Mechanical Vibrations

Analysis of a Beam-Cart System

(Project of May 2021)

Gianmarco Lavacca, *224558*

September 6, 2024

I. Introduction

The purpose of this report is to analyse a mechanical system (as shown in Figure 1) and its behavior once subjected to a periodic force generated by a shaker. The laboratory experience in question is divided into two parts:

- 1 DoF: only the beam anchored to the frame is considered
- 2 DoF: the whole system (beam and cart) is considered

Five tests were carried out for each of the two phases of the experience. The analysis of the results aims at determining some specific parameters of the system, such as resonance frequency, stiffness constant of the elastic bodies, damping ratio, period of oscillation and so on.
In addition to this, we want to determine the transfer functions that link the force exerted on the system to the accelerations of the two bodies (cart and beam).

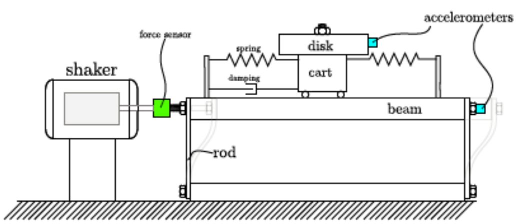

Figure 1: System sketch with accelerometers and force sensor

Every procedure presented in this report has been performed for each of the 5 tests.

II. 1 DoF System

As shown below, the single DoF system is composed of a cart with a disk fixed onto it. They are connected to the frame by two springs (with stiffness $k/2$) and a damper (with damping coefficient c).

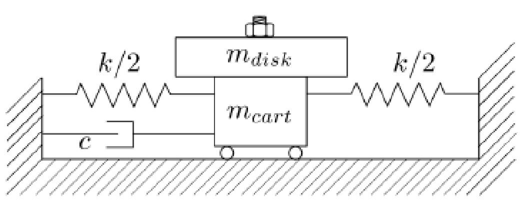

Figure 2: 1 DoF system

Therefore, considering x the DoF of the system, the equation of motion will be as follows:

$$M\ddot{x} + c\dot{x} + kx = f(t)$$

Where M is the total mass of the cart and disk and $f(t)$ is the periodic force to which the body is subjected.

A. Signal Filtering

The data obtained during the laboratory experience were loaded into Matlab. Given that those were raw signals, they had to be filtered to remove noise as much as possible. The original and filtered signals are shown in Figure 3.
Without the noise, it was possible to identify the peaks precisely and proceed to the calculation of the required parameters.

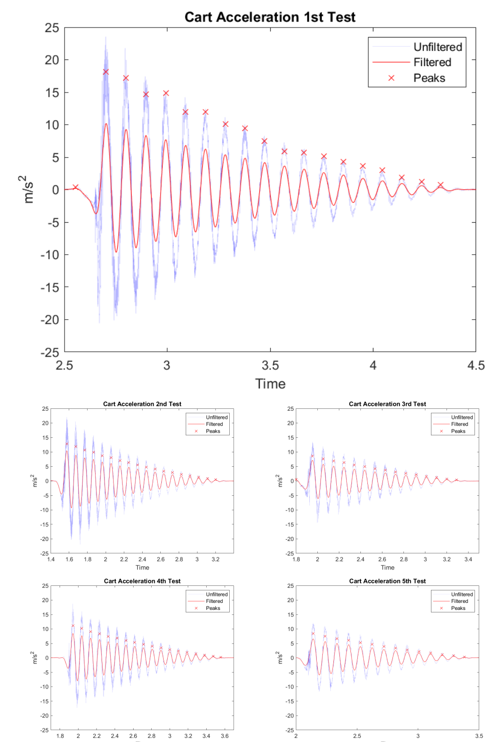

Figure 3: Double Filtered Cart Acceleration

B. Oscillation Period and Damping Ratio

The oscillation period was calculated by averaging the individual time intervals between one peak and the next. Therefore, first the relevant peaks of the experiment were isolated, since the system was only set in motion for a certain period of time; then, once the number of useful peaks and their respective sampling time had been identified, the average on their time intervals could be performed.

$$T = \frac{t_{tot}}{n_{peaks} - 1}$$

To calculate the damping ratio, instead, the amplitude of the first (strongest) peak and the one after it was used to obtain:

$$\delta = \frac{\log(\frac{x_1}{x_2})}{m_{tot}}$$

and then the damping ratio was derived:

$$\xi = \frac{\delta}{\sqrt{4\pi^2 + \delta^2}}$$

At this point, mean values and standard deviations across the five tests were derived, obtaining:

$$\begin{cases} \hat{T} = \overline{T} \pm \sigma_T = (0.0952 \pm 0.0002)s \\ \hat{\xi} = \overline{\xi} \pm \sigma_\xi = (0.0355 \pm 0.0020) \end{cases}$$

We can safely say that the tests' results are consistent and the calculations accurate since the standard deviations are small enough with respect to the mean values of our parameters.

C. Natural Frequency, Stiffness, Damping Coefficient

With the knowledge of the oscillation period and the damping ratio, the other parameters were calculated through the following equations:

$$\begin{cases} \omega_n = \frac{2\pi}{T\sqrt{1-\xi^2}} \\ k = \omega_n^2 M \\ c = 2\xi\omega_n M \end{cases}$$

Then, as done in the previous paragraph, their mean values and standard deviations were calculated across the five tests performed in the laboratory experience:

$$\begin{cases} \hat{\omega_n} = \overline{\omega}_n \pm \sigma_{\omega_n} = (66.0218 \pm 0.1402)\frac{rad}{s} \\ \hat{k} = \overline{k} \pm \sigma_k = (2247 \pm 9.5451)\frac{N}{m} \\ \hat{c} = \overline{c} \pm \sigma_c = (2.4198 \pm 0.1392)\frac{Ns}{m} \end{cases}$$

The same considerations as before about consistency and accuracy of the calculations can be applied here.

