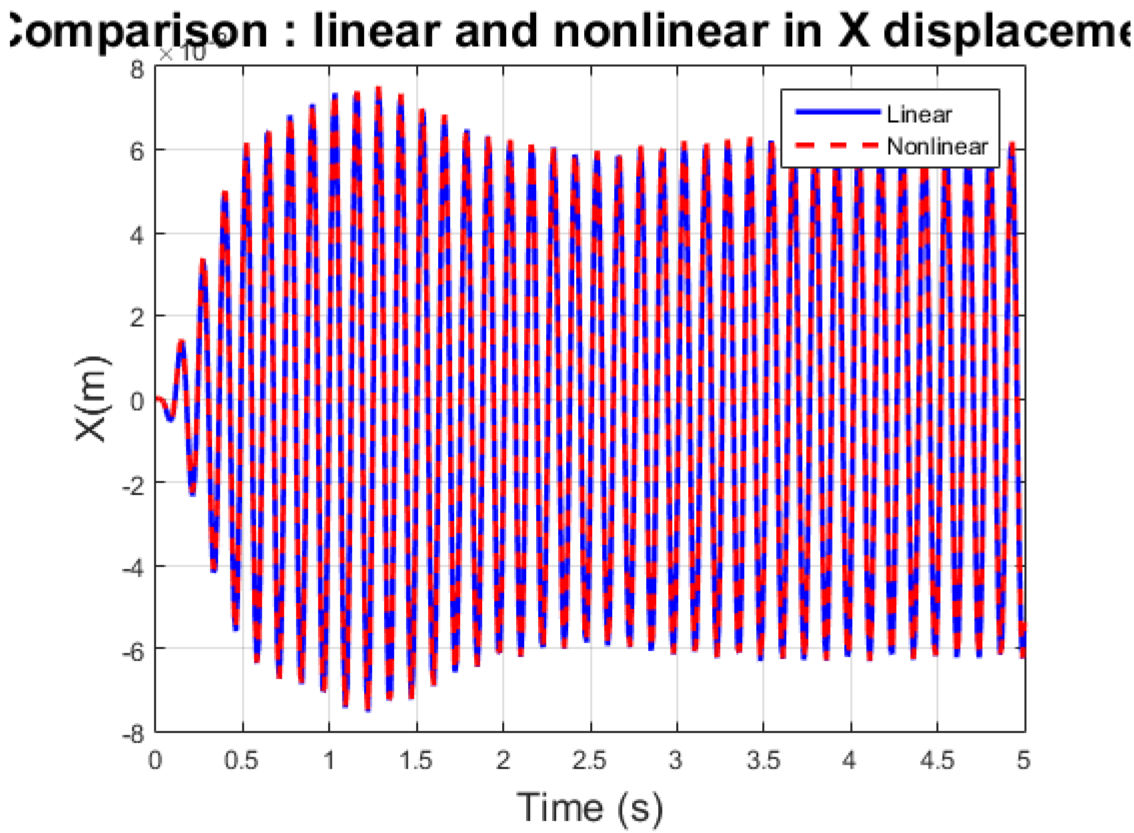

Single-Sided Amplitude Spectrum of X (Center
Single-Sided Amplitude Spectrum of Y (Center
Single-Sided Amplitude Spectrum of Z (Center
X
Y
Z
|U(f)|
Time (s)
Frequency (Hz)


Time Response of θx
Single-Sided Amplitude Spectrum of θx
Time Response of θy
Single-Sided Amplitude Spectrum of θy
Time Response of θz
Single-Sided Amplitude Spectrum of θz
θx
θy
θz
|U(f)|
Time (s)
Frequency (Hz)

