MAE 315: Lab 3 Report

By: Teagan Kilian

Due: 11/13/2022

Abstract

Vibrations occur when the internal equilibrium of an object is disturbed. Many factors contribute to the characteristics of a specific oscillation and all of these factors are incredibly important to engineers. Oscillating beams are looked at as spring-mass systems where the beam and its material properties determine the stiffness of its internal "spring." Geometry of a system has a large effect on the vibrating behavior of a beam. Specifically, as length increases, the natural frequency of the beam decreases. Similarly, increasing the end-mass of a cantilever beam causes its natural frequency to decrease. The damping coefficient of a system has no effect on the natural frequency of a beam since it does not change any of its material properties; however, increasing the damping coefficient causes a decrease in the damped frequency for that beam. Since material properties had a large effect on how beams react in oscillation, it is clear that different materials will act differently under the same vibrating conditions. Through experimentation, it was found that both density and material spring constant are large contributors to the mass-spring system for beams of various materials.

Table of Contents

Background

As an engineer, it is crucial to be hyper-aware of vibration and the effects they have on different materials and structures. There are a plethora of man made and natural occurrences that generate vibrations and they can either be seen as an enemy or they can be used to one's advantage. Specifically, cantilever beams are one type of geometry that is seen in many different applications.

This lab allowed for a deeper look into the attributes affecting natural frequency of a system and specifically what type of change occurred. In practice, both beam theory and the mass-spring system equation of motion are utilized to analyze reactions to forced frequency which requires knowledge of the geometry of the beam and fundamental material properties.

$$my'' + cy' + ky = F(t)$$

The above ordinary, second order differential equation is used to calculate the theoretical maximum displacement of the free end of a beam where m = mass of beam, c = damping coefficient, k = spring constant, and F(t) = an arbitrary forcing function. All coefficient variables to the above differential equation are properties of the material; however, manipulation of each will cause a unique change in how the system reacts.

Procedure

This lab was composed of three different activities including the shaker experiment, the hammer experiment, and various computer-generated simulations. Both experiments utilized a solid carbon steel bar and LabVIEW computer software to collect data. As directed by the lab handout, the bar was cantilevered into a shaker with one degree of freedom. Accelerometers were attached to the bar both on the free end and the clamped end. A measuring tape was used to validate that the bar was the correct length before clamping it to the shaker.

Initially, all dimensions of the bar were measured and the accelerometer was turned on. The shaker functions with the use of an input function generator which was also turned on. During this portion of the experiment, the function generator was set to sweep mode. As the frequency values changed, the characteristics of the shaking bar were observed. The frequency where the highest amplitude was observed was noted. Next, the noted frequency was taken to be the middle value in a range of five discrete frequencies that were produced by the function generator. Before collecting data at each frequency value, the output was changed to sine wave and the input was changed to the first frequency value. The values of frequency are now held constant in order to collect data about the behavior of the beam under these conditions. This procedure was repeated for each of the four other chosen frequencies. Additionally, this entire procedure was repeated for each beam length including 16 inch, 20 inch, and 24 inch.

The next portion of the experiment was the hammer test. During this test, the beam was transferred from the clamp located on the shaker to a clamp located on a stationary mount. A tape measure was again used to measure the length of the beam so the procedure could be repeated for a 16 inch, 20 inch, and 24 inch beam. Once the setup was configured, the LabView software was changed from “Save Peak Data” and “Continue Acquisition” to “Save Spectra and

Time Series Data." Then, LabVIEW was ready to collect data. LabVIew was toggled into data acquisition mode and an impulse was quickly delivered to the very end of the bar using a hammer. The best data will result from the hammer being in contact with the bar for the smallest amount of time.

The purpose of the simulation portion of the experiment was to test the effect of different attributes on the natural and damped frequency of a beam. The simulations were produced by a vi labeled 'VISBSIMv9" which allowed the definition of beam length, width, thickness, end-weight, damping coefficient, and material. While width and thickness were held constant throughout the whole procedure, every other variable had a chance to be varied while the remaining variables were held constant. The default values were length = 20 inches, end weight = 0kg, damping coefficient = 0.01, and material type was carbon steel.

All of the data collected by the LabView software through the accelerometers and the vi software were saved as both fft and dat files. These files were saved in a .txt format which were imported into MATLAB in order to complete the lab analysis.

Results

One major finding of this experiment was the relationship between beam length and frequency. As seen in Figure 1, there is a very strong negative correlation between the two variables. This trend can be seen for both damped and natural frequencies. Another finding which can be observed in the same figure is the fact that damped frequency will always be a smaller value than natural frequency. This makes physical sense since damping limits the amplitude of a signal and changes the period of oscillations. Further, the difference between damped and natural frequencies decreases as beam length increases leading to the assumption that longer beams are more stable in their oscillations and are consequently better for applications in which a more stable reaction to vibrations is favorable.

Although the purpose of this experiment was to view the natural frequency of this beam, inevitably, damping occurred. Because this set up was done in a lab setting, the system is not perfect and many factors including energy loss due to work against gravity and drag caused the actual natural frequency of the beam to not be observed (Table 1). In a perfect situation, where the natural frequency is applied to a cantilever beam, resonance will occur and theoretically, the oscillations will be able to continue at the same amplitude forever.

Table 1 (Shaker Experiment)

Length of Beam (inches)	Theoretical Natural Frequency (rad/s)	Damped Frequency (rad/s)	Experimental Natural Frequency (Hz)	Damping Ratio	Spring Constant (KN/m)
16 +/- 1/64	202.8347 +/- 12.6834	172.7876 +/- 3.1416	32.2821 +/- .5	0.5238 +/- 0	4.99 +/- .939
20 +/- 1/64	129.8142 +/- 8.1159	116.2389 +/- 3.1416	20.6606 +/- .5	0.4452 +/- 0	2.5549 +/- .4807
24 +/- 1/64	90.1487 +/- 5.6355	84.8232 +/- 3.1416	14.3476 +/- .5	0.3386 +/- 0	1.4785 +/- .2782

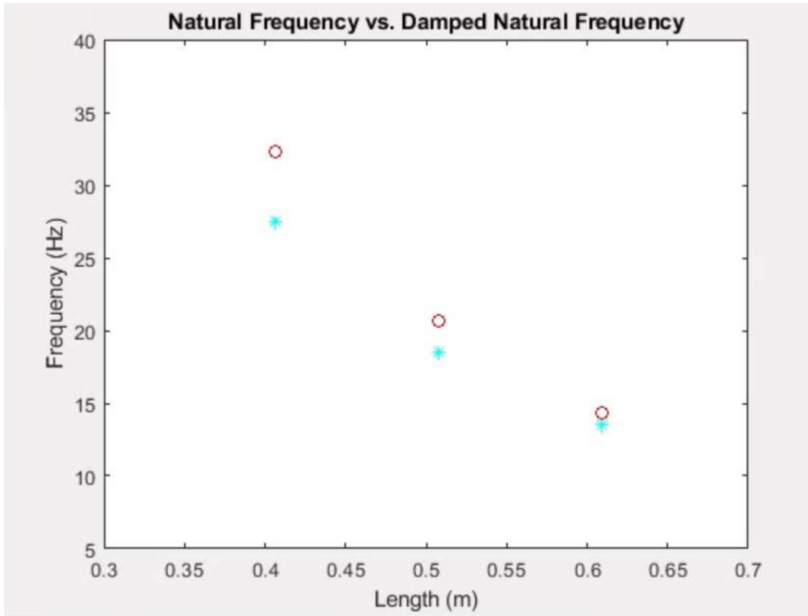


Figure 1: "O" denotes ω_n and "*" denotes ω_d

As concluded during the simulation, the natural frequency of a system is not dictated by the damping coefficient. During simulation 3.2, the value of the damping coefficient was varied while every other variable was held constant. As seen in Figure 2, each peak on the graph occurs at the same frequency value depicting that 'c' does not affect the natural frequency of a system. As the damping coefficient increases, the response amplitude decreases due to the fact that the damping coefficient decreases the range of the oscillating motion. In contrast, the damped frequency of a system is affected by the damping coefficient. In fact, as the damping coefficient increases, the damped frequency decreases as seen in Table 2.

Table 2 (Simulation 3.3.2)

Material	Theoretical Frequency (Hz)	Simulated Frequency (Hz)	Theoretical Natural Frequency (rad/s)	Spring Constant (KN/m)
Stainless Steel	19.8969	19.3691	125.0159	2.3937
Carbon Steel	20.3885	20.3445	129.8142	2.5549
Aluminum	20.6606	20.6193	128.105	0.8558

In addition to the length of the beam, end-mass can also change the natural frequency of a system. Mass of weight added to the end of a beam increases, the natural frequency of the beam decreases as seen in Table 3. The addition of mass will cause a decrease in magnitude of response as it would take more energy to oscillate a large mass compared to a small one.

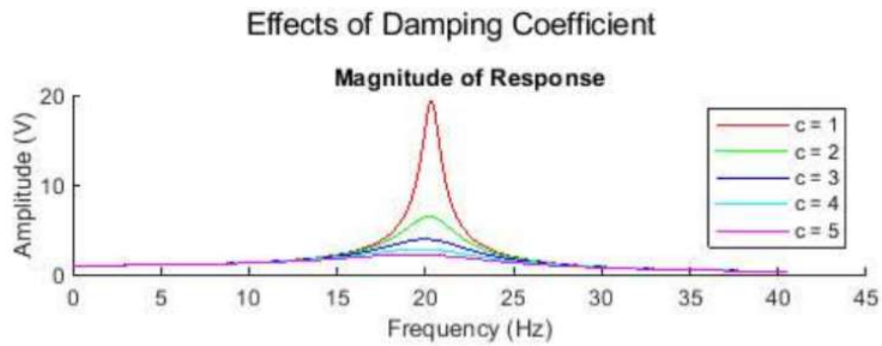


Figure 2: Amplitudes of systems with various damping coefficients

Table 3 (Simulation 3.3.1)

End Weight (kg)	Theoretical Frequency (Hz)	Simulated Natural Frequency (Hz)	Theoretical Natural Frequency (rad/s)
0.25	12.6941	12.5049	79.7597
0.4	10.8315	10.6707	68.0565
0.65	8.9851	8.85224	56.4552

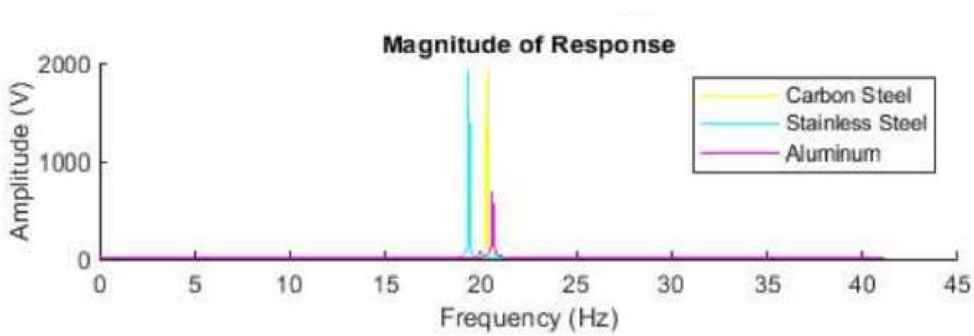


Figure 3: Amplitude vs Frequency for various materials

Thirdly, material type is a significant factor in the natural frequency of a beam - due largely to the fact that many variables are reliant on material properties. The material type of a beam can determine the spring constant and mass (due to material density) which will cause a difference in behavior for beams of the same geometry. Interestingly, carbon steel and aluminum have very similar natural frequencies although they have quite different calculated spring constants. Stainless steel and carbon steel have similar spring constants as well as reaching similar amplitudes (Figure 3). Based on this, it can be deduced that the spring constant of a material is a very large factor in the amplitude of oscillations while a beam is in resonance.

Another critical result found throughout this experiment was the fact that the natural frequency of a system does not change with variables the way that damped frequency does. In the case of a cantilevered beam, one set up with all of its properties has one natural frequency (and modes of that frequency). The system may be able to oscillate at a variety of frequencies depending on the input to the system, but only the natural frequency will produce the most consistent results. When a beam is oscillating at its natural frequency, it is known as resonance and this is when the vibration is most stable. As seen in Figure 4, the first - and largest peak - is considered the natural frequency, and each successive peak is a mode of the natural frequency.

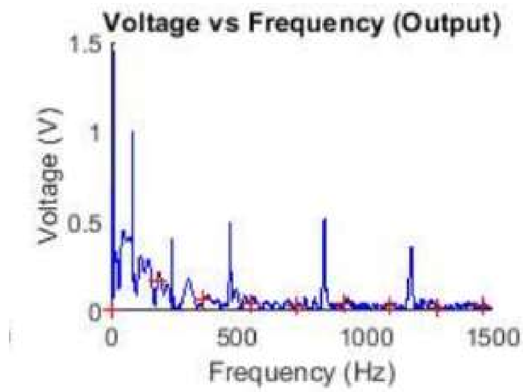


Figure 4: Hammer experiment results for 24 in. beam

Conclusion

Throughout the experiment and simulations, many factors were investigated to determine their effect on a vibrating system. Attributes including length, damping coefficient, material type, and end-mass were all varied and applied to beams of identical width and thickness. Experiments utilizing computer software were carefully executed to maintain accuracy and diminish human error. Computer simulations were conducted in order to further decrease error and allow the testing of variables which would not be possible in the lab setting available.

It was learned that within engineering design, it is crucial to consider how the system should react to vibration when choosing material type and geometry of a system. In a system created to function on Earth, it is almost inevitable that damping due to drag will occur and one must take this into account.

Another aspect of design which could be affected by vibrations is the cohesiveness of natural frequencies between different parts of a structure. There may be certain situations where it is favorable to have parts of a system oscillating out of phase with one another canceling out the movement, but there also may be times when parts having different natural frequencies could lead to large failures.

More than anything, it is important for engineers to understand vibrations and use them to their advantage.