III. 2 DoF System

In this section the entire system is analysed: the beam supporting the cart is connected to the frame via two rods with elastic constant k_{rod}. This brings the number of DoF to two. The system can then be modeled as shown in the following figure:

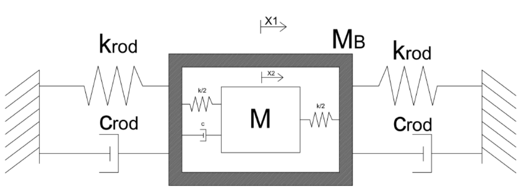

Figure 4: 2 DoF system

Considering the way the rods are connected to the beam, the elastic constant was calculated as:

$$k_{rod} = \frac{12EI}{l^3}$$

The equations of motion of the system are as follows:

$$\begin{bmatrix} M_{beam} & 0 \\ 0 & M_{tot} \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} 2c_r + c & -c \\ -c & c \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 2k_r + k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f(t) \\ 0 \end{bmatrix}$$

A. Analytical Transfer Functions

From the equations of motion we can obtain the transfer functions between the force applied to the system and the displacements of both degrees of freedom.
Considering that the Laplace transform of the acceleration is $A_i(s) = s^2 X_i(s)$, the transfer functions for the accelerations will be:

$$G_b(s) = \frac{s^2(M_{tot}s^2 + cs + k)}{C_1 s^4 + C_2 s^3 + C_3 s^2 + C_4 s + C_5}$$

$$G_c(s) = \frac{s^2(cs + k)}{C_1 s^4 + C_2 s^3 + C_3 s^2 + C_4 s + C_5}$$

with:

$$\begin{cases} C_1 = M_{tot} M_{beam} \\ C_2 = (2c_r + c)M_{tot} + M_{beam}c \\ C_3 = (2k_r + k)M_{tot} + 2c_r c + M_{beam}k \\ C_4 = 2k_r c + 2c_r k \\ C_5 = 2k_r k \end{cases}$$

Including a $\pm 3\sigma_i$ $(i = c, k)$ uncertainty on the parameters found in the previous section, we can plot the transfer functions and find that the range of uncertainty is very small.

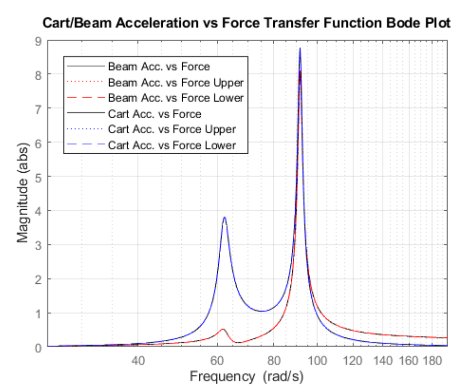

Figure 5: Cart-Beam Acceleration Analytical TFs

B. Experimental Transfer Functions

After obtaining and plotting the analytical transfer functions, the next step is to check the consistency of the experimental transfer functions. First we estimate them from the data of each test:

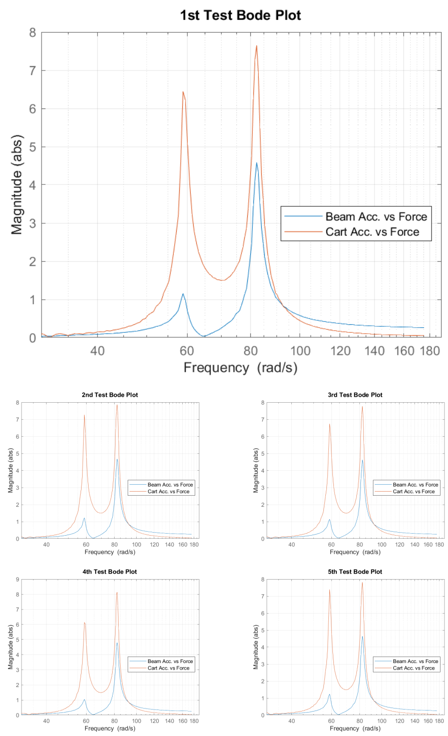

Figure 6: Cart-Beam Acceleration Experimental TFs

The mean experimental transfer function is then plotted:

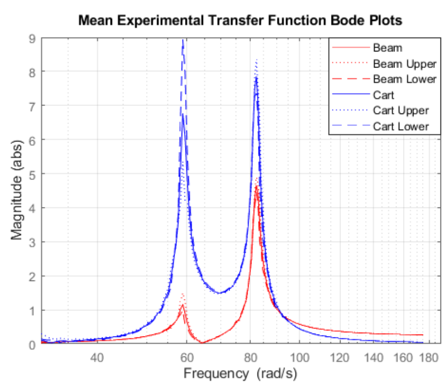

Figure 7: Cart-Beam Acceleration Mean Experimental TF

C. Analytical vs. Experimental Transfer Functions

Pairing the experimental transfer function of each body with its analytical counterpart (8) we can see that there is not a perfect fit. This is due to the presence of uncertainty both on the analytical and experimental formulation.