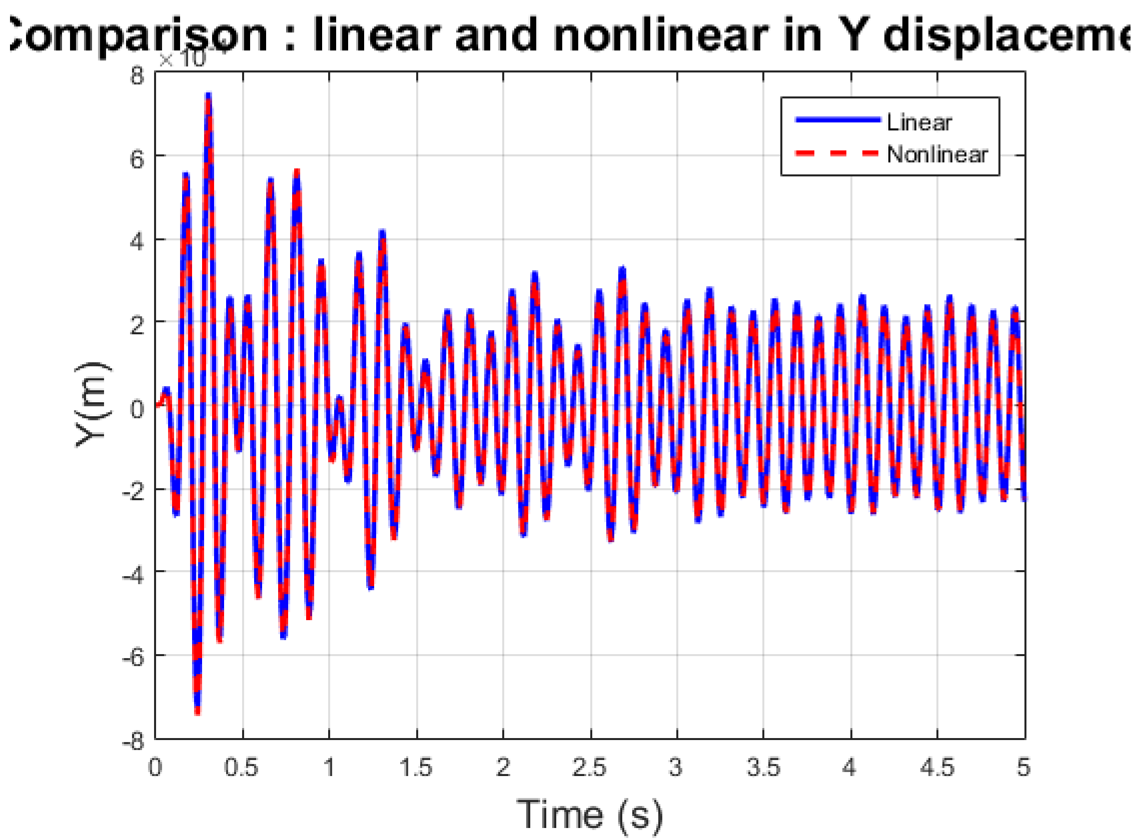
Comparison : linear and nonlinear

```
figure;
plot(t,x(:,1),'b-',t,x_non(:,1),'r--','linewidth',2);
title('Comparison : linear and nonlinear in X displacement','fontsize',18);
xlabel('Time (s)','fontsize',15);
ylabel('X(m)','fontsize',15);
grid on;
legend('Linear','Nonlinear');
grid on;

figure;
plot(t,x(:,2),'b-',t,x_non(:,2),'r--','linewidth',2);
title('Comparison : linear and nonlinear in Y displacement','fontsize',18);
xlabel('Time (s)','fontsize',15)
ylabel('Y(m)','fontsize',15);
legend('Linear','Nonlinear');
grid on;

figure;
plot(t,x(:,3),'b-',t,x_non(:,3),'r--','linewidth',2);
title('Comparison : linear and nonlinear in Z displacement','fontsize',18);
xlabel('Time (s)','fontsize',15);
ylabel('Z(m)','fontsize',15);
legend('Linear','Nonlinear');
grid on;

figure;
plot(t,x(:,4)*180/pi,'g-',t,x_non(:,4)*180/pi,'r--','linewidth',2);
title('Comparison : linear and nonlinear in X rotation','fontsize',18);
xlabel('Time (s)','fontsize',15);
ylabel('\alpha_X (Deg)','fontsize',15);
legend('Linear','Nonlinear');
grid on;

figure;
plot(t,x(:,5)*180/pi,'g-',t,x_non(:,5)*180/pi,'r--','linewidth',2);
title('Comparison : linear and nonlinear in Y rotation','fontsize',18);
xlabel('Time (s)','fontsize',15);
ylabel('\beta_Y (Deg)','fontsize',15);
legend('Linear','Nonlinear');
grid on;

figure;
plot(t,x(:,6)*180/pi,'g-',t,x_non(:,6)*180/pi,'r--','linewidth',2);
```