Appendix

Equations

Differential equation representing the motion of a spring-mass system

$$my'' + cy' + ky = F(t)$$

Damping ratio equation

$$\zeta = \frac{c}{2\, m\, w_n}$$

Damping ratio estimation for $\zeta << 1$

$$\zeta = \frac{1}{2\pi} \ln\left(\frac{Y1}{Y2}\right)$$

Theoretical natural frequency equation (rad/s)

$$\omega_n\left(\frac{rad}{s}\right) = \sqrt{\frac{k}{m}}$$

Conversion from radians per second to Hertz

$$f_n(Hertz) = \frac{\omega_n}{2\pi}$$

Tables

Shaker Experiment

Length of Beam (inches)	Theoretical Natural Frequency (rad/s)	Damped Frequency (rad/s)	Experimental Natural Frequency (Hz)	Damping Ratio	Spring Constant (KN/m)
16 +/- 1/64	202.8347 +/- 12.6834	172.7876 +/- 3.1416	32.2821 +/- .5	0.5238 +/- 0	4.99 +/- .939
20 +/- 1/64	129.8142 +/- 8.1159	116.2389 +/- 3.1416	20.6606 +/- .5	0.4452 +/- 0	2.5549 +/- .4807
24 +/- 1/64	90.1487 +/- 5.6355	84.8232 +/- 3.1416	14.3476 +/- .5	0.3386 +/- 0	1.4785 +/- .2782

Hammer Experiment

Length of Beam (inches)	Theoretical Natural Frequency (rad/s)	Damped Frequency (rad/s)	Experimental Natural Frequency (Hz)	Damping Ratio	Spring Constant (KN/m)
16 +/- 1/64	202.8347 +/- 12.6834	172.7876 +/- 3.1416	0.3662 +/- .5	0.106 +/- .2205E-7	4.99 +/- .939
20 +/- 1/64	129.8142 +/- 8.1159	116.2389 +/- 3.1416	18.6768 +/- .5	0.0911 +/- .9032E-7	2.5549 +/- .4807
24 +/- 1/64	90.1487 +/- 5.6355	84.823 +/- 3.1416	13.5498 +/- .5	0.0585 +/- .2879E-7	1.4785 +/- /.2782

Simulation 3.1

Length (inches)	Frequency (Hz) Simulated	Theoretical Natural Frequency (Hz)	Damping Ratio
16	31.7883	32.2821	0.0013
20	20.3445	20.6606	0.0016
24	12.0382	12.2252	0.0021

Simulation 3.2

Damping Coefficient	Theoretical Natural Frequency (rad/s)	Simulated Natural Frequency (Hz)	Damping Ratio	Damped Frequency (Hz)
1	129.8142	20.3445	0.1596	20.3957 + 0.0000i
3	129.8142	20.3445	0.4789	18.1376 + 0.0000i
5	129.8142	20.3445	0.7981	12.4478 + 0.0000i
7	129.8142	20.3445	1.1174	0.0000 +10.2998i
9	129.8142	20.3445	1.4366	0.0000 +21.3104i

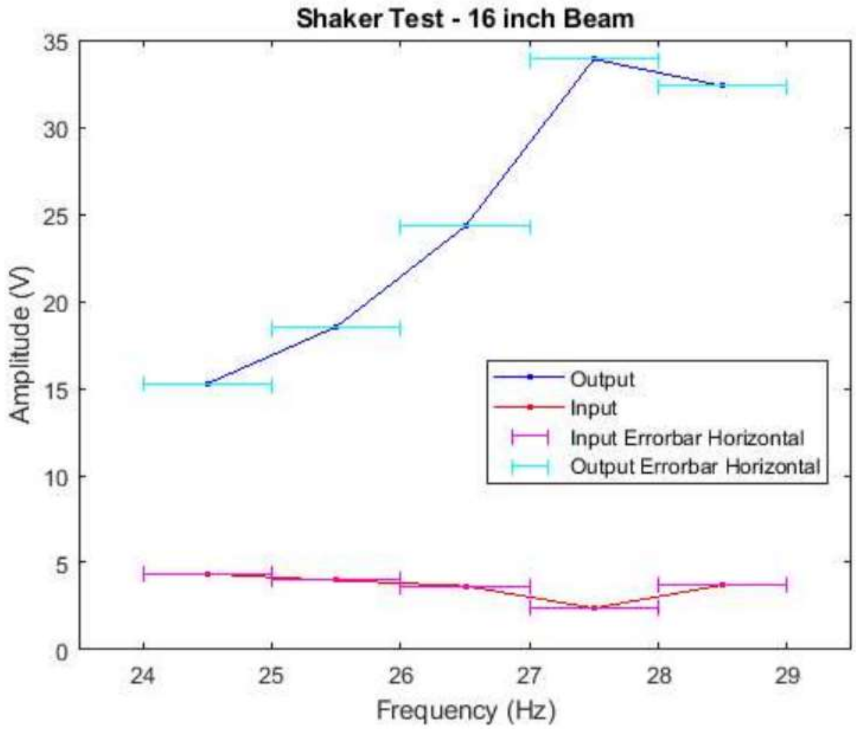
Simulation 3.3.1

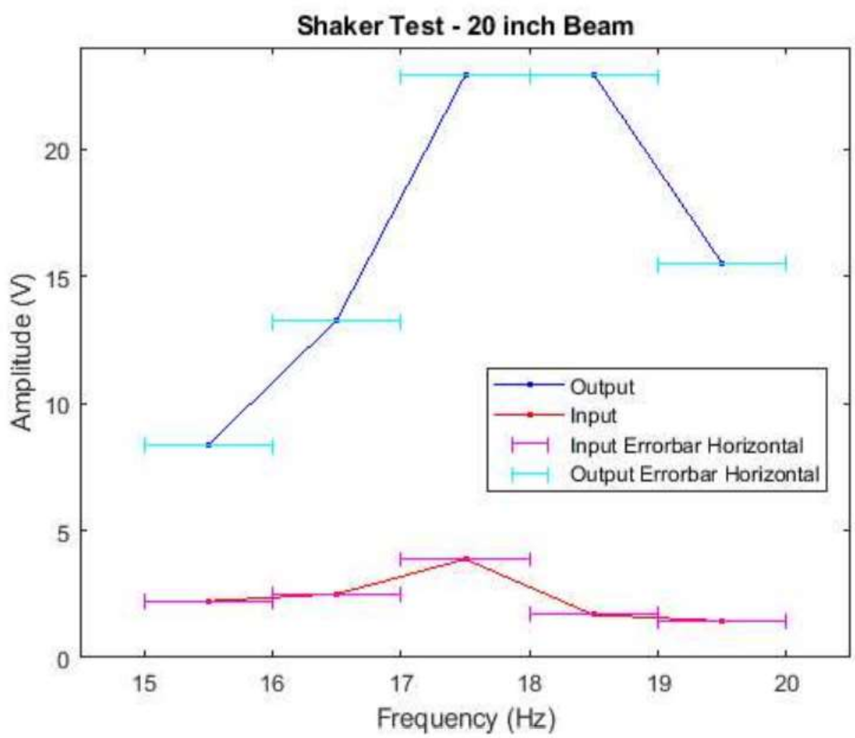
End Weight (kg)	Theoretical Frequency (Hz)	Simulated Natural Frequency (Hz)	Theoretical Natural Frequency (rad/s)
0.25	12.6941	12.5049	79.7597
0.4	10.8315	10.6707	68.0565
0.65	8.9851	8.85224	56.4552

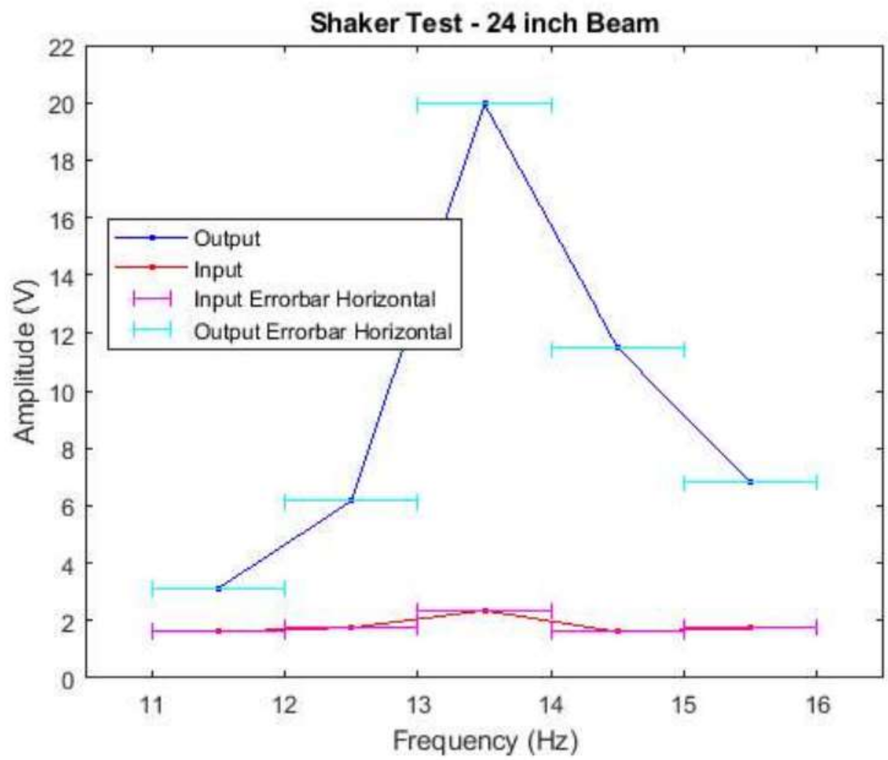
Simulation 3.3.2

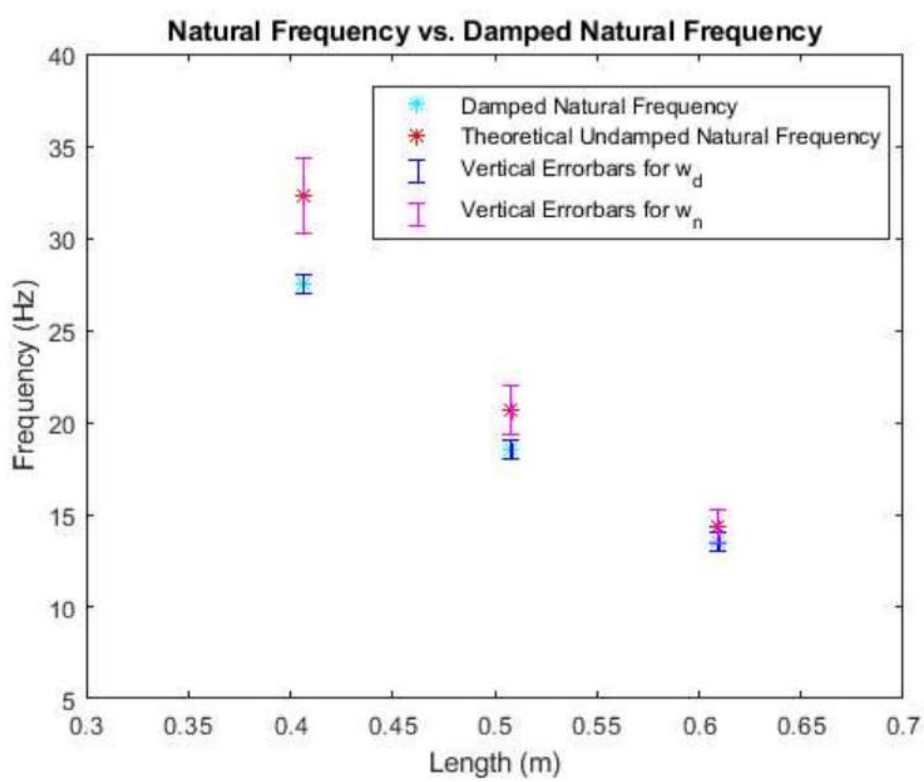
Material	Theoretical Frequency (Hz)	Simulated Frequency (Hz)	Theoretical Natural Frequency (rad/s)	Spring Constant (KN/m)
Stainless Steel	19.8969	19.3691	125.0159	2.3937
Carbon Steel	20.3885	20.3445	129.8142	2.5549
Aluminum	20.6606	20.6193	128.105	0.8558

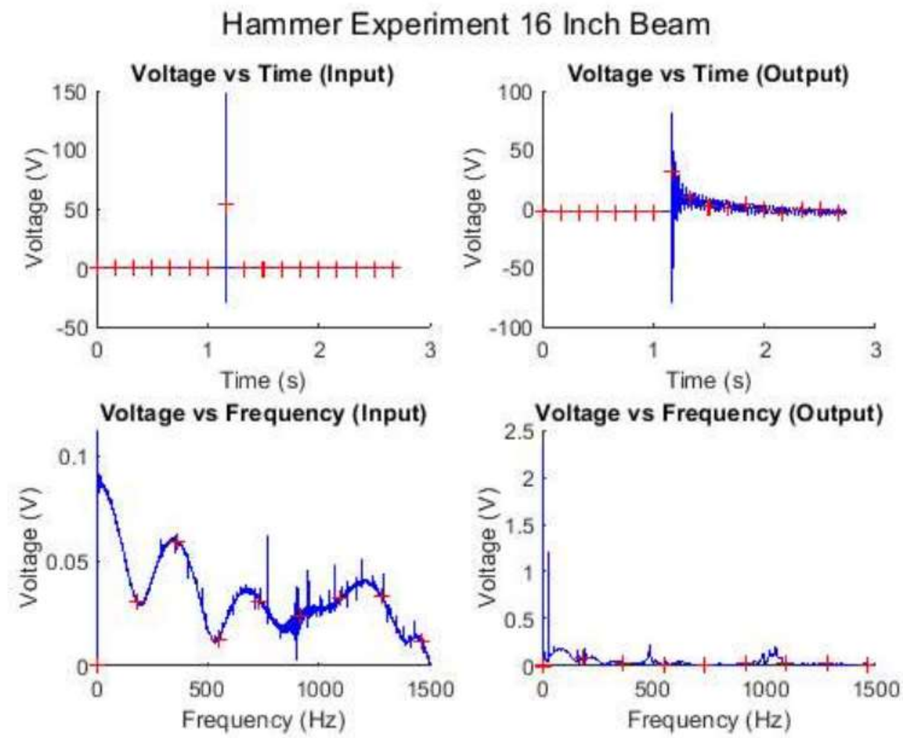
Plots

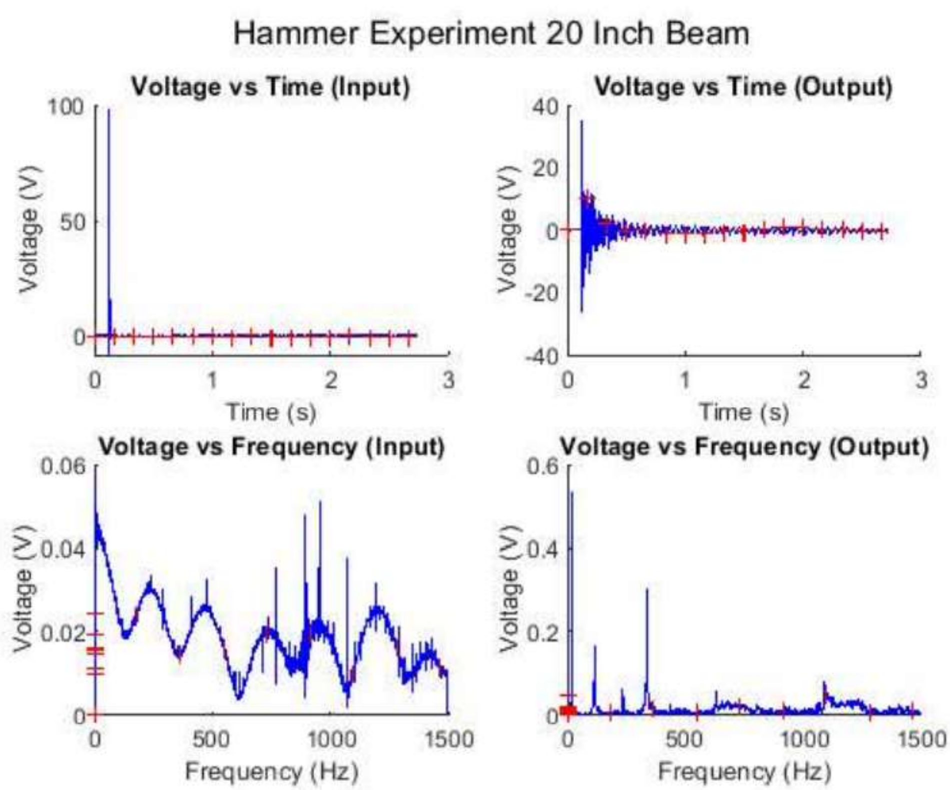

Shaker Test - 16 inch Beam
Amplitude (V)
Frequency (Hz)
Output
Input
Input Errorbar Horizontal
Output Errorbar Horizontal
0
5
10
15
20
25
30
35
24
25
26
27
28
29