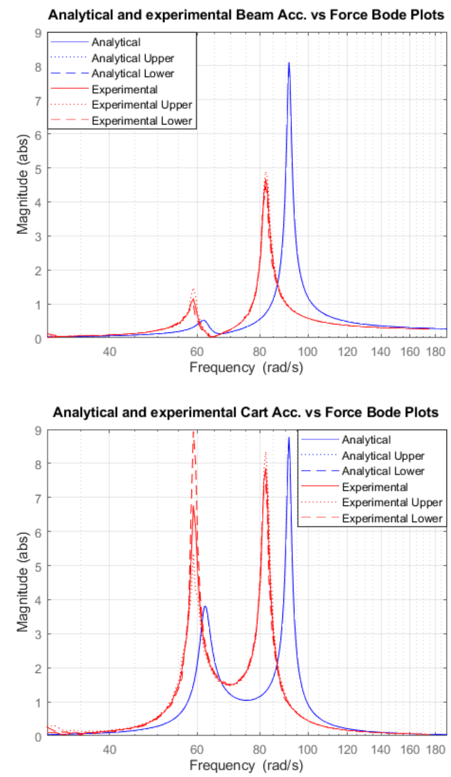

Figure 8: Cart-Beam Analytical VS Experimental TFs

D. Fitting Transfer Functions

Given the poor fit between analytical and experimental, we can now consider the damping and the stiffness coefficients of both rods and springs as unknowns and try to optimize them making the analytical formulation fit the experimental data.
This is performed by minimizing the error between the two curves through a nonlinear programming solver.
The optimized parameter found are:

$$\begin{cases} k = 294.787\frac{N}{m} \\ c = -11.204\frac{Ns}{m} \\ k_r = 1.625 \times 10^4\frac{N}{m} \\ c_r = 6.103\frac{Ns}{m} \end{cases} \qquad \begin{cases} k = 2.587 \times 10^3\frac{N}{m} \\ c = 1.585 \times 10^{-7}\frac{Ns}{m} \\ k_r = 1.133 \times 10^4\frac{N}{m} \\ c_r = -1.086 \times 10^{-6}\frac{Ns}{m} \end{cases}$$

for the beam and the cart transfer functions respectively.
We can see that these values are not comparable to the analytical ones. Moreover, one of the two damping factors in each set is negative, therefore not realistic. This results could be due to a bad optimization by the solver or to the definition of the error given to it.

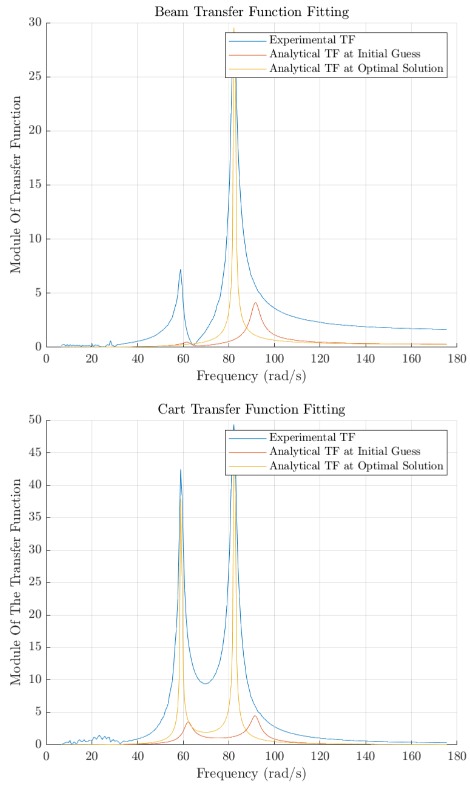

Figure 9: Cart-Beam TFs Fitting

References

[1] E. Dalla Ricca. Analysis of a beam-cart system: Laboratory experience. Technical report, University of Trento, 2021.