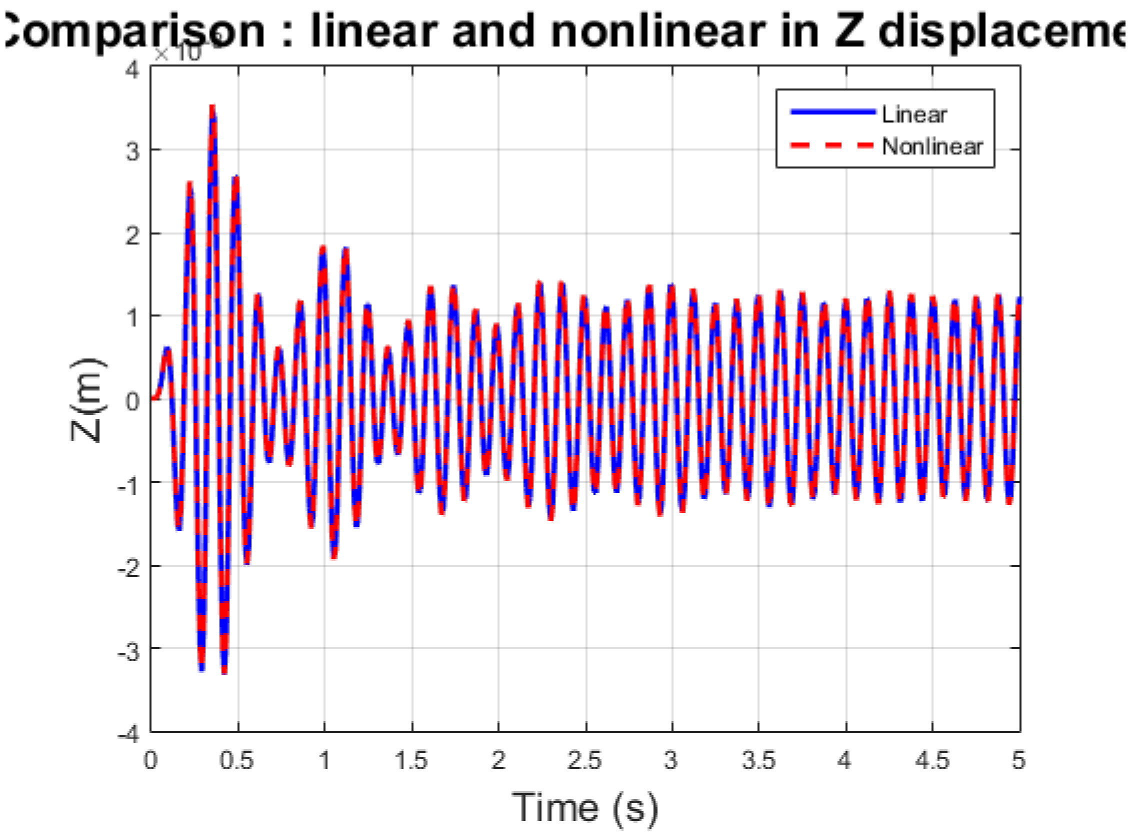
```
title('Comparison : linear and nonlinear in Z rotation','fontsize',18);
xlabel('Time (s)','fontsize',15);
ylabel('\gamma_Z (Deg)','fontsize',15);
legend('Linear','Nonlinear');
grid on;

dec = round(0.1*length(t));
e1 = sqrt(mean((x_non(:,1)-x(:,1)).^2))/max(abs(x(end-dec:end,1)))*100;
e2 = sqrt(mean((x_non(:,2)-x(:,2)).^2))/max(abs(x(end-dec:end,2)))*100;
e3 = sqrt(mean((x_non(:,3)-x(:,3)).^2))/max(abs(x(end-dec:end,3)))*100;
e4 = sqrt(mean((x_non(:,4)-x(:,4)).^2))/max(abs(x(end-dec:end,4)))*100;
e5 = sqrt(mean((x_non(:,5)-x(:,5)).^2))/max(abs(x(end-dec:end,5)))*100;
e6 = sqrt(mean((x_non(:,6)-x(:,6)).^2))/max(abs(x(end-dec:end,6)))*100;

ex = linspace(1,6,6);
ey = [e1 e2 e3 e4 e5 e6];
figure;
plot(ex(1),ey(1),'bs',ex(2),ey(2),'rs',ex(3),ey(3),'ms',ex(4),ey(4),'ys',ex(5),ey(5),'
gs',ex(6),ey(6),'ks','linewidth',4);
title('Max Error Relative To Linear Signal','fontsize',18);
xlabel('coordinates','fontsize',15);
ylabel('Percent of Error','fontsize',15);
legend('X','Y','Z','\theta_X','\theta_Y','\theta_Z');
grid on;
```