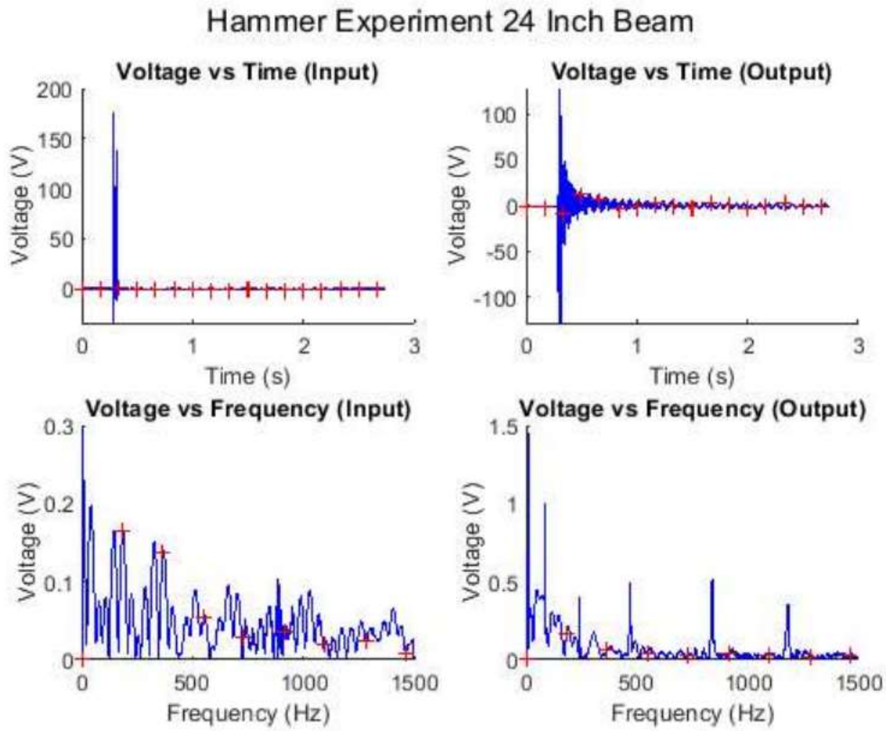

Shaker Test - 20 inch Beam
Amplitude (V)
Frequency (Hz)
Output
Input
Input Errorbar Horizontal
Output Errorbar Horizontal
0
5
10
15
20
15
16
17
18
19
20

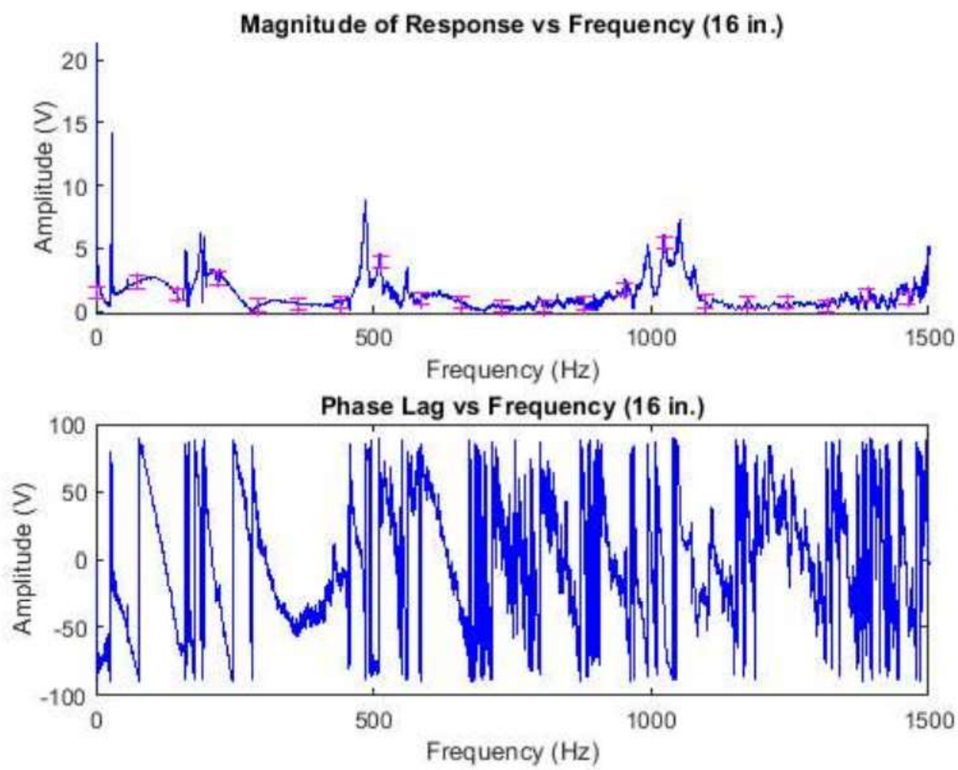

Shaker Test - 24 inch Beam
Amplitude (V)
Frequency (Hz)
Output
Input
Input Errorbar Horizontal
Output Errorbar Horizontal
0
2
4
6
8
10
12
14
16
18
20
22
11
12
13
14
15
16

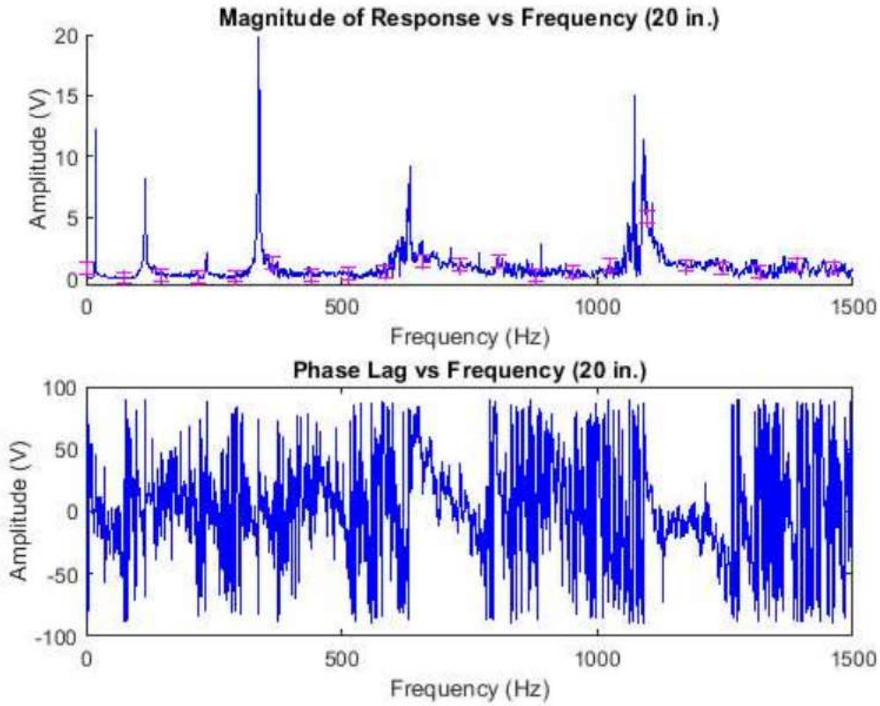

Natural Frequency vs. Damped Natural Frequency
Frequency (Hz)
Length (m)
Damped Natural Frequency
Theoretical Undamped Natural Frequency
Vertical Errorbars for w_d
Vertical Errorbars for w_n
5
10
15
20
25
30
35
40
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7

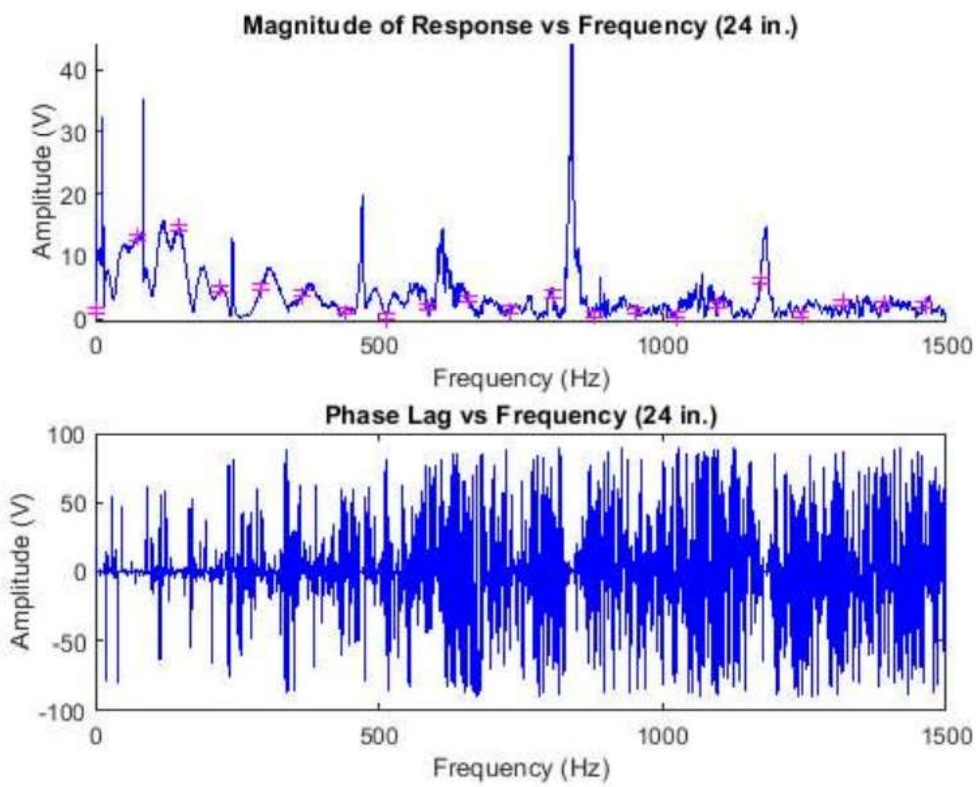

Hammer Experiment 16 Inch Beam
Voltage vs Time (Input)
Voltage vs Time (Output)
Voltage vs Frequency (Input)
Voltage vs Frequency (Output)
Voltage (V)
Time (s)
Frequency (Hz)

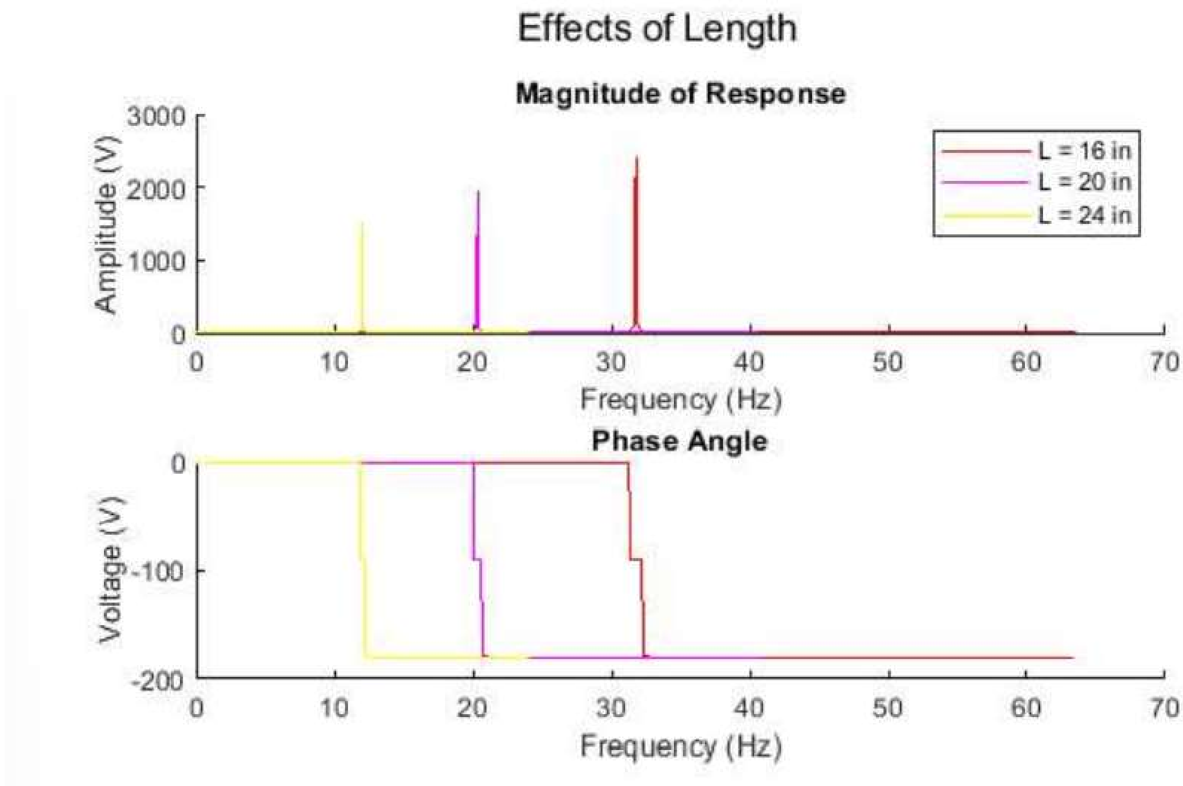

Hammer Experiment 20 Inch Beam
Voltage vs Time (Input)
Voltage vs Time (Output)
Voltage vs Frequency (Input)
Voltage vs Frequency (Output)
Voltage (V)
Time (s)
Frequency (Hz)

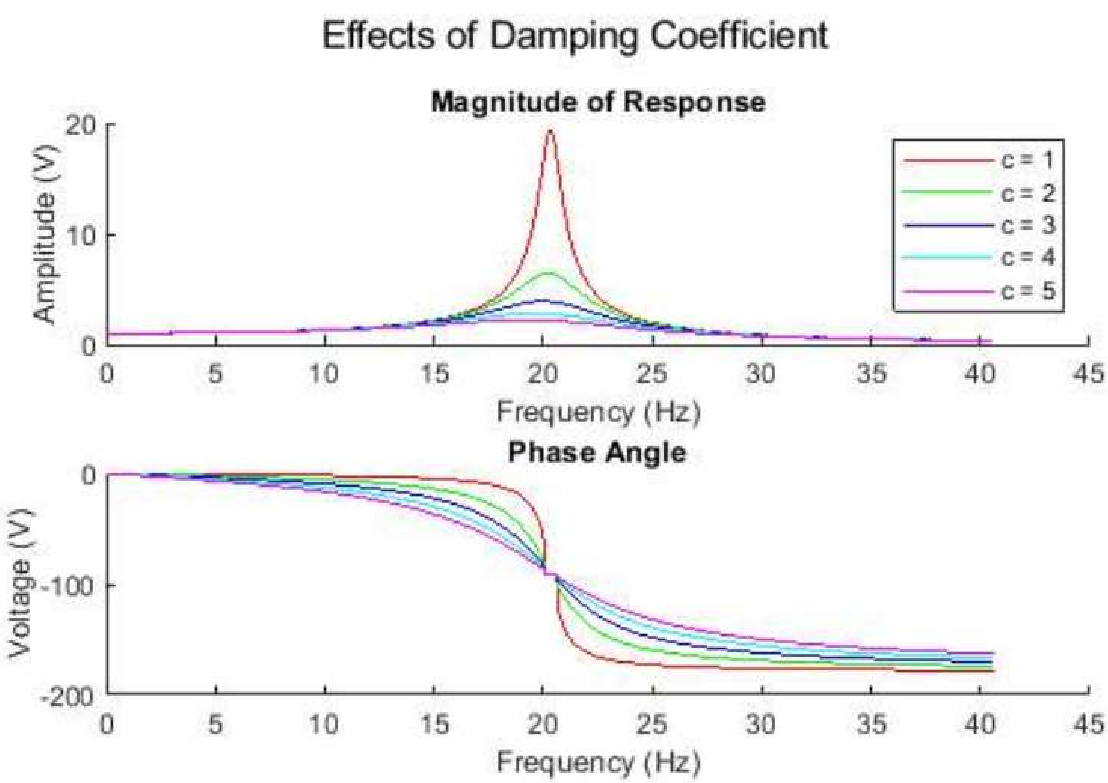

Hammer Experiment 24 Inch Beam
Voltage vs Time (Input)
Voltage vs Time (Output)
Voltage (V)
Time (s)
Voltage vs Frequency (Input)
Voltage vs Frequency (Output)
Frequency (Hz)