[1]

```
%% Preliminary Commands
clc;
close all;
clear all;

%% Known Quantities
m_cart = 0.3759; %Kg
m_disk = 0.1396; %Kg
m_tot = m_cart + m_disk; %Kg

%% Experimental Data Acquisition
OneDOF1 = readtable('Laboratory_Data\1dof_1.txt');
OneDOF2 = readtable('Laboratory_Data\1dof_2.txt');
OneDOF3 = readtable('Laboratory_Data\1dof_3.txt');
OneDOF4 = readtable('Laboratory_Data\1dof_4.txt');
OneDOF5 = readtable('Laboratory_Data\1dof_5.txt');

time1 = OneDOF1(:,1);
s_noise1 = OneDOF1(:,3);
time1 = table2array(time1);
s_noise1 = table2array(s_noise1);

time2 = OneDOF2(:,1);
s_noise2 = OneDOF2(:,3);
time2 = table2array(time2);
s_noise2 = table2array(s_noise2);

time3 = OneDOF3(:,1);
s_noise3 = OneDOF3(:,3);
s_noise3 = table2array(s_noise3);
time3 = table2array(time3);

time4 = OneDOF4(:,1);
s_noise4 = OneDOF4(:,3);
time4 = table2array(time4);
s_noise4 = table2array(s_noise4);

time5 = OneDOF5(:,1);
s_noise5 = OneDOF5(:,3);
time5 = table2array(time5);
s_noise5 = table2array(s_noise5);

%% Signal Filtering
n1 = 1000;
n2 = 1000;
tot = length(s_noise1);
s_temp1 = smooth(s_noise1,n1/tot);
s_smooth1 = smooth(s_temp1,n2/tot);
tot = length(s_noise2);
s_temp2 = smooth(s_noise2,n1/tot);
s_smooth2 = smooth(s_temp2,n2/tot);
tot = length(s_noise3);
s_temp3 = smooth(s_noise3,n1/tot);
s_smooth3 = smooth(s_temp3,n2/tot);
tot = length(s_noise4);
```

```
s_temp4 = smooth(s_noise4,n1/tot);
s_smooth4 = smooth(s_temp4,n2/tot);
tot = length(s_noise5);
s_temp5 = smooth(s_noise5,n1/tot);
s_smooth5 = smooth(s_temp5,n2/tot);

%% Peak Finding
[p_val1,p_loc1] = findpeaks(s_smooth1,'MinPeakDistance',0.09,'MinPeakHeight',0.1);
[p_val2,p_loc2] = findpeaks(s_smooth2,'MinPeakDistance',0.09,'MinPeakHeight',0.1);
[p_val3,p_loc3] = findpeaks(s_smooth3,'MinPeakDistance',0.09,'MinPeakHeight',0.1);
[p_val4,p_loc4] = findpeaks(s_smooth4,'MinPeakDistance',0.09,'MinPeakHeight',0.1);
[p_val5,p_loc5] = findpeaks(s_smooth5,'MinPeakDistance',0.09,'MinPeakHeight',0.1);

%% Plot First Test
figure
plot(time1,s_noise1,'Color',[0,0,1,0.1])

hold on

plot(time1,s_smooth1,'r')

hold on

plot(time1(p_loc1),s_noise1(p_loc1),'rx')
xlabel('Time')
ylabel('m/s^2')
xlim([2.5,4.5]);
ylim([-25,25]);
title('Cart Acceleration 1st Test')
legend('Unfiltered','Filtered','Peaks')

hold off
saveas(gcf, 'Plots\1. Cart Acceleration 1st Test.png');

%% Plot Second Test
figure
plot(time2,s_noise2,'Color',[0,0,1,0.1])

hold on

plot(time2,s_smooth2,'r')

hold on

plot(time2(p_loc2),s_temp2(p_loc2),'rx')
xlabel('Time')
ylabel('m/s^2')
xlim([1.4,3.4]);
ylim([-25,25]);
title('Cart Acceleration 2nd Test')
legend('Unfiltered','Filtered','Peaks')

hold off
saveas(gcf, 'Plots\2. Cart Acceleration 2nd Test.png');
```

```
%% Plot Third Test
figure
plot(time3,s_noise3,'Color',[0,0,1,0.1])

hold on

plot(time3,s_smooth3,'r')

hold on

plot(time3(p_loc3),s_temp3(p_loc3),'rx')
xlabel('Time')
ylabel('m/s^2')
xlim([1.8,3.5]);
ylim([-25,25]);
title('Cart Acceleration 3rd Test')
legend('Unfiltered','Filtered','Peaks')

hold off
saveas(gcf, 'Plots\3. Cart Acceleration 3rd Test.png');

%% Plot Fourth Test
figure
plot(time4,s_noise4,'Color',[0,0,1,0.1])

hold on

plot(time4,s_smooth4,'r')

hold on

plot(time4(p_loc4),s_temp4(p_loc4),'rx')
xlabel('Time')
ylabel('m/s^2')
xlim([1.7,3.7]);
ylim([-25,25]);
title('Cart Acceleration 4th Test')
legend('Unfiltered','Filtered','Peaks')

hold off
saveas(gcf, 'Plots\4. Cart Acceleration 4th Test.png');

%% Plot Fifth Test
figure
plot(time5,s_noise5,'Color',[0,0,1,0.1])

hold on

plot(time5,s_smooth5,'r')

hold on

plot(time5(p_loc5),s_temp5(p_loc5),'rx')
xlabel('Time')
ylabel('m/s^2')
```

```
xlim([2,3.5]);
ylim([-25,25]);
title('Cart Acceleration 5th Test')
legend('Unfiltered','Filtered','Peaks')

hold off
saveas(gcf, 'Plots\5. Cart Acceleration 5th Test.png');

%% Damping Ratio
[x_val1,x_loc1] = max(p_val1);
[x_val2,x_loc2] = max(p_val2);