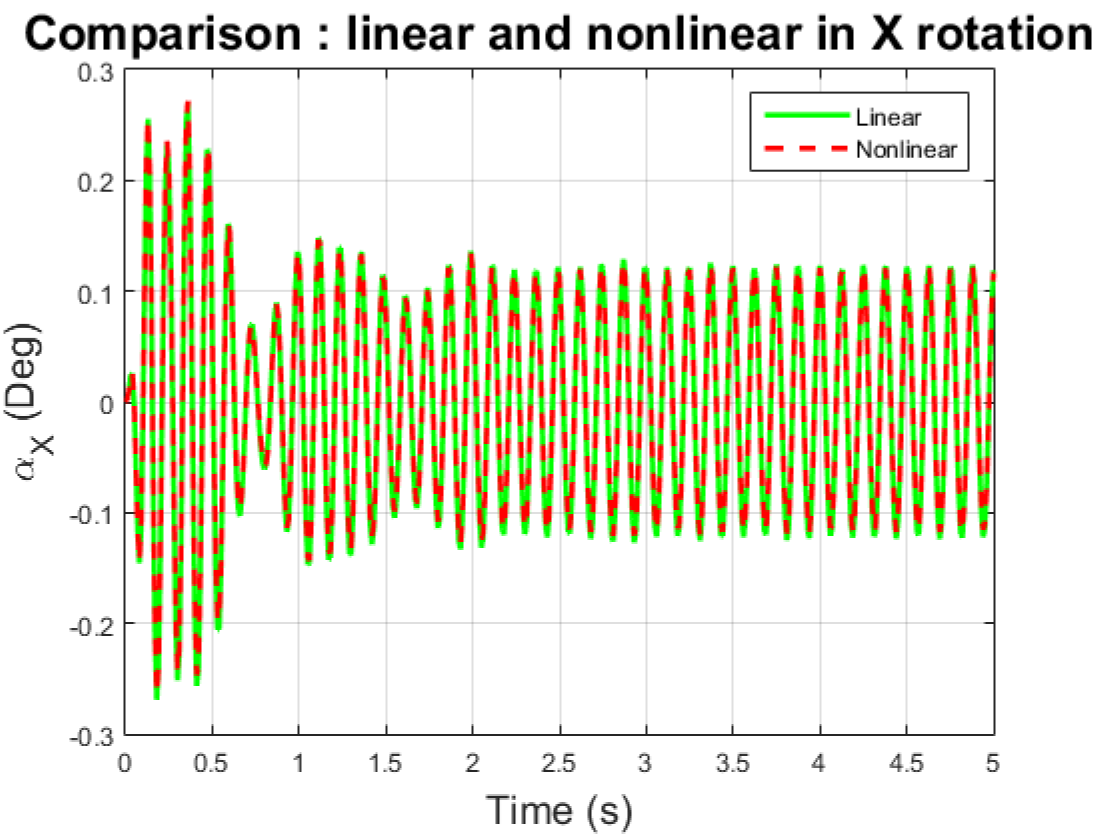
Comparison : linear and nonlinear in X displaceme
× 10
8
6
4
2
0
-2
-4
-6
-8
X(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (s)
Linear
Nonlinear