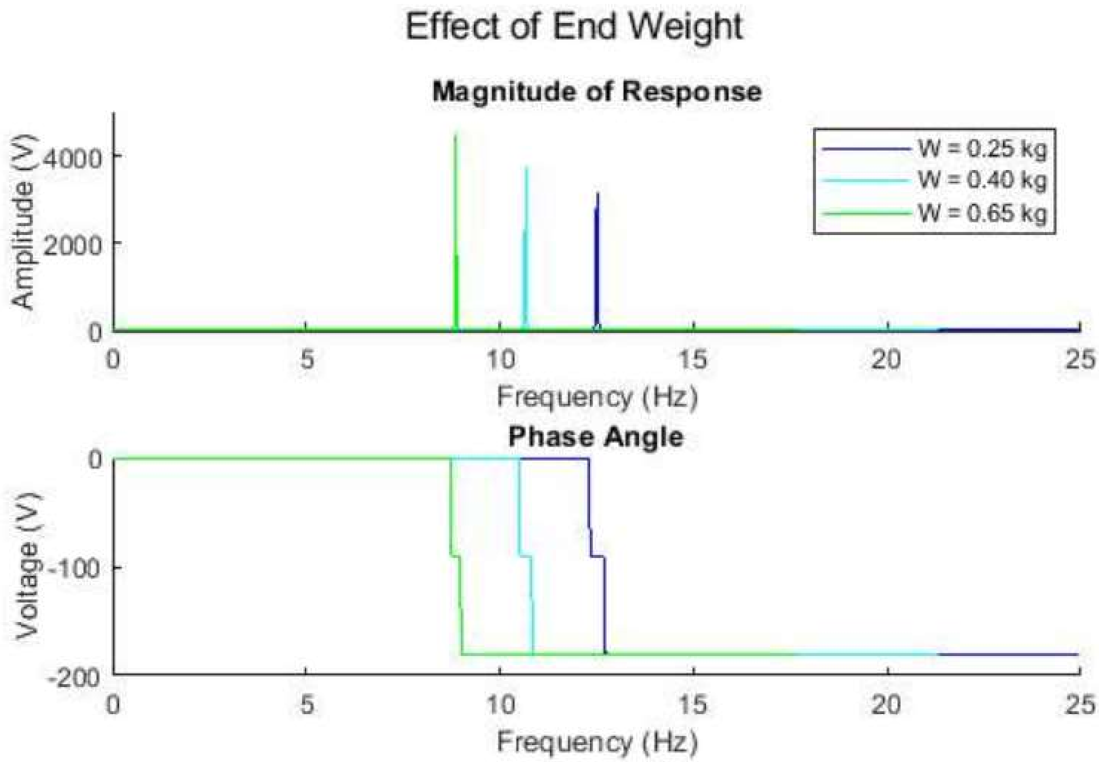

Magnitude of Response vs Frequency (16 in.)
Amplitude (V)
Frequency (Hz)
Phase Lag vs Frequency (16 in.)

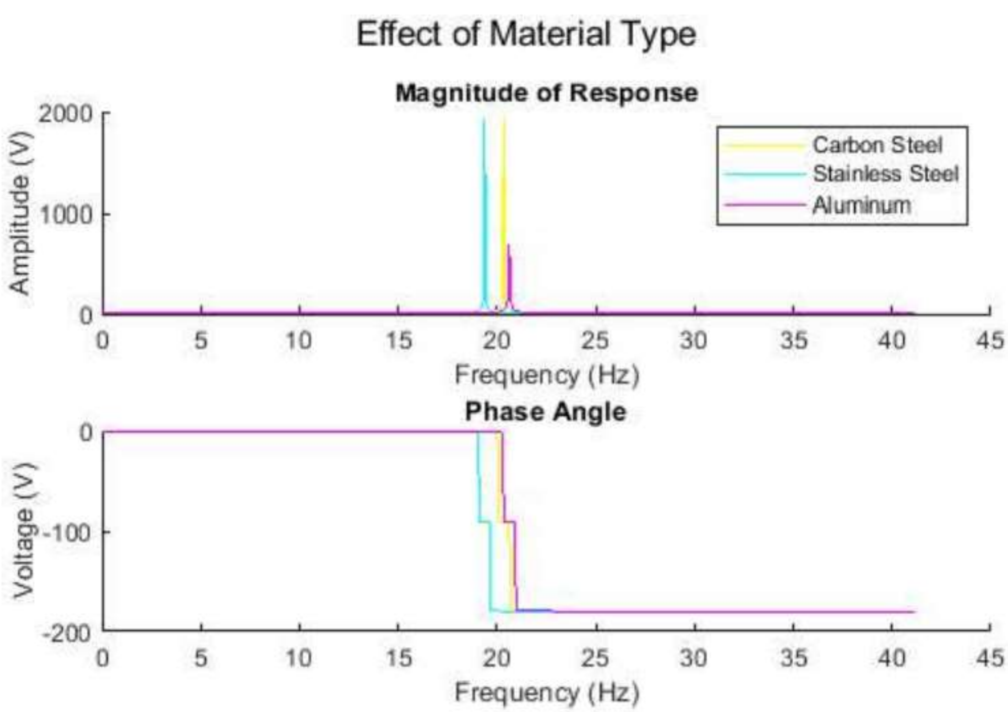

Magnitude of Response vs Frequency (20 in.)
Amplitude (V)
Frequency (Hz)
Phase Lag vs Frequency (20 in.)
Amplitude (V)
Frequency (Hz)


Magnitude of Response vs Frequency (24 in.)
Amplitude (V)
Frequency (Hz)
Phase Lag vs Frequency (24 in.)
Amplitude (V)
Frequency (Hz)


Effects of Length
Magnitude of Response
Amplitude (V)
L = 16 in
L = 20 in
L = 24 in
Frequency (Hz)
Phase Angle
Voltage (V)
Frequency (Hz)


Effects of Damping Coefficient
Magnitude of Response
Amplitude (V)
c = 1
c = 2
c = 3
c = 4
c = 5
Frequency (Hz)
Phase Angle
Voltage (V)
Frequency (Hz)


Effect of End Weight
Magnitude of Response
W = 0.25 kg
W = 0.40 kg
W = 0.65 kg
Amplitude (V)
4000
2000
0
0
5
10
15
20
25
Frequency (Hz)
Phase Angle
Voltage (V)
0
-100
-200
0
5
10
15
20
25
Frequency (Hz)


Effect of Material Type
Magnitude of Response
Carbon Steel
Stainless Steel
Aluminum
Amplitude (V)
2000
1000
0
0
5
10
15
20
25
30
35
40
45
Frequency (Hz)
Phase Angle
Voltage (V)
0
-100
-200
0
5
10
15
20
25
30
35
40
45
Frequency (Hz)

Contents


```
% Teagan Kilian
% MAE 315
% Lab 3
```

Shaker Test

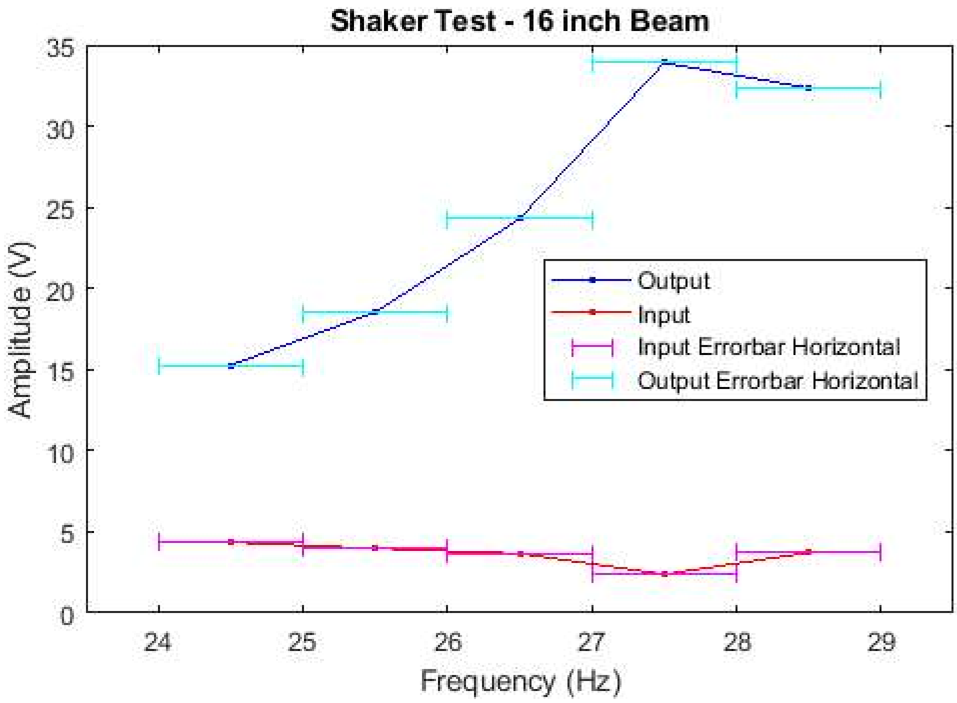
```
% Import data

% 16 inch beam
shaker16data    = importdata ('M002_B_Shaker_16.txt');
index_shaker16 = shaker16data(:,1);
freq_shaker16  = shaker16data(:,2);
out_shaker16   = shaker16data(:,3);
in_shaker16    = shaker16data(:,4);

% 20 inch beam
shaker20data    = importdata ('M002_B_Shaker_20.txt');
index_shaker20 = shaker20data(:,1);
freq_shaker20  = shaker20data(:,2);
out_shaker20   = shaker20data(:,3);
in_shaker20    = shaker20data(:,4);

% 24 inch beam
shaker24data    = importdata ('M002_B_Shaker_24.txt');
index_shaker24 = shaker24data(:,1);
freq_shaker24  = shaker24data(:,2);
out_shaker24   = shaker24data(:,3);
in_shaker24    = shaker24data(:,4);

% Plot Shaker Test Data

% 16 inch beam plot
figure (1)
p1      = plot(freq_shaker16, out_shaker16, 'b.-');
hold    on
p2      = plot(freq_shaker16, in_shaker16, '.r-' );
xlabel 'Frequency (Hz)'
ylabel 'Amplitude (V)'
title  'Shaker Test - 16 inch Beam'
xlim   ([23.5,29.5]);
ylim   ([0,35]);
[v_max16, index16] = max(out_shaker16);
w_d16   = freq_shaker16(index16); % damped natural frequency of 16in beam
```

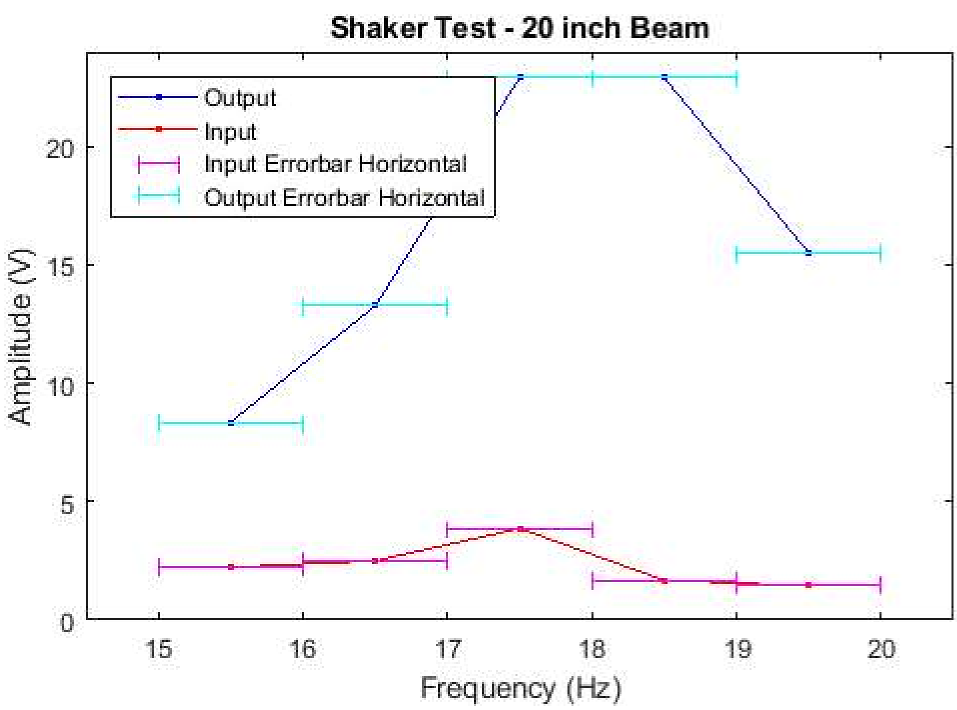
```
% 20 inch beam plot
figure  (2)
p3      = plot(freq_shaker20, out_shaker20, 'b.-');
hold    on
p4      = plot(freq_shaker20, in_shaker20, 'r.-' );
xlabel 'Frequency (Hz)'
ylabel 'Amplitude (V)'
title  'Shaker Test - 20 inch Beam'
xlim   ([14.5,20.5]);
ylim   ([0,24]);
legend ('Output', 'Input');
[v_max20, index20] = max(out_shaker20);
w_d20   = freq_shaker20(index20); % damped natural frequency of 20in beam


% 24 inch beam plot
figure (3)
p5     = plot(freq_shaker24, out_shaker24, 'b.-');
hold   on
p6     = plot(freq_shaker24, in_shaker24, 'r.-' );
xlabel 'Frequency (Hz)'
ylabel 'Amplitude (V)'
title  'Shaker Test - 24 inch Beam'
xlim   ([10.5,16.5]);
ylim   ([0,22]);
legend ('Output', 'Input');
[v_max24, index24] = max(out_shaker24);
w_d24  = freq_shaker24(index24); % damped natural frequency of 24in beam

w_dval = [w_d16, w_d20, w_d24];

% Symbolic Expressions

syms L b h res_tape res_f res_rho rho E bi_r n_bits c u_E y1 y2 samp w_d

% calculations
% general equations
vol    = L .* b .* h;                 % volume of beam
m_beam = rho .* vol;                  % mass of beam
m_eq   = ((33./140) .* m_beam);       % mass of entire system
I      = (1/12) .* b .* (h.^3);       % moment of inertia
k_eq   = (3 .* E .* I) ./ (L.^3);     % spring constant of the beam
w_nr   = sqrt(k_eq ./ m_eq);          % undamped natural frequency (rad/s)
w_n    = w_nr ./ (2 .* pi);           % undamped natural frequency (Hz)
zeta   = (1/(2*pi)).*log(y1./y2);     % damping ratio
zetas  = sqrt((-(w_d./w_n).^2)+1);
zeta_s = sqrt((-(w_dval./w_nval).^2)+1);

% Uncertainties

% partial derivatives
dIdL     = diff(I, L);
dIdb     = diff(I, b);
dIdh     = diff(I, h);
dw_ndE   = diff(w_n, E);
dw_ndb   = diff(w_n, b);
dw_ndh   = diff(w_n, h);
dw_ndL   = diff(w_n, L);
dw_ndrho = diff(w_n, rho);
dmdrho   = diff(m_eq, rho);
dmdL     = diff(m_eq, L);
```