[x_val3,x_loc3] = max(p_val3);
[x_val4,x_loc4] = max(p_val4);
[x_val5,x_loc5] = max(p_val5);

Delta1 = log(p_val1(x_loc1+1)/p_val1(x_loc1+2))/m_tot;
Delta2 = log(p_val2(x_loc2+1)/p_val2(x_loc2+2))/m_tot;
Delta3 = log(p_val3(x_loc3+1)/p_val3(x_loc3+2))/m_tot;
Delta4 = log(p_val4(x_loc4+1)/p_val4(x_loc4+2))/m_tot;
Delta5 = log(p_val5(x_loc5+1)/p_val5(x_loc5+2))/m_tot;

Xi1 = Delta1/sqrt(4*(pi^2)+Delta1^2);
Xi2 = Delta2/sqrt(4*(pi^2)+Delta2^2);
Xi3 = Delta3/sqrt(4*(pi^2)+Delta3^2);
Xi4 = Delta4/sqrt(4*(pi^2)+Delta4^2);
Xi5 = Delta5/sqrt(4*(pi^2)+Delta5^2);
Xi = [Xi1,Xi2,Xi3,Xi4,Xi5];

MeanXi = mean(Xi);
SigmaXi = std(Xi,1);

%% Period
T1 = mean(diff(time1(p_loc1(x_loc1+1:end)))); %s
T2 = mean(diff(time2(p_loc2(x_loc2+1:end)))); %s
T3 = mean(diff(time3(p_loc3(x_loc3+1:end)))); %s
T4 = mean(diff(time4(p_loc4(x_loc4+1:end)))); %s
T5 = mean(diff(time5(p_loc5(x_loc5+1:end)))); %s
T = [T1,T2,T3,T4,T5];

MeanT = mean(T);
SigmaT = std(T,1);

%% Damped Natural Frequency
OmegaD1 = 2*pi/T1; %Hz
OmegaD2 = 2*pi/T2; %Hz
OmegaD3 = 2*pi/T3; %Hz
OmegaD4 = 2*pi/T4; %Hz
OmegaD5 = 2*pi/T5; %Hz

%% Natural Frequency
OmegaN1 = OmegaD1/sqrt(1-(Xi1^2)); %rad/s
OmegaN2 = OmegaD2/sqrt(1-(Xi2^2)); %rad/s
OmegaN3 = OmegaD3/sqrt(1-(Xi3^2)); %rad/s
OmegaN4 = OmegaD4/sqrt(1-(Xi4^2)); %rad/s
OmegaN5 = OmegaD5/sqrt(1-(Xi5^2)); %rad/s
```

```matlab
OmegaN = [OmegaN1,OmegaN2,OmegaN3,OmegaN4,OmegaN5];

MeanOmegaN = mean(OmegaN);
SigmaOmegaN = std(OmegaN,1);

%% Stiffness Coefficient
K1 = m_tot*(OmegaN1^2);
K2 = m_tot*(OmegaN2^2);
K3 = m_tot*(OmegaN3^2);
K4 = m_tot*(OmegaN4^2);
K5 = m_tot*(OmegaN5^2);
K = [K1,K2,K3,K4,K5];

MeanK = mean(K);
SigmaK = std(K,1);

%% Damping Coefficient
C1 = 2*m_tot*Xi1*OmegaN1;
C2 = 2*m_tot*Xi2*OmegaN2;
C3 = 2*m_tot*Xi3*OmegaN3;
C4 = 2*m_tot*Xi4*OmegaN4;
C5 = 2*m_tot*Xi5*OmegaN5;
C = [C1,C2,C3,C4,C5];

MeanC = mean(C);
SigmaC = std(C,1);

%% Data File Writing
save('Data_SingleDOF','MeanK','SigmaK','MeanC','SigmaC','MeanOmegaN','SigmaOmegaN');
```

```
%% Preliminary Commands
clc;
close all;
clear all;

%% Known Quantities
Young_modulus = 210; %GPa
Density = 7850; %Kg/m^3

m_cart = 0.3759; %Kg
m_disk = 0.1396; %Kg
m_beam = 4.7764; %Kg
m_shaker = 0.2000; %Kg
m_tot = m_cart + m_disk; %Kg

rod_length = 0.1; %m
rod_width = 0.025; %m
rod_thickness = 0.0015; %m
rod_damping_ratio = 0.01;
rod_inertia = rod_width*(rod_thickness^3)/12; %mm^4
rod_stiffness = 12*(Young_modulus*10^9)*rod_inertia/(rod_length^3);

m_rod = Density*rod_width*rod_thickness*rod_length; %Kg
rod_damping_factor = 2*rod_damping_ratio*sqrt(rod_stiffness*m_beam);

beam_length = 605; %mm
beam_width = 30; %mm

Fs = 6400; %Hz
min_Freq = 5; %Hz
max_Freq = 30; %Hz
Duration = 40; %s

load('Data_SingleDOF');

%% Experimental Data Acquisition
TwoDOF1 = readtable('Laboratory_Data\2dof_1.txt');
TwoDOF2 = readtable('Laboratory_Data\2dof_2.txt');
TwoDOF3 = readtable('Laboratory_Data\2dof_3.txt');
TwoDOF4 = readtable('Laboratory_Data\2dof_4.txt');
TwoDOF5 = readtable('Laboratory_Data\2dof_5.txt');

time1 = TwoDOF1(:,1);
force1 = TwoDOF1(:,2);
cart_noise1 = TwoDOF1(:,3);
beam_noise1 = TwoDOF1(:,4);
time1 = table2array(time1);
force1 = table2array(force1);
cart_noise1 = table2array(cart_noise1);
beam_noise1 = table2array(beam_noise1);

time2 = TwoDOF2(:,1);
force2 = TwoDOF2(:,2);
cart_noise2 = TwoDOF2(:,3);
beam_noise2 = TwoDOF2(:,4);
```

```
time2 = table2array(time2);
force2 = table2array(force2);
cart_noise2 = table2array(cart_noise2);
beam_noise2 = table2array(beam_noise2);

time3 = TwoDOF3(:,1);
force3 = TwoDOF3(:,2);
cart_noise3 = TwoDOF3(:,3);
beam_noise3 = TwoDOF3(:,4);
time3 = table2array(time3);
force3 = table2array(force3);
cart_noise3 = table2array(cart_noise3);
beam_noise3 = table2array(beam_noise3);

time4 = TwoDOF4(:,1);
force4 = TwoDOF4(:,2);
cart_noise4 = TwoDOF4(:,3);
beam_noise4 = TwoDOF4(:,4);
time4 = table2array(time4);