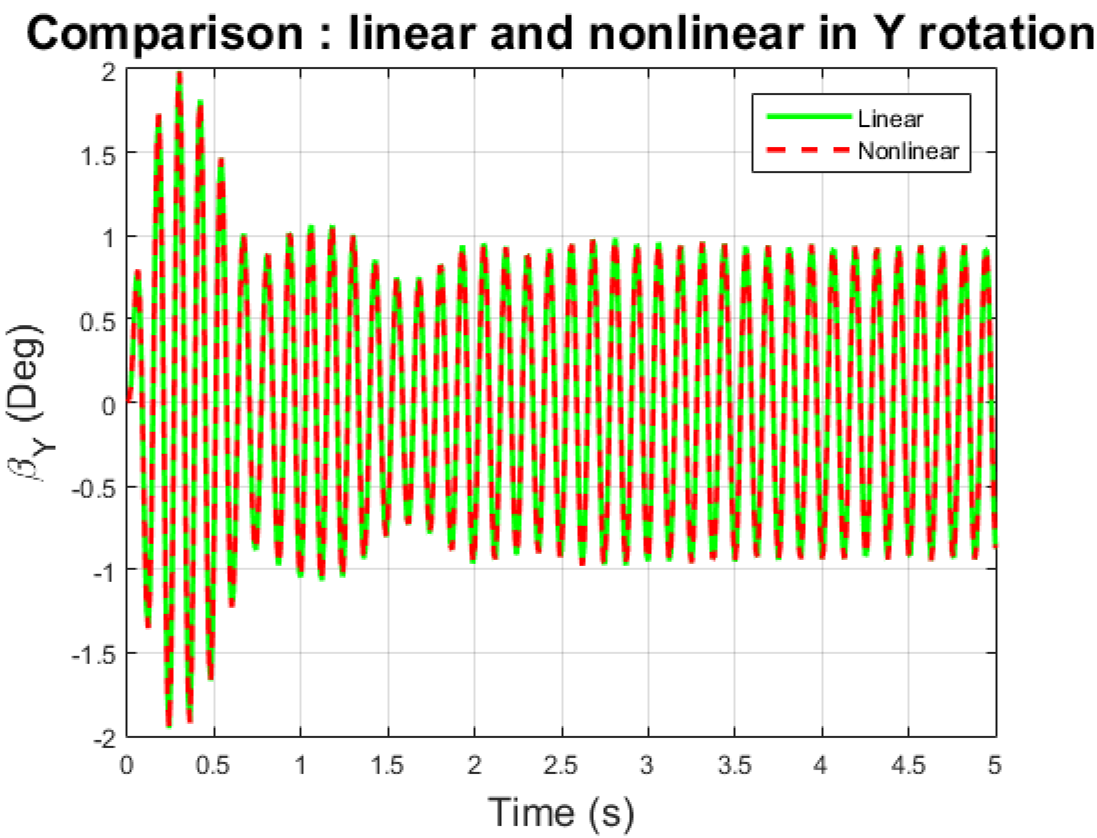
Comparison : linear and nonlinear in Y displaceme
× 10
8
6
4
2
0
-2
-4
-6
-8
Y(m)
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (s)
Linear
Nonlinear