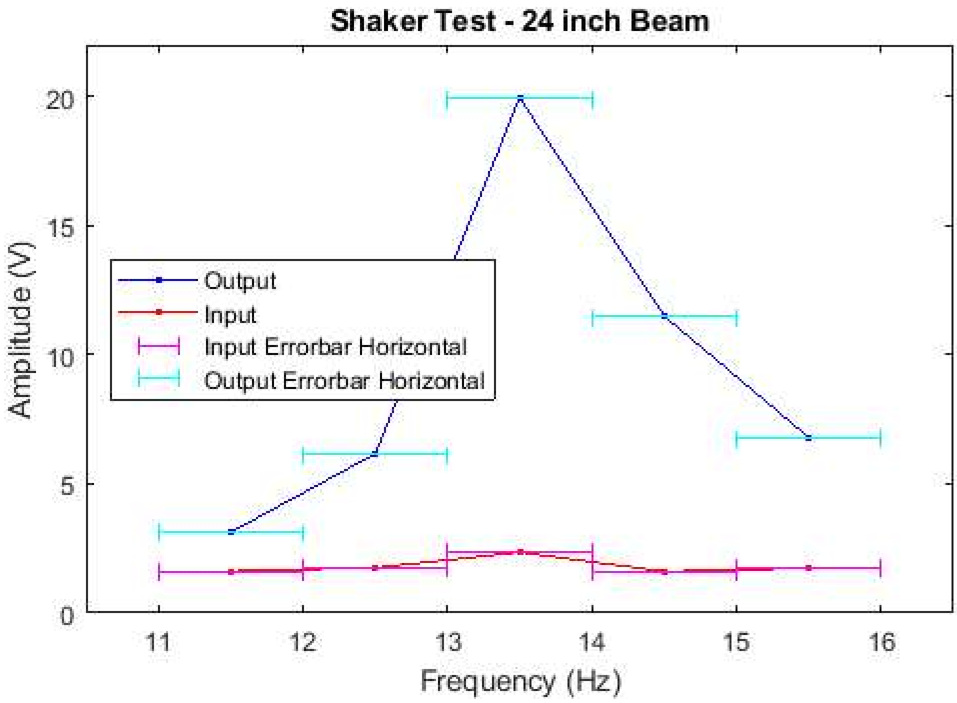
```
dmdh     = diff(m_eq, h);
dmdb     = diff(m_eq, b);
dzdE     = diff(zeta_s, E);
dzdb     = diff(zeta_s, b);
dzdL     = diff(zeta_s, L);
dzdrho   = diff(zeta_s, rho);
dzdh     = diff(zeta_s, h);
dzdw     = diff(zeta_s, w_d);
dkdb     = diff(k_eq, b);
dkdh     = diff(k_eq, h);
dkdE     = diff(k_eq, E);
dkdL     = diff(k_eq, L);

% Uncertainty equations
u_w   = .5 * res_f;                    % uncertainty in frequency
u_V   = .5 * bi_r ./ ((2.^n_bits) - 1); % uncertainty in voltage
u_L   = .5 * res_tape;                 % uncertainty in beam length
u_b   = .5 * res_tape;                 % uncertainty in beam width
u_h   = .5 * res_tape;                 % uncertainty in beam height
u_rho = .5 * res_rho;                  % uncertainty in density
u_m   = sqrt((dmdL.*u_L).^2 + (dmdh.*u_h).^2 + ...
  (dmdb.*u_b).^2 + (dmdrho.*u_rho).^2); % uncertainty in mass
u_I   = sqrt((dIdb.*u_b).^2 + (dIdh.*u_h).^2 + ...
  (dIdL.*u_L).^2);                     % uncertainty in moment of inertia
u_w_n = sqrt((dw_ndE.*u_E).^2 + (dw_ndb.*u_b).^2 + ...
  (dw_ndh.*u_h).^2 + (dw_ndL.*u_L).^2 + (dw_ndrho.*u_rho).^2);
                                       % uncertainty in natural frequency
u_zs  = sqrt((dzdE.*u_E).^2 + (dzdb.*u_b).^2 + ...
    (dzdL.*u_L).^2 + (dzdrho.*u_rho).^2 + (dzdh.*u_h).^2 + (dzdw.*u_w).^2);
                                       % uncertainty in damping ratio
u_k   = sqrt((dkdb.*u_b).^2 + (dkdh.*u_h).^2 + ...
    (dkdE.*u_E).^2 + (dkdL.*u_L).^2);
                                       % uncertainty in damping coeff

% Evaluate Symbolic Expressions

% assign values
E      = 2.06 .* 10 .^ 11;          % youngs modulus (Pa)
rho    = 7850;                      % densitiy (kg/m^3)
bi_r   = 5;                         % bipolar range (V)
n_bits = 24;                        % number of bits
L      = [16, 20, 24] .* .0254;     % lenghts of beam (meters)
b      = 1 .* .0254;                % width of beam (meters)
h      = 0.25 .* .0254;             % thickness of beam (meters)

% Evaluate Expressions
w_nval = double(subs(w_n));           % Hz
w_nrad = double(subs(w_nr));          % rad/s
k_eqs  = double(subs(k_eq));          % N/m
w_dval = [w_d16, w_d20, w_d24];       % Hz
w_drad = double(w_dval) .* 2 .* pi;   % rad/s


% evaluate uncertainty
res_tape  = 1/32 .* .0254;          % m
res_f     = 1;                      % Hz
res_rho   = .001;                   % kg/m^3
u_E       = .001;                  % Pa
u_w_ns    = double(subs(u_w_n));
u_w_ds    = double(subs(u_w));
u_Vs      = subs(u_V);
```

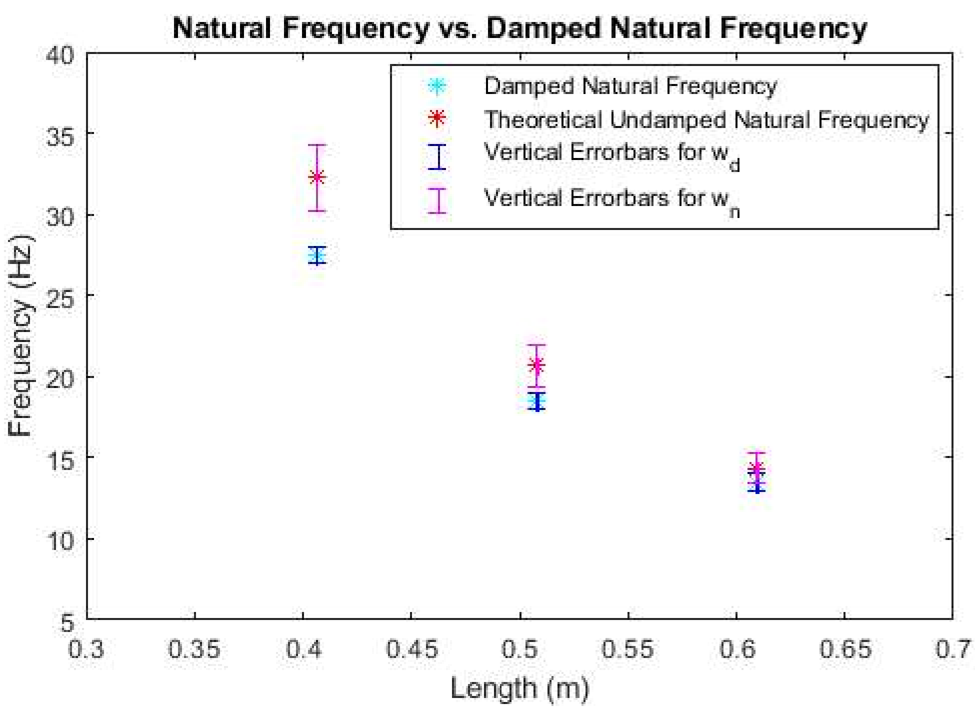
```
u_Ls      = subs(u_L);
u_zs      = double(subs(u_zs));
u_ks      = double(subs(u_k));

% w_d and w_n plots

figure  (4)
w1      = plot (L, w_dval, 'c*');
hold    on
w2      = plot (L, w_nval, 'r*');
xlabel  'Length (m)'
ylabel  'Frequency (Hz)'
title   'Natural Frequency vs. Damped Natural Frequency'
xlim    ([.3 .7]);
ylim    ([5 40]);

% Plot Error Bars

% error bars for 16 inch curve
figure (1)
err16V = u_Vs .* ones(size(in_shaker16));
err16w = u_w_ds .* ones(size(in_shaker16));

e1 = errorbar(freq_shaker16, in_shaker16, err16w, 'Horizontal', 'm',...
    "LineStyle", 'none');
e2 = errorbar(freq_shaker16, out_shaker16, err16w, 'Horizontal', 'c',...
    "LineStyle", 'none');
errorbar(freq_shaker16, in_shaker16, err16V, 'Vertical', 'm',...
    "LineStyle", 'none')
errorbar(freq_shaker16,out_shaker16, err16V, 'Vertical', 'c',...
    "LineStyle", 'none')
hold off

legend ([p1,p2,e1,e2],{'Output', 'Input','Input Errorbar Horizontal',...
    'Output Errorbar Horizontal'},'Location','best')


% error bars for 20 inch beam
figure (2)
err20V = u_Vs .* ones(size(in_shaker20));
err20w = u_w_ds .* ones(size(in_shaker20));

e3 = errorbar(freq_shaker20, in_shaker20, err20w, 'Horizontal', 'm',...
    "LineStyle", 'none');
e4 = errorbar(freq_shaker20, out_shaker20, err20w, 'Horizontal', 'c',...
    "LineStyle", 'none');
errorbar(freq_shaker20, in_shaker20, err20V, 'Vertical', 'm',...
    "LineStyle", 'none')
errorbar(freq_shaker20,out_shaker20, err20V, 'Vertical', 'c',...
    "LineStyle", 'none')
hold off

legend ([p3,p4,e3,e4],{'Output', 'Input','Input Errorbar Horizontal',...
    'Output Errorbar Horizontal'},'Location','best');

% error bars for 24 inch beam
figure (3)
err24V = u_Vs .* ones(size(in_shaker24));
err24w = u_w_ds .* ones(size(in_shaker24));

e5 = errorbar(freq_shaker24, in_shaker24, err24w, 'Horizontal', 'm',...
```

```
    "LineStyle", 'none');
e6 = errorbar(freq_shaker24, out_shaker24, err24w, 'Horizontal', 'c',...
    "LineStyle", 'none');
errorbar(freq_shaker24, in_shaker24, err24V, 'Vertical', 'm',...
    "LineStyle", 'none')
errorbar(freq_shaker24,out_shaker24, err24V, 'Vertical', 'c',...
    "LineStyle", 'none')
hold off

legend ([p5,p6,e5,e6],{'Output', 'Input','Input Errorbar Horizontal',...
    'Output Errorbar Horizontal'},'Location','best');

% error bars on wd vs wn graph

figure   (4)
err_L    = u_Ls .* ones(size(L));
err_wd   = u_w_ds .* ones(size(L));
e7       = errorbar(L,w_dval, err_wd, 'Vertical', 'b', "LineStyle",...
    'none');
errorbar (L, w_dval, err_L, 'Horizontal', 'b', "LineStyle", 'none')
err_wn   = u_w_ns .* ones(size(L));
e8       = errorbar(L, w_nval, err_wn, 'Vertical', 'm', "LineStyle",...
    'none');
errorbar (L,w_nval, err_L, 'Horizontal', 'm', "LineStyle", 'none')
hold      off

legend([w1,w2,e7,e8], {'Damped Natural Frequency',...
    'Theoretical Undamped Natural Frequency',...
    'Vertical Errorbars for w_d', 'Vertical Errorbars for w_n'},...
    'Location', 'best')
```


Shaker Test - 16 inch Beam
Amplitude (V)
Frequency (Hz)
Output
Input
Input Errorbar Horizontal
Output Errorbar Horizontal
0
5
10
15
20
25
30
35
24
25
26
27
28
29


Shaker Test - 20 inch Beam
Amplitude (V)
Frequency (Hz)
Output
Input
Input Errorbar Horizontal
Output Errorbar Horizontal
0
5
10
15
20
15
16
17
18
19
20