force4 = table2array(force4);
cart_noise4 = table2array(cart_noise4);
beam_noise4 = table2array(beam_noise4);

time5 = TwoDOF5(:,1);
force5 = TwoDOF5(:,2);
cart_noise5 = TwoDOF5(:,3);
beam_noise5 = TwoDOF5(:,4);
time5 = table2array(time5);
force5 = table2array(force5);
cart_noise5 = table2array(cart_noise5);
beam_noise5 = table2array(beam_noise5);

%% POINT 1-2
h11 = tf([m_tot MeanC MeanK 0 0], ...
          [(m_beam*m_tot) ...
          (MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot) ...
          (m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot +
MeanK*m_tot) ...
          (2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC) ...
          (2*rod_stiffness*MeanK)]);

h21 = tf([0 MeanC MeanK 0 0], ...
          [(m_beam*m_tot) ...
          (MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot) ...
          (m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot +
MeanK*m_tot) ...
          (2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC) ...
          (2*rod_stiffness*MeanK)]);
%{
h12 = tf([0,MeanC,MeanK,0,0], ...
          [m_beam*m_tot ...
          (MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot) ...
          (m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot +
MeanK*m_tot) ...
          (2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC) ...
```

```
          2*rod_stiffness*MeanK]);

h22 = tf([m_beam,MeanC + 2*rod_damping_factor,MeanK + 2*rod_stiffness,0,0], ...
          [m_beam*m_tot ...
          (MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot) ...
          (m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot +↙
MeanK*m_tot) ...
          (2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC) ...
          2*rod_stiffness*MeanK]);
%}

%=====================Error Propagation========================
%h11
errC = 3*SigmaC;
errK = 3*SigmaK;
Num_h11 = m_tot+MeanC+MeanK;
Den_h11 = m_beam*m_tot + MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot ...
      +m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot ...
      +MeanK*m_tot + 2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC ...
      +2*rod_stiffness*MeanK;
eNh11 = sqrt(errC^2+errK^2);
eDh11 = sqrt(((m_beam+m_tot+2*rod_damping_factor+2*rod_stiffness)^2)*(errC^2) ...
               +((m_beam+m_tot+2*rod_damping_factor+2*rod_stiffness)^2)*(errK^2));
eh11 = (Num_h11/Den_h11)*sqrt((eNh11/(Num_h11))^2+(eDh11/(Den_h11))^2);

h11_Upper = h11+eh11;
h11_Lower = h11-eh11;

%h21
errC = 3*SigmaC;
errK = 3*SigmaK;
Num_h21 = MeanC+MeanK;
Den_h21 = m_beam*m_tot + MeanC*m_beam + MeanC*m_tot + 2*rod_damping_factor*m_tot ...
          +m_beam*MeanK + 2*rod_damping_factor*MeanC + 2*rod_stiffness*m_tot +↙
MeanK*m_tot ...
          +2*rod_damping_factor*MeanK + 2*rod_stiffness*MeanC ...
          +2*rod_stiffness*MeanK;
eNh21 = sqrt(errC^2+errK^2);
eDh21 = sqrt(((m_beam+m_tot+2*rod_damping_factor+2*rod_stiffness)^2)*(errC^2) ...
               +((m_beam+m_tot+2*rod_damping_factor+2*rod_stiffness)^2)*(errK^2));
eh21 = (Num_h21/Den_h21)*sqrt((eNh21/(Num_h21))^2+(eDh21/(Den_h21))^2);

h21_Upper = h21+eh21;
h21_Lower = h21-eh21;

%=======================Plots================================
opts = bodeoptions('cstprefs');
opts.PhaseVisible = 'off';
opts.MagUnits = 'abs';

figure(1)
hold on
bodeplot(h11,'k',h11_Upper,'r:',h11_Lower,'r--',{4*2*pi,31*2*pi},opts);
bodeplot(h21,'k',h21_Upper,'b:',h21_Lower,'b--',{4*2*pi,31*2*pi},opts);
```

```
grid on
title('Cart/Beam Acceleration vs Force Transfer Function Bode Plot')
legend('Beam Acc. vs Force','Beam Acc. vs Force Upper','Beam Acc. vs Force Lower', ...
       'Cart Acc. vs Force','Cart Acc. vs Force Upper','Cart Acc. vs Force Lower')
%hold off
saveas(gcf, 'Plots\6. Cart-Beam Acceleration vs Force Transfer Function Bode Plot.↙
png');

%% POINT 3
%========================Cart===================================