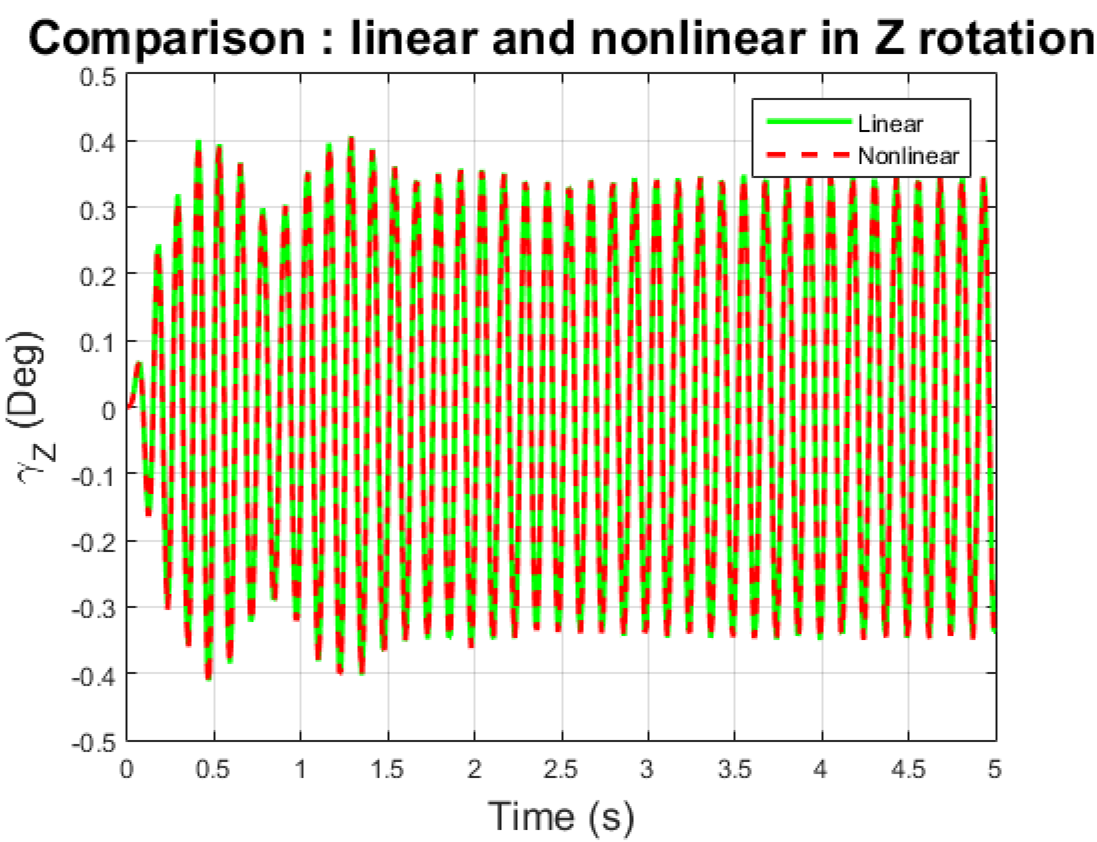


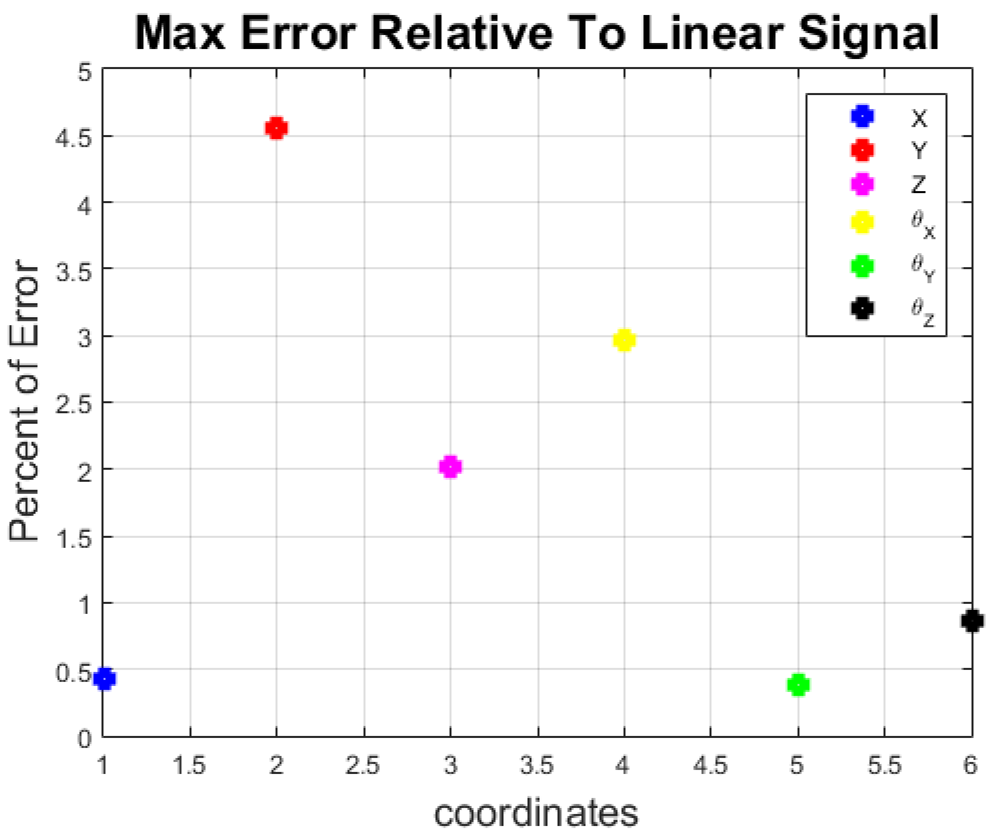
Comparison : linear and nonlinear in Z displaceme
Linear
Nonlinear
Z(m)
Time (s)



Comparison : linear and nonlinear in X rotation
Linear
Nonlinear
α_X (Deg)
Time (s)


Comparison : linear and nonlinear in Y rotation
Linear
Nonlinear
β_Y (Deg)
Time (s)


Comparison : linear and nonlinear in Z rotation
Linear
Nonlinear
γ_Z (Deg)
Time (s)



Max Error Relative To Linear Signal
Percent of Error
coordinates
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
X
Y
Z
θ_X
θ_Y
θ_Z