Charlie Nitschelm
ME 646
3/3/2019

Lab 3

Part 1 – Beam/Strain Gauge Response

1a)

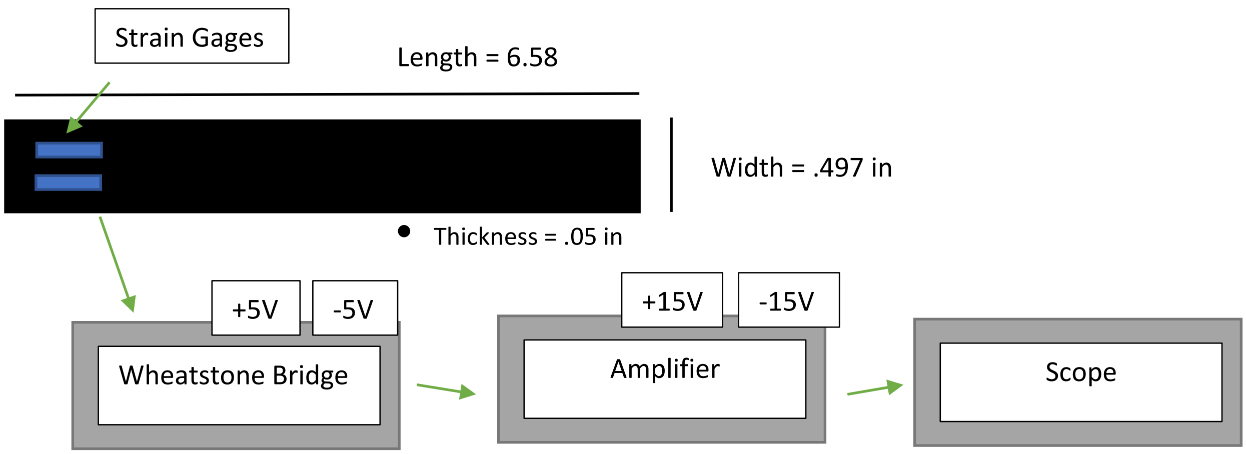