Hammer Test

```
% Import Data
% 16 inch beam
ham16traw = importdata ('10_24_2022m002_b_hammer_16fft.txt');
ham16fraw = importdata ('10_24_2022m002_b_hammer_16dat.txt');

ham16tdat = ham16traw.data;
ham16fdat = ham16fraw.data;

ham_16_t  = ham16tdat(:,1); % seconds
ham_16_it = ham16tdat(:,3); % V x 10^2
ham_16_ot = ham16tdat(:,2); % V x 10^2
```

```
ham_16_if = ham16fdat(:,3); % V x 10^2
ham_16_of = ham16fdat(:,2); % V x 10^2
ham_16_f  = ham16fdat(:,1); % Hz

% 20 inch beam
ham20traw = importdata ('10_24_2022m002_b_hammer_20fft.txt');
ham20fraw = importdata ('10_24_2022m002_b_hammer_20dat.txt');

ham20tdat = ham20traw.data;
ham20fdat = ham20fraw.data;

ham_20_t  = ham20tdat(:,1); % seconds
ham_20_it = ham20tdat(:,3); % V x 10^2
ham_20_ot = ham20tdat(:,2); % V x 10^2
ham_20_if = ham20fdat(:,3); % V x 10^2
ham_20_of = ham20fdat(:,2); % V x 10^2
ham_20_f  = ham20fdat(:,1); % Hz

% 24 inch beam
ham24traw = importdata ('10_24_2022m002_b_hammer_24fft.txt');
ham24fraw = importdata ('10_24_2022m002_b_hammer_24dat.txt');

ham24tdat = ham24traw.data;
ham24fdat = ham24fraw.data;

ham_24_t  = ham24tdat(:,1); % seconds
ham_24_it = ham24tdat(:,3); % V x 10^2
ham_24_ot = ham24tdat(:,2); % V x 10^2
ham_24_if = ham24fdat(:,3); % V x 10^2
ham_24_of = ham24fdat(:,2); % V x 10^2
ham_24_f  = ham24fdat(:,1); % Hz

% Define Variables and Equations

zetah = (1 ./ (2 .* pi)) .* log (y1 ./ y2);


% Calculate Uncertainty

u_t   = .5 .* (1 ./ samp);              % Uncertainty in time
u_V   = .5 * bi_r ./ ((2.^n_bits) - 1); % Uncertainty in amplitude
u_f   = .5 .* res_f;                    % Uncertainty in frequency

dzdy1 = diff(zetah, y1);
dzdy2 = diff(zetah, y2);

u_zh  = sqrt((dzdy1.*u_V).^2 + (dzdy2.*u_V).^2);
                                       % Uncertainty in damping coeff

% Assign Values
samp    = 3000;                        % Hz
bi_r    = 5;                           % V
n_bits = 24;
res_f   = 1;                           % Hz
y1      = [2.35409, 0.534513, 1.44697]; % V
y2      = [1.20922, 0.301462, 1.00206]; % V

% Evaluate Expressions

u_th       = subs(u_t);
u_Vh       = subs(u_V);
```

```matlab
u_w        = double(subs(u_w_n));
u_fh       = double(subs(u_f));
u_zh       = double(subs(u_zh));
u_kh       = double(subs(u_k));

% Plot Data

% error bar sizes
errorf   = u_f .* ones(size(ham_16_f));  % frequency error bar size
errorVf  = u_V .* ones(size(ham_16_f));  % amp error bar size from f data
errorVt  = u_V .* ones(size(ham_16_t));  % amp error bar size from t data
errort   = u_t .* ones(size(ham_16_t));  % time error bar size

% Hammer Test 16 inch Beam
figure  (5)
subplot (2,2,1)
hold    on
plot    (ham_16_t, ham_16_it, 'B')
errorbar(ham_16_t(1:500:end), ham_16_it(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
errorbar(ham_16_t(1:500:end), ham_16_it(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Input)'

subplot (2,2,2)
hold    on
plot    (ham_16_t, ham_16_ot, 'B')
errorbar(ham_16_t(1:500:end), ham_16_ot(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
errorbar(ham_16_t(1:500:end), ham_16_ot(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Output)'

subplot (2,2,3)
hold    on
plot    (ham_16_f, ham_16_if, 'B')
xlim    ([0, 1500])
errorbar(ham_16_f(1:500:end), ham_16_if(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_16_f(1:500:end), ham_16_if(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Input)'

subplot (2,2,4)
hold    on
plot    (ham_16_f, ham_16_of, 'B')
xlim    ([0, 1500])
ylim    ([0 2.5])
errorbar(ham_16_f(1:500:end), ham_16_of(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_16_f(1:500:end), ham_16_of(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Output)'
```

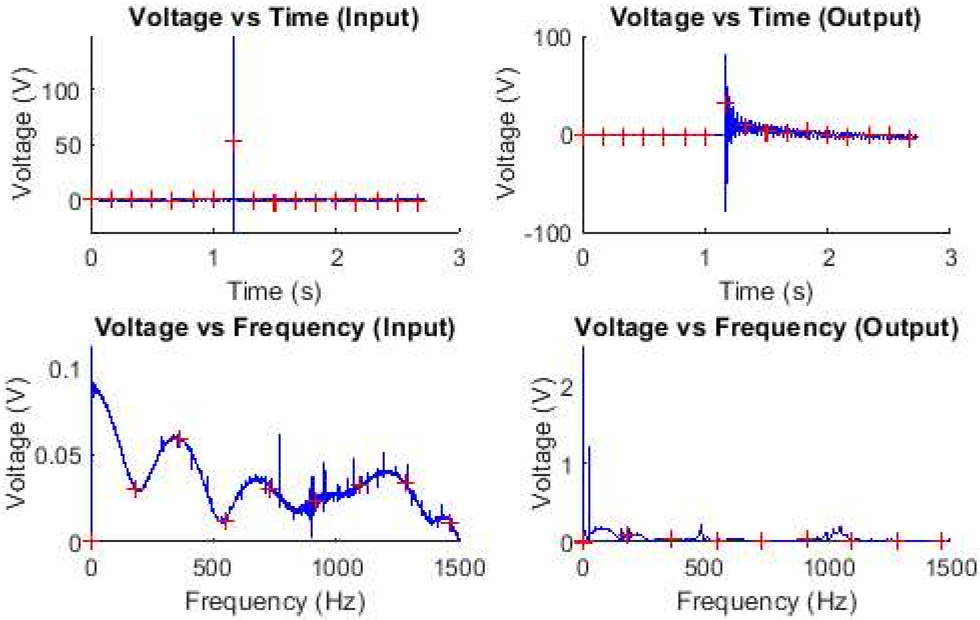
```
[v_max16, index16] = max(ham_16_of);
w_d16h  = ham_16_f(index16); % damped natural frequency of 24 inch beam
sgtitle ('Hammer Experiment 16 Inch Beam ')

% Hammer Test 20 inch Beam
figure  (6)
subplot (2,2,1)
hold    on
plot    (ham_20_t, ham_20_it, 'B')
errorbar(ham_20_t(1:500:end), ham_20_it(1:500:end), errorVt(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_20_t(1:500:end), ham_20_it(1:500:end), errorVt(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Input)'

subplot (2,2,2)
hold    on
plot    (ham_20_t, ham_20_ot, 'B')
errorbar(ham_20_t(1:500:end), ham_20_ot(1:500:end), errorVt(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_20_t(1:500:end), ham_20_ot(1:500:end), errorVt(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Output)'

subplot (2,2,3)
hold    on
plot    (ham_20_f, ham_20_if, 'B')
xlim    ([0, 1500])
errorbar(ham_20_f(1:500:end), ham_20_if(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_20_t(1:500:end), ham_20_if(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Input)'

subplot (2,2,4)
hold    on
plot    (ham_20_f, ham_20_of, 'B')
xlim    ([0, 1500])
errorbar(ham_20_f(1:500:end), ham_20_of(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_20_t(1:500:end), ham_20_of(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Output)'
[v_max20, index20] = max(ham_20_of);
w_d20h  = ham_20_f(index20); % damped natural frequency of 24 inch beam
sgtitle ('Hammer Experiment 20 Inch Beam')

% Hammer Test 24 inch Beam
figure  (7)
subplot (2,2,1)
hold    on
plot    (ham_24_t, ham_24_it, 'B')
errorbar(ham_24_t(1:500:end), ham_24_it(1:500:end), errorVt(1:500:end),...
```

```
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_24_t(1:500:end), ham_24_it(1:500:end), errorVt(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Input)'

subplot (2,2,2)
hold    on
plot    (ham_24_t, ham_24_ot, 'B')
errorbar(ham_24_t(1:500:end), ham_24_ot(1:500:end), errorVt(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_24_t(1:500:end), ham_24_ot(1:500:end), errorVt(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Time (s)'
ylabel  'Voltage (V)'
title   'Voltage vs Time (Output)'

subplot (2,2,3)
hold    on
plot    (ham_24_f, ham_24_if, 'B')
xlim    ([0, 1500])
errorbar(ham_24_f(1:500:end), ham_24_if(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_24_f(1:500:end), ham_24_if(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Input)'

subplot (2,2,4)
hold    on
plot    (ham_24_f, ham_24_of, 'B')
xlim    ([0, 1500])
errorbar(ham_24_f(1:500:end), ham_24_of(1:500:end), errorVf(1:500:end),...
    'Horizontal', 'R', "LineStyle", 'none')
errorbar(ham_24_f(1:500:end), ham_24_of(1:500:end), errorVf(1:500:end),...
    'Vertical', 'R', "LineStyle", 'none')
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Voltage vs Frequency (Output)'
[v_max24, index24] = max(ham_24_of);
w_d24h  = ham_24_f(index24); % damped natural frequency of 24 inch beam
sgtitle ('Hammer Experiment 24 Inch Beam ')


% Transfer Function Calculations

% Transfer Function for 16 inch Beam
fft_ham_16o = fft(ham_16_ot);
fft_ham_16i = fft(ham_16_it);
Hf16        = fft_ham_16o ./ fft_ham_16i;
Hf16i       = imag(Hf16);
Hf16r       = real(Hf16);
mag1_resp16 = sqrt((Hf16i .^ 2) + (Hf16r .^ 2));
mag2_resp16 = abs(Hf16);
phase16     = atan(Hf16i ./ Hf16r) .* (180 ./ pi);

% Transfer Function for 20 inch Beam
fft_ham_20o = fft(ham_20_ot);
fft_ham_20i = fft(ham_20_it);
```

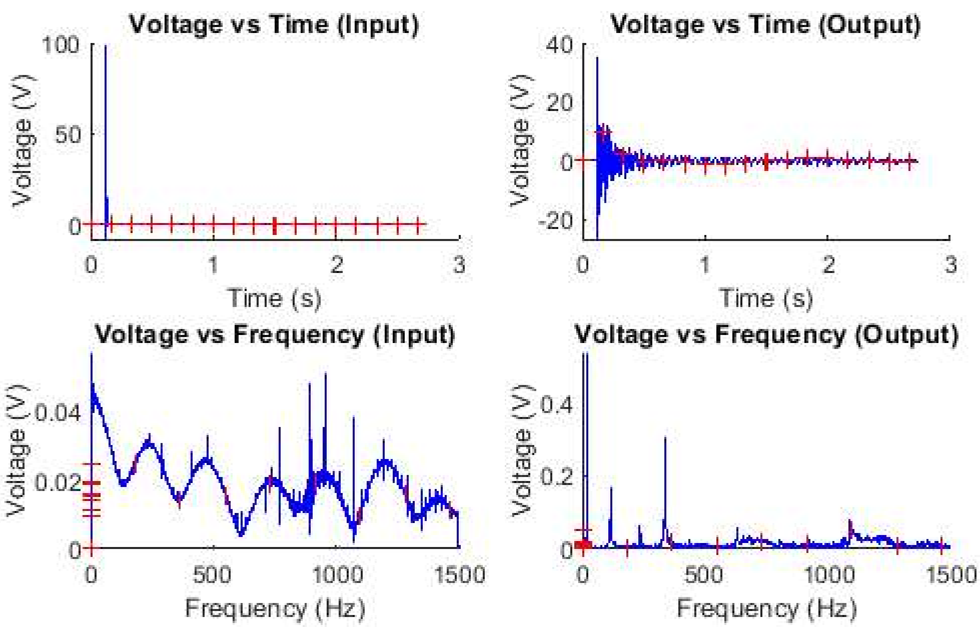
```
Hf20        = fft_ham_20o ./ fft_ham_20i;
Hf20i       = imag(Hf20);
Hf20r       = real(Hf20);
mag1_resp20 = sqrt((Hf20i .^ 2) + (Hf20r .^ 2));
mag2_resp20 = abs(Hf20);
phase20     = atan(Hf20i ./ Hf20r) .* (180 ./ pi);

% Transfer Function for 20 inch Beam
fft_ham_24o = fft(ham_24_ot);
fft_ham_24i = fft(ham_24_it);
Hf24        = fft_ham_24o ./ fft_ham_20i;
Hf24i       = imag(Hf24);
Hf24r       = real(Hf24);
mag1_resp24 = sqrt((Hf24i .^ 2) + (Hf24r .^ 2));
mag2_resp24 = abs(Hf24);
phase24     = atan(Hf20i ./ Hf24r) .* (180 ./ pi);

% Plot Fourier Transfer Functions

% 16 inch beam
figure   (8)
subplot  (2,1,1)
hold     on
plot     (ham_16_f, mag2_resp16, 'b')
xlim     ([0, 1500]);
errorf   = u_f .* ones(size(ham_16_f));     % frequency error bar size
errorV   = u_V .* ones(size(mag2_resp16));  % amplitude error bar size
errorbar (ham_16_f(1:200:end), mag2_resp16(1:200:end), errorf(1:200:end),...
    'Vertical', 'm', "LineStyle", 'none');
errorbar (ham_16_f(1:200:end), mag2_resp16(1:200:end), errorV(1:200:end),...
    'Horizontal', 'm', "LineStyle", 'none');
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Magnitude of Response vs Frequency (16 in.)')

subplot  (2,1,2)
plot     (ham_16_f, phase16, 'b')
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Phase Lag vs Frequency (16 in.)')
xlim     ([0, 1500]);

% 20 inch beam
figure   (9)
subplot  (2,1,1)
hold     on
plot     (ham_20_f, mag2_resp20, 'b')
xlim     ([0, 1500]);
errorbar (ham_20_f(1:200:end), mag2_resp20(1:200:end), errorf(1:200:end),...
    'Vertical', 'm', "LineStyle", 'none');
errorbar (ham_20_f(1:200:end), mag2_resp20(1:200:end), errorV(1:200:end),...
    'Horizontal', 'm', "LineStyle", 'none');
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Magnitude of Response vs Frequency (20 in.)')

subplot  (2,1,2)
plot     (ham_20_f, phase20, 'b')
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Phase Lag vs Frequency (20 in.)')
```

```
xlim     ([0, 1500]);


% 24 inch beam
figure   (10)
subplot  (2,1,1)
hold     on
plot     (ham_24_f, mag2_resp24, 'b')
xlim     ([0, 1500]);
errorbar (ham_24_f(1:200:end), mag2_resp24(1:200:end), errorf(1:200:end),...
    'Vertical', 'm', "LineStyle", 'none');
errorbar (ham_24_f(1:200:end), mag2_resp24(1:200:end), errorV(1:200:end),...
    'Horizontal', 'm', "LineStyle", 'none');
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Magnitude of Response vs Frequency (24 in.)')

subplot  (2,1,2)
plot     (ham_24_f, phase24, 'b')
xlabel   ('Frequency (Hz)')
ylabel   ('Amplitude (V)')
title    ('Phase Lag vs Frequency (24 in.)')
xlim     ([0, 1500]);
hold     off

% Hammer Test Calculations

% Evaluate Expressions

w_nval = double(subs(w_n));           % Hz
w_nrad = double(subs(w_nr));          % rad/s
k_eqs  = double(subs(k_eq));          % N/m
w_dh   = [w_d16h, w_d20h, w_d24h];    % Hz
w_drad = double(w_dval) .* 2 .* pi;   % rad/s
zetah  = double(subs(zetah));
```