[Tf1c,Fr1c] = tfestimate(force1,cart_noise1,[],[],[],Fs);
[Tf2c,Fr2c] = tfestimate(force2,cart_noise2,[],[],[],Fs);
[Tf3c,Fr3c] = tfestimate(force3,cart_noise3,[],[],[],Fs);
[Tf4c,Fr4c] = tfestimate(force4,cart_noise4,[],[],[],Fs);
[Tf5c,Fr5c] = tfestimate(force5,cart_noise5,[],[],[],Fs);
Tfc = [Tf1c,Tf2c,Tf3c,Tf4c,Tf5c];

Tm1c = abs(Tf1c);
Tm2c = abs(Tf2c);
Tm3c = abs(Tf3c);
Tm4c = abs(Tf4c);
Tm5c = abs(Tf5c);

lim1 = find((interp1(4:31,Fr1c,'nearest') == 4),1,'first');
lim2 = find((interp1(4:31,Fr1c,'nearest') == 31),1,'last');

sys1c = frd(Tf1c(lim1:lim2),Fr1c(lim1:lim2).*(2*pi));
sys2c = frd(Tf2c(lim1:lim2),Fr2c(lim1:lim2).*(2*pi));
sys3c = frd(Tf3c(lim1:lim2),Fr3c(lim1:lim2).*(2*pi));
sys4c = frd(Tf4c(lim1:lim2),Fr4c(lim1:lim2).*(2*pi));
sys5c = frd(Tf5c(lim1:lim2),Fr5c(lim1:lim2).*(2*pi));

%==========================Beam================================
[Tf1b,Fr1b] = tfestimate(force1,beam_noise1,[],[],[],Fs);
[Tf2b,Fr2b] = tfestimate(force2,beam_noise2,[],[],[],Fs);
[Tf3b,Fr3b] = tfestimate(force3,beam_noise3,[],[],[],Fs);
[Tf4b,Fr4b] = tfestimate(force4,beam_noise4,[],[],[],Fs);
[Tf5b,Fr5b] = tfestimate(force5,beam_noise5,[],[],[],Fs);
Tfb = [Tf1b,Tf2b,Tf3b,Tf4b,Tf5b];

Tm1b = abs(Tf1b);
Tm2b = abs(Tf2b);
Tm3b = abs(Tf3b);
Tm4b = abs(Tf4b);
Tm5b = abs(Tf5b);

lim3 = find((interp1(4:31,Fr1b,'nearest') == 4),1,'first');
lim4 = find((interp1(4:31,Fr1b,'nearest') == 31),1,'last');

sys1b = frd(Tf1b(lim3:lim4),Fr1b(lim3:lim4).*(2*pi));
sys2b = frd(Tf2b(lim3:lim4),Fr2b(lim3:lim4).*(2*pi));
sys3b = frd(Tf3b(lim3:lim4),Fr3b(lim3:lim4).*(2*pi));
sys4b = frd(Tf4b(lim3:lim4),Fr4b(lim3:lim4).*(2*pi));
sys5b = frd(Tf5b(lim3:lim4),Fr5b(lim3:lim4).*(2*pi));
```

```
%=======================Plots=================================
figure(2)
hold on
bode(sys1b,{31,190},opts);
bode(sys1c,{31,190},opts);
grid on
title('1st Test Bode Plot');
legend('Beam Acc. vs Force','Cart Acc. vs Force');
hold off
saveas(gcf, 'Plots\7. 1st Test Bode Plot.png');

figure(3)
hold on
bode(sys2b,{31,190},opts);
bode(sys2c,{31,190},opts);
grid on
title('2nd Test Bode Plot');
legend('Beam Acc. vs Force','Cart Acc. vs Force');
hold off
saveas(gcf, 'Plots\8. 2nd Test Bode Plot.png');

figure(4)
hold on
bode(sys3b,{31,190},opts);
bode(sys3c,{31,190},opts);
grid on
title('3rd Test Bode Plot');
legend('Beam Acc. vs Force','Cart Acc. vs Force');
hold off
saveas(gcf, 'Plots\9. 3rd Test Bode Plot.png');

figure(5)
hold on
bode(sys4b,{31,190},opts);
bode(sys4c,{31,190},opts);
grid on
title('4th Test Bode Plot');
legend('Beam Acc. vs Force','Cart Acc. vs Force');
hold off
saveas(gcf, 'Plots\10. 4th Test Bode Plot.png');

figure(6)
hold on
bode(sys5b,{31,190},opts);
bode(sys5c,{31,190},opts);
grid on
title('5th Test Bode Plot');
legend('Beam Acc. vs Force','Cart Acc. vs Force');
hold off
saveas(gcf, 'Plots\11. 5th Test Bode Plot.png');

%% POINT 4
%======================Cart===================================
MeanTfc = mean(Tfc,2);
SigmaTfc_R = std(real(Tfc),0,2);
```

```
SigmaTfc_I = std(imag(Tfc),0,2);
SigmaTfc = complex(SigmaTfc_R,SigmaTfc_I);
Upper_Tfc = MeanTfc+3*SigmaTfc;
Lower_Tfc = MeanTfc-3*SigmaTfc;

sys_meanC = frd(MeanTfc(lim1:lim2),Fr1c(lim1:lim2).*(2*pi));
sys_upperC = frd(Upper_Tfc(lim1:lim2),Fr1c(lim1:lim2).*(2*pi));
sys_lowerC = frd(Lower_Tfc(lim1:lim2),Fr1c(lim1:lim2).*(2*pi));

%========================Beam==================================
MeanTfb = mean(Tfb,2);
SigmaTfb_R = std(real(Tfb),0,2);
SigmaTfb_I = std(imag(Tfb),0,2);
SigmaTfb = complex(SigmaTfb_R,SigmaTfb_I);
Upper_Tfb = MeanTfb+3*SigmaTfb;
Lower_Tfb = MeanTfb-3*SigmaTfb;

sys_meanB = frd(MeanTfb(lim3:lim4),Fr1b(lim3:lim4).*(2*pi));
sys_upperB = frd(Upper_Tfb(lim3:lim4),Fr1b(lim3:lim4).*(2*pi));
sys_lowerB = frd(Lower_Tfb(lim3:lim4),Fr1b(lim3:lim4).*(2*pi));

%========================Plots================================
figure(7)
hold on
bodeplot(sys_meanB,'r',sys_upperB,'r:',sys_lowerB,'r--',{31,190},opts);
bodeplot(sys_meanC,'b',sys_upperC,'b:',sys_lowerC,'b--',{31,190},opts);
grid on
title('Mean Experimental Transfer Function Bode Plots');
legend('Beam', ...
       'Beam Upper', ...
       'Beam Lower', ...
       'Cart', ...
       'Cart Upper', ...
       'Cart Lower');
hold off
saveas(gcf, 'Plots\12. Mean Experimental Transfer Function Bode Plots.png');

%% POINT 5
figure(8)
hold on
bodeplot(h11,'b',h11_Upper,'b:',h11_Lower,'b--',{30,190},opts);
bodeplot(sys_meanB,'r',sys_upperB,'r:',sys_lowerB,'r--',{30,190},opts);
grid on
title('Analytical and experimental Beam Acc. vs Force Bode Plots');
legend('Analytical', ...
       'Analytical Upper', ...
       'Analytical Lower', ...
       'Experimental', ...
       'Experimental Upper', ...
       'Experimental Lower');
hold off
saveas(gcf, 'Plots\13. Analytical and experimental Beam Acc. vs Force Bode Plots.↙
png');

figure(9)
```

```
hold on
bodeplot(h21,'b',h21_Upper,'b:',h21_Lower,'b--',{30,190},opts);
bodeplot(sys_meanC,'r',sys_upperC,'r:',sys_lowerC,'r--',{30,190},opts);
grid on
title('Analytical and experimental Cart Acc. vs Force Bode Plots');
legend('Analytical', ...
       'Analytical Upper', ...
       'Analytical Lower', ...
       'Experimental', ...
       'Experimental Upper', ...
       'Experimental Lower');
hold off
saveas(gcf, 'Plots\14. Analytical and experimental Cart Acc. vs Force Bode Plots.↙
png');

%% POINT 6
s11 = sqrt(-1)*Fr1b(lim3:lim4).*(2*pi);
s21 = sqrt(-1)*Fr1c(lim1:lim2).*(2*pi);

par_h11 = @(K,C,rod_K,rod_C) ...
             (m_tot.*s11.^4 + C.*s11.^3 + K.*s11.^2)./ ...
             ((m_beam*m_tot).*s11.^4+ ...
             (C*m_beam + C*m_tot + 2*rod_C*m_tot).*s11.^3+ ...
             (m_beam*K + 2*rod_C*C + 2*rod_K*m_tot + K*m_tot).*s11.^2+ ...
             (2*rod_C*K + 2*rod_K*C).*s11+ ...
            2*rod_K*K);
par_h21 = @(K,C,rod_K,rod_C) ...
             (C.*s21.^3 + K.*s21.^2)./ ...
             ((m_beam*m_tot).*s21.^4+ ...
             (C*m_beam + C*m_tot + 2*rod_C*m_tot).*s21.^3+ ...
             (m_beam*K + 2*rod_C*C + 2*rod_K*m_tot + K*m_tot).*s21.^2+ ...
             (2*rod_C*K + 2*rod_K*C).*s21+ ...
            2*rod_K*K);

mod_par_h11 = abs(par_h11(2000,1,2e-05,5e-10));
mod_par_h21 = abs(par_h21(2000,1,2e-05,5e-10));

MeanTmC = abs(MeanTfc(lim1:lim2).*(2*pi));
MeanTmB = abs(MeanTfb(lim3:lim4).*(2*pi));

err_h11 = @(x) ...
            rms(MeanTmB-abs(par_h11(x(1),x(2),x(3),x(4))));
err_h21 = @(x) ...
            rms(MeanTmC-abs(par_h21(x(1),x(2),x(3),x(4))));
K0 = MeanK;
C0 = MeanC;
rod_K0 = rod_stiffness;
rod_C0 = rod_damping_factor;
x0 = [K0,C0,rod_K0,rod_C0];

x_opt_h11 = fminsearch(err_h11,x0);
x_opt_h21 = fminsearch(err_h21,x0);

%=======================Plots================================
figure(10)
```

```
hold on
plot(Fr1c(lim1:lim2).*(2*pi),MeanTmB)
plot(Fr1c(lim1:lim2).*(2*pi),abs(par_h11(x0(1),x0(2),x0(3),x0(4))))
plot(Fr1c(lim1:lim2).*(2*pi),abs(par_h11(x_opt_h11(1),x_opt_h11(2),x_opt_h11(3),↙
x_opt_h11(4))))
grid on
xlabel('Frequency (rad/s)')
ylabel('Module Of Transfer Function')
title('Beam Transfer Function Fitting')
legend('Experimental TF', ...
       'Analytical TF at Initial Guess', ...
       'Analytical TF at Optimal Solution')
hold off
saveas(gcf, 'Plots\15. Beam Transfer Function Fitting.png');

figure(11)
hold on
plot(Fr1c(lim1:lim2).*(2*pi),MeanTmC)
plot(Fr1c(lim1:lim2).*(2*pi),abs(par_h21(x0(1),x0(2),x0(3),x0(4))))
plot(Fr1c(lim1:lim2).*(2*pi),abs(par_h21(x_opt_h21(1),x_opt_h21(2),x_opt_h21(3),↙
x_opt_h21(4))))
grid on
xlabel('Frequency (rad/s)')
ylabel('Module Of The Transfer Function')
title('Cart Transfer Function Fitting')
legend('Experimental TF', ...
       'Analytical TF at Initial Guess', ...
       'Analytical TF at Optimal Solution')
hold off
saveas(gcf, 'Plots\16. Cart Transfer Function Fitting.png');
```