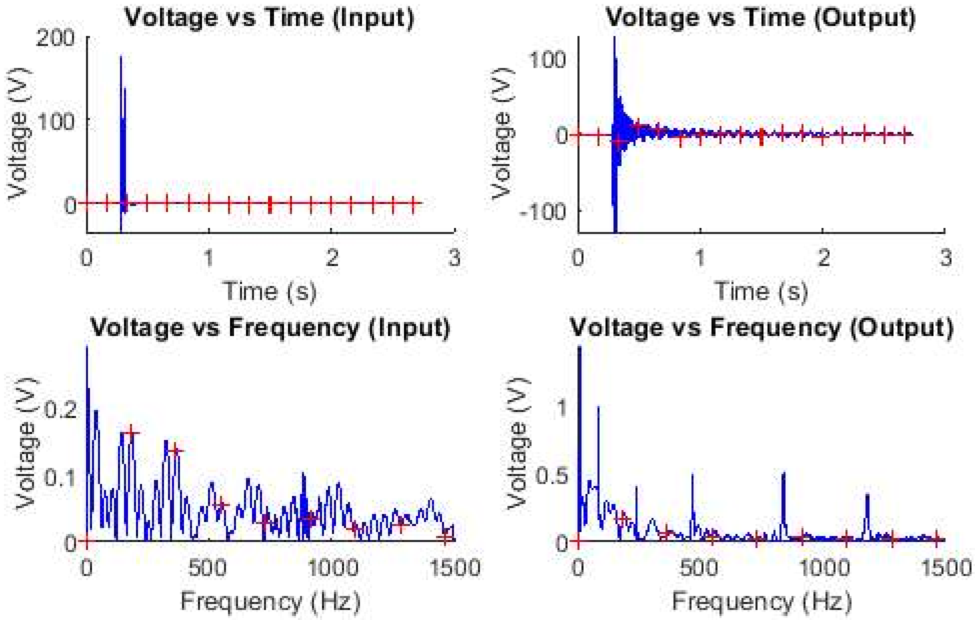
Hammer Experiment 16 Inch Beam

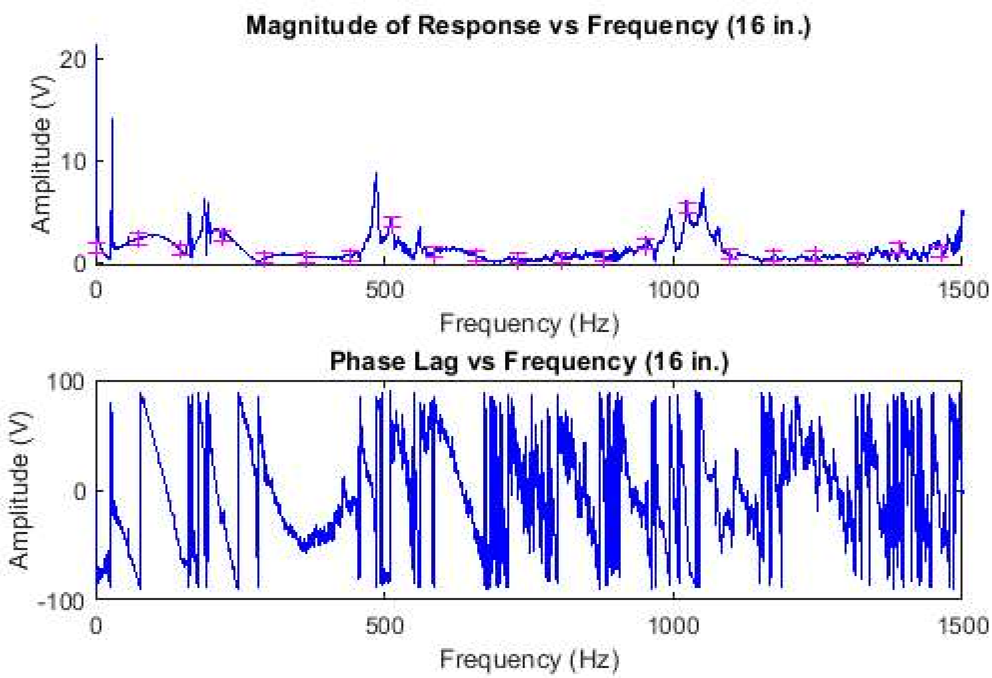


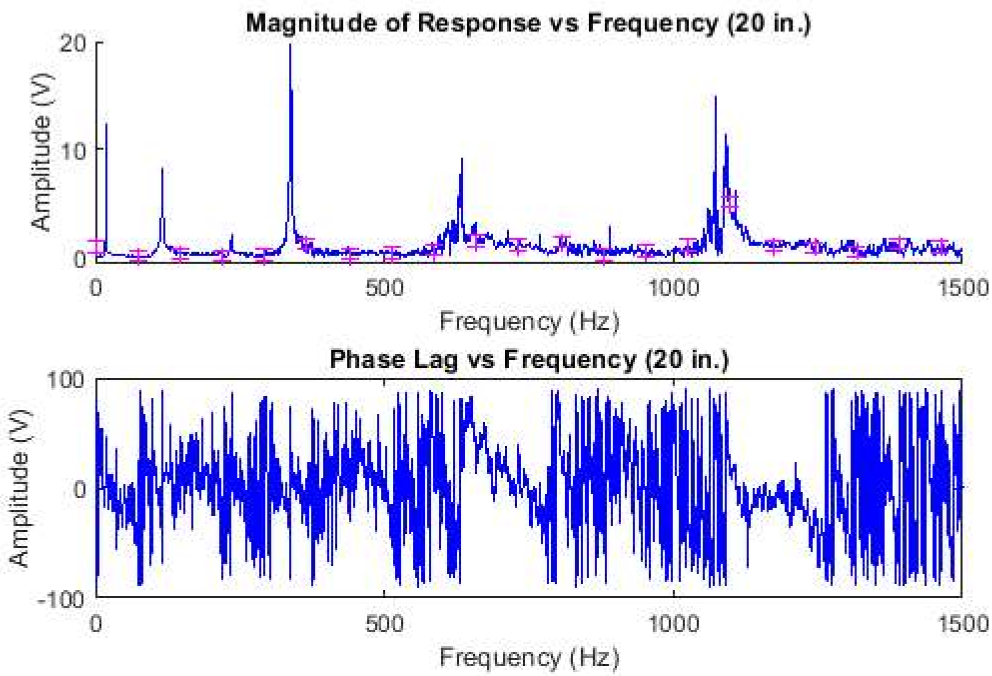
Hammer Experiment 20 Inch Beam

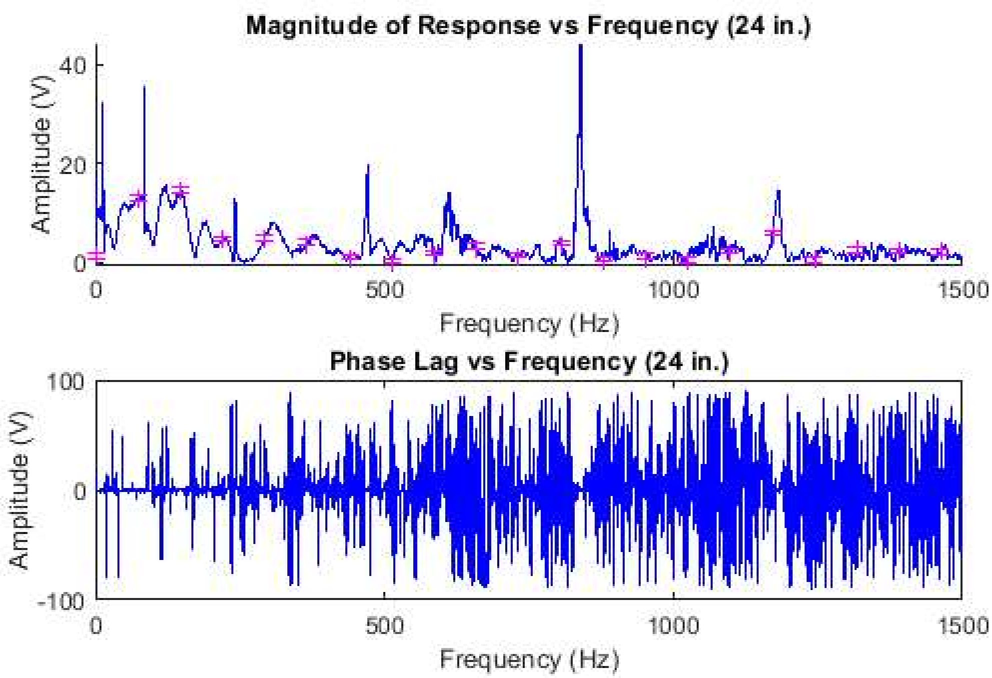


Hammer Experiment 24 Inch Beam









Simulations

```
syms L b h res_tape W res_freq res_rho rho E bi_r n_bits c u_E

% Define Governing Equations
vol    = L .* b .* h;                  % volume of beam
m_beam = rho .* vol;                   % mass of beam
m_eq   = ((33./140) .* m_beam) + W; % mass of entire system
I      = (1/12) .* b .* (h.^3);      % moment of inertia
k_eq   = (3 .* E .* I) ./ (L.^3);   % spring constant of the beam
w_nr   = sqrt(k_eq ./ m_eq);         % undamped natural frequency (rad/s)
w_n    = w_nr ./ (2 .* pi);          % undamped natural frequency (Hz)
zeta   = c ./ (2 * m_eq * w_n);      % damping ratio
```

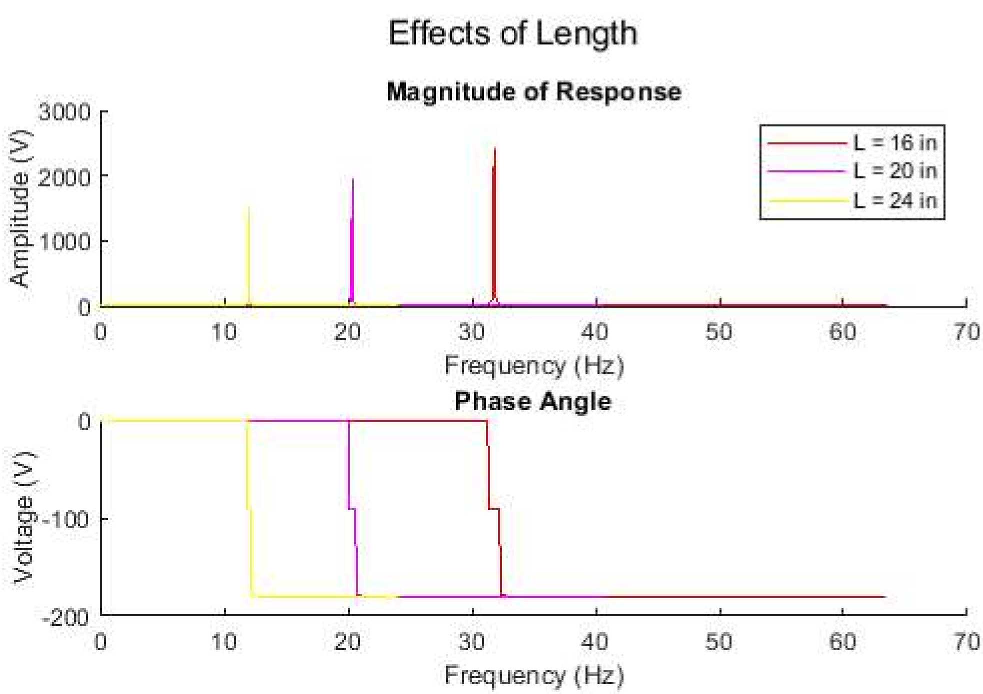
```
w_d    = w_n * sqrt(1 - (zeta).^2); % damped natural frequency

E      = 2.06 .* 10 .^ 11;          % Pa
rho    = 7850;                      % kg/m^3
bi_r   = 5;                         % V
n_bits = 24;                        % bits
```

Simulation 3.1

Effects of length

```
% import data
sim31_L1_dat = importdata ('10_24_2022kilian3.1kilian dat.txt');
sim31_L1_dat = sim31_L1_dat.data;
sim31_L1_fft = importdata ('10_24_2022kilian3.1kilian fft.txt');
sim31_L1_fft = sim31_L1_fft.data;
sim31_L2_dat = importdata ('10_24_2022kilian3.1.2kilian dat.txt');
sim31_L2_dat = sim31_L2_dat.data;
sim31_L2_fft = importdata ('10_24_2022kilian3.1.2kilian fft.txt');
sim31_L2_fft = sim31_L2_fft.data;
sim31_L3_dat = importdata ('10_24_2022kilian3.1.3kilian dat.txt');
sim31_L3_dat = sim31_L3_dat.data;
sim31_L3_fft = importdata ('10_24_2022kilian3.1.3kilian fft.txt');
sim31_L3_fft = sim31_L3_fft.data;

% L = 16 in
freq_L1 = sim31_L1_dat(:,1); % Hz
amp_L1  = sim31_L1_fft(:,2); % V
volt_L1 = sim31_L1_dat(:,2); % V

% L = 20 in
freq_L2 = sim31_L2_dat(:,1); % Hz
amp_L2  = sim31_L2_fft(:,2); % V
volt_L2 = sim31_L2_dat(:,2); % V

% L = 24 in
freq_L3 = sim31_L3_dat(:,1); % Hz
amp_L3  = sim31_L3_fft(:,2); % V
volt_L3 = sim31_L3_dat(:,2); % V

% assign values
L1 = .4064;  % length (m)
L2 = .508;   % length (m)
L3 = .6604;  % length (m)
w  = .0254;  % width (m)
th = .00635; % thickness (m)
c  = .01;    % damping coefficcient
W  = 0;      % end weight (kg)
L  = [L1, L2, L3];
h  = th;
b  = w;

w_n_L  = double(subs(w_n));    % Hz
zeta_L = double(subs(zeta));

% plot

figure  (11)
subplot (2,1,1)
```

```
hold    on
pL11    = plot(freq_L1, amp_L1,'-r');
pL12    = plot(freq_L2, amp_L2,'-m');
pL13    = plot(freq_L3, amp_L3,'-y');
xlabel  'Frequency (Hz)'
ylabel  'Amplitude (V)'
title   'Magnitude of Response'

subplot (2,1,2)
hold    on
pL21    = plot(freq_L1, volt_L1,'-r');
pL22    = plot(freq_L2, volt_L2,'-m');
pL23    = plot(freq_L3, volt_L3,'-y');
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Phase Angle'

legend  ([pL11, pL12, pL13], 'L = 16 in', 'L = 20 in', 'L = 24 in')
sgtitle 'Effects of Length'
```

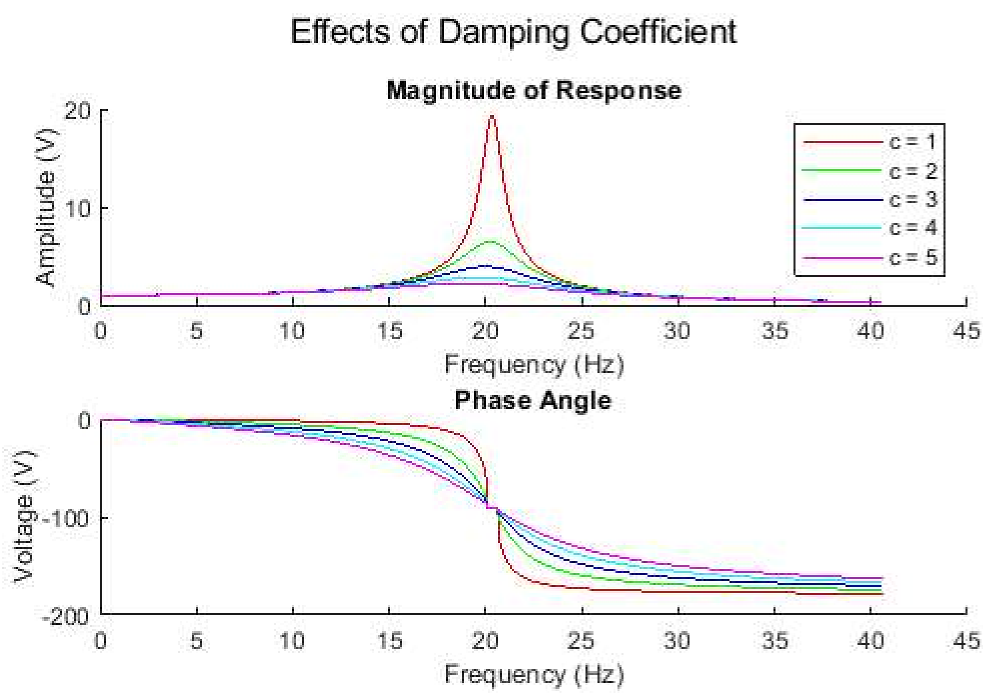


Simulation 3.2

```
% Effects of damping coefficient
% import data
sim32_c1_dat = importdata ('10_24_2022kilian3.2kilian dat.txt');
sim32_c1_dat = sim32_c1_dat.data;
sim32_c1_fft = importdata ('10_24_2022kilian3.2kilian fft.txt');
sim32_c1_fft = sim32_c1_fft.data;
sim32_c2_dat = importdata ('10_24_2022kilian3.2.2kilian dat.txt');
sim32_c2_dat = sim32_c2_dat.data;
sim32_c2_fft = importdata ('10_24_2022kilian3.2.2kilian fft.txt');
sim32_c2_fft = sim32_c2_fft.data;
sim32_c3_dat = importdata ('10_24_2022kilian3.2.3kilian dat.txt');
sim32_c3_dat = sim32_c3_dat.data;
sim32_c3_fft = importdata ('10_24_2022kilian3.2.3kilian fft.txt');
sim32_c3_fft = sim32_c3_fft.data;
```

```
sim32_c4_dat = importdata ('10_24_2022kilian3.2.4kilian dat.txt');
sim32_c4_dat = sim32_c4_dat.data;
sim32_c4_fft = importdata ('10_24_2022kilian3.2.4kilian fft.txt');
sim32_c4_fft = sim32_c4_fft.data;
sim32_c5_dat = importdata ('10_24_2022kilian3.2.5kilian dat.txt');
sim32_c5_dat = sim32_c5_dat.data;
sim32_c5_fft = importdata ('10_24_2022kilian3.2.5kilian fft.txt');
sim32_c5_fft = sim32_c5_fft.data;

% c = 1
freq_c1 = sim32_c1_dat(:,1); % Hz
amp_c1  = sim32_c1_fft(:,2); % V
volt_c1 = sim32_c1_dat(:,2); % V

% c = 3
freq_c2 = sim32_c2_dat(:,1); % Hz
amp_c2  = sim32_c2_fft(:,2); % V
volt_c2 = sim32_c2_dat(:,2); % V

% c = 5
freq_c3 = sim32_c3_dat(:,1); % Hz
amp_c3  = sim32_c3_fft(:,2); % V
volt_c3 = sim32_c3_dat(:,2); % V

% c = 7
freq_c4 = sim32_c4_dat(:,1); % Hz
amp_c4  = sim32_c4_fft(:,2); % V
volt_c4 = sim32_c4_dat(:,2); % V

% c = 9
freq_c5 = sim32_c5_dat(:,1); % Hz
amp_c5  = sim32_c5_fft(:,2); % V
volt_c5 = sim32_c5_dat(:,2); % V

% assign values
L  = .508;          % m
W  = 0;             % kg
c  = [1 3 5 7 9];
w  = .0254;         % m
th = .00635;        % m
h  = th;
b  = w;

% Evaluate Expressions
w_n_c  = double(subs(w_n)) .* 2 .* pi; % rad/s
zeta_c = double(subs(zeta));
w_d_c  = double(subs(w_d));            % Hz

figure  (12)
subplot (2,1,1)
hold    on
pc11    = plot (freq_c1, amp_c1, '-r');
pc12    = plot (freq_c2, amp_c2, '-g');
pc13    = plot (freq_c3, amp_c3, '-b');
pc14    = plot (freq_c4, amp_c4, '-c');
pc15    = plot (freq_c5, amp_c5, '-m');
xlabel  'Frequency (Hz)'
ylabel  'Amplitude (V)'
title   'Magnitude of Response'
subplot (2,1,2)
hold    on
```

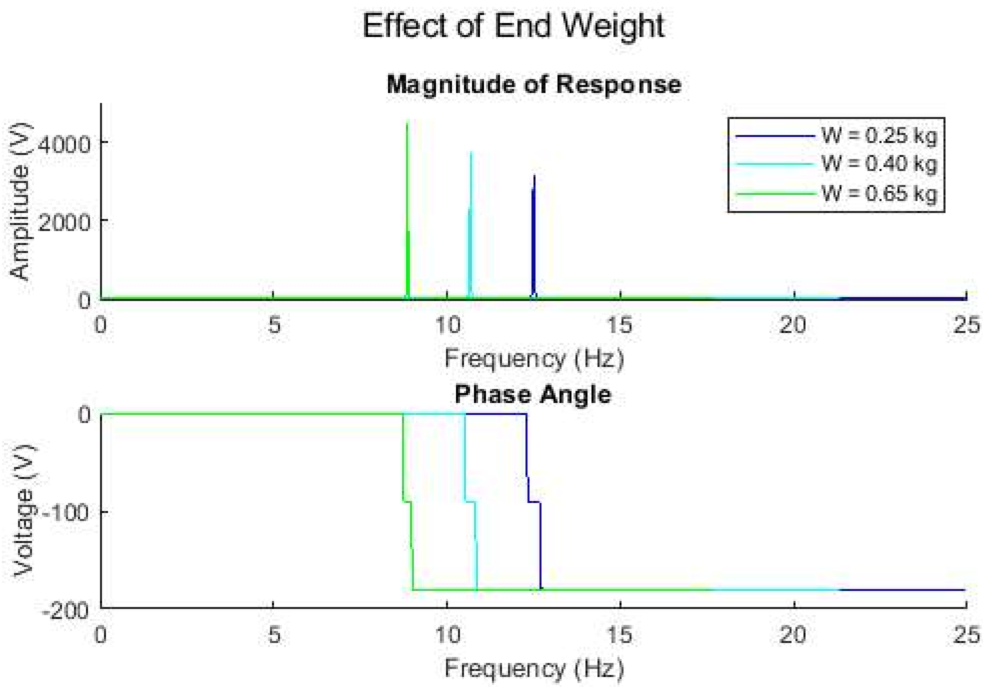
```
pc21    = plot (freq_c1, volt_c1, '-r');
pc22    = plot (freq_c2, volt_c2, '-g');
pc23    = plot (freq_c3, volt_c3, '-b');
pc24    = plot (freq_c4, volt_c4, '-c');
pc25    = plot (freq_c5, volt_c5, '-m');
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Phase Angle'
legend  ([pc11, pc12, pc13, pc14, pc15], ...
    'c = 1', 'c = 2', 'c = 3', 'c = 4', 'c = 5')
sgtitle 'Effects of Damping Coefficient'
```



Simulation 3.3.1

```
% Effects of end weight
% import data
sim331_W25_dat = importdata('10_24_2022kilian3.3.1kilian dat.txt');
sim331_W25_dat = sim331_W25_dat.data;
sim331_W25_fft = importdata('10_24_2022kilian3.3.1kilian fft.txt');
sim331_W25_fft = sim331_W25_fft.data;
sim331_W4_dat  = importdata('10_24_2022kilian3.3.2kilian dat.txt');
sim331_W4_dat  = sim331_W4_dat.data;
sim331_W4_fft  = importdata('10_24_2022kilian3.3.2kilian fft.txt');
sim331_W4_fft  = sim331_W4_fft.data;
sim331_W65_dat = importdata('10_24_2022kilian3.3.3kilian dat.txt');
sim331_W65_dat = sim331_W65_dat.data;
sim331_W65_fft = importdata('10_24_2022kilian3.3.3kilian fft.txt');
sim331_W65_fft = sim331_W65_fft.data;

% W = .25 kg
freq_W1 = sim331_W25_dat(:,1); % Hz
amp_W1  = sim331_W25_fft(:,2); % V
volt_W1 = sim331_W25_dat(:,2); % V

% W = .4 kg
freq_W2 = sim331_W4_dat(:,1);  % Hz
```

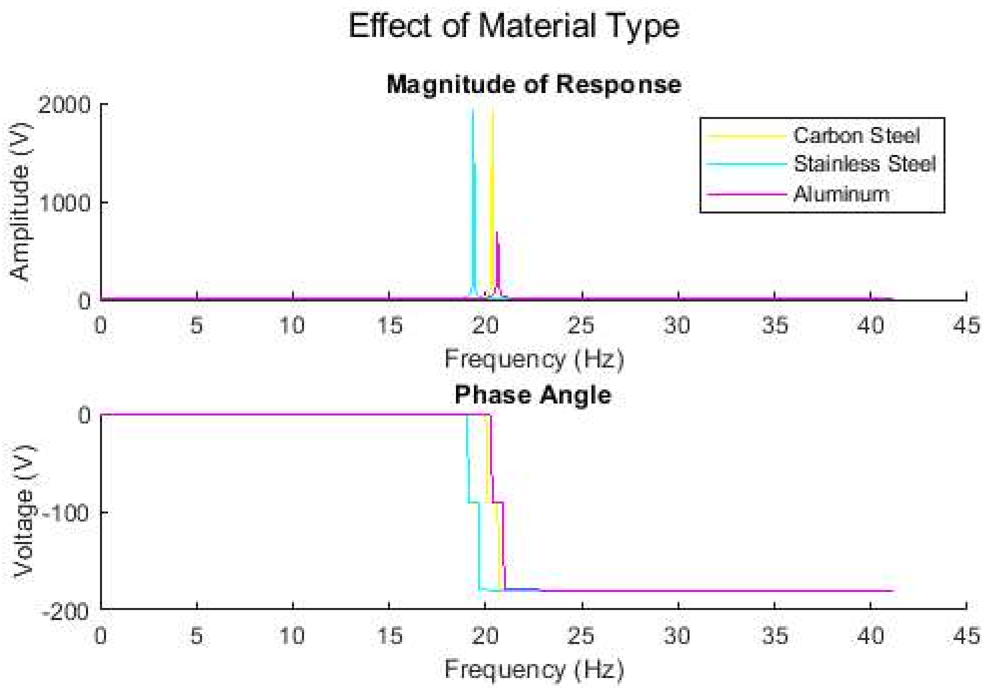
```
amp_W2  = sim331_W4_fft(:,2);  % V
volt_W2 = sim331_W4_dat(:,2);  % V

% W = .65 kg
freq_W3 = sim331_W65_dat(:,1); % Hz
amp_W3  = sim331_W65_fft(:,2); % V
volt_W3 = sim331_W65_dat(:,2); % V

% assign values
L  = .508;      % m
w  = .0254;     % m
th = .00635;    % m
c  = .01;
W  = [.25 .4 .65]; % kg
h  = th;
b  = w;

% Evaluate Expressions
w_n_Wrad  = double(subs(w_n)) .* 2 .* pi; % rad/s
zeta_W    = double(subs(zeta));
w_n_WHz   = double(subs(w_n));            % Hz


figure  (13)
subplot (2,1,1)
hold    on
pW11    = plot (freq_W1, amp_W1, '-b' );
pW12    = plot (freq_W2, amp_W2, '-c' );
pW13    = plot (freq_W3, amp_W3, '-g' );
ylim    ([0 5000])
xlabel  'Frequency (Hz)'
ylabel  'Amplitude (V)'
title   'Magnitude of Response'

subplot (2,1,2)
hold    on
pW21    = plot (freq_W1, volt_W1, '-b' );
pW22    = plot (freq_W2, volt_W2, '-c' );
pW23    = plot (freq_W3, volt_W3, '-g' );
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Phase Angle'

legend  ([pW11, pW12, pW13], 'W = 0.25 kg', 'W = 0.40 kg', 'W = 0.65 kg')
sgtitle 'Effect of End Weight'
```



Simulation 3.3.2

```
% Effects of material type
% import data
sim332_carb_dat  = importdata('10_24_2022kilian3.3.2.1kilian dat.txt');
sim332_carb_dat  = sim332_carb_dat.data;
sim332_carb_fft  = importdata('10_24_2022kilian3.3.2.1kilian fft.txt');
sim332_carb_fft  = sim332_carb_fft.data;
sim332_stain_dat = importdata('10_24_2022kilian3.3.2.2kilian dat.txt');
sim332_stain_dat = sim332_stain_dat.data;
sim332_stain_fft = importdata('10_24_2022kilian3.3.2.2kilian fft.txt');
sim332_stain_fft = sim332_stain_fft.data;
sim332_alum_dat  = importdata('10_24_2022kilian3.3.2.3kilian dat.txt');
sim332_alum_dat  = sim332_alum_dat.data;
sim332_alum_fft  = importdata('10_24_2022kilian3.3.2.3kilian fft.txt');
sim332_alum_fft  = sim332_alum_fft.data;

% material = carbon steel
freq_carb  = sim332_carb_dat(:,1);  % Hz
amp_carb   = sim332_carb_fft(:,2);  % V
volt_carb  = sim332_carb_dat(:,2);  % V

% material = stainless steel
freq_stain = sim332_stain_dat(:,1); % Hz
amp_stain  = sim332_stain_fft(:,2); % V
volt_stain = sim332_stain_dat(:,2); % V

% material = aluminum
freq_alum  = sim332_alum_dat(:,1);  % Hz
amp_alum   = sim332_alum_fft(:,2);  % V
volt_alum  = sim332_alum_dat(:,2);  % V

% assign values
L   = .508;      % m
w   = .0254;     % m
th  = .00635;    % m
c   = .01;
```

```
W    = 0;          % kg
b    = w;
h    = th;
rhoss = 7930;     % kg/m^3
rhocs = 7850;     % kg/m^3
rhoal = 2700;     % kg/m^3
Ess    = 1.93 .* 10 .^ 11;    % Pa
Ecs    = 2.06 .* 10 .^ 11;    % Pa
Eal    = .69 .* 10 .^ 11;     % Pa
rho    = [rhoss, rhocs, rhoal];
E      = [Ess, Ecs, Eal];

% Evaluate Expressions
w_n_mrad  = double(subs(w_n)) .* 2 .* pi; % rad/s
zeta_m    = double(subs(zeta));
w_n_mHz   = double(subs(w_n)) ;           % Hz
k_eq_m    = double(subs(k_eq));           % N/m


figure  (14)
subplot (2,1,1)
hold    on
p1c     = plot(freq_carb, amp_carb, '-y');
p1s     = plot(freq_stain, amp_stain, '-c');
p1a     = plot(freq_alum, amp_alum, '-m');
xlabel  'Frequency (Hz)'
ylabel  'Amplitude (V)'
title   'Magnitude of Response'

subplot (2,1,2)
hold    on
p2c     = plot(freq_carb, volt_carb, '-y');
p2s     = plot(freq_stain, volt_stain, '-c');
p2a     = plot(freq_alum, volt_alum, '-m');
xlabel  'Frequency (Hz)'
ylabel  'Voltage (V)'
title   'Phase Angle'
legend  ([p1c, p1s, p1a], 'Carbon Steel', 'Stainless Steel', 'Aluminum')
sgtitle 'Effect of Material Type'


% Calculated Values

syms L b h res_tape res_freq res_rho rho E bi_r n_bits c u_E

% Define Governing Equations
vol     = L .* b .* h;                % volume of beam
m_beam = rho .* vol;                  % mass of beam
m_eq    = ((33./140) .* m_beam);      % mass of entire system
I       = (1/12) .* b .* (h.^3);      % moment of inertia
k_eq    = (3 .* E .* I) ./ (L.^3);    % spring constant of the beam
w_nr    = sqrt(k_eq ./ m_eq);         % undamped natural frequency (rad/s)
w_n     = w_nr ./ (2 .* pi);          % undamped natural frequency (Hz)
zeta    = c ./ (2 * m_eq * w_n);      % damping ratio
w_d     = w_n * sqrt(1 - (zeta).^2); % damped natural frequency
```



Published with MATLAB® R2022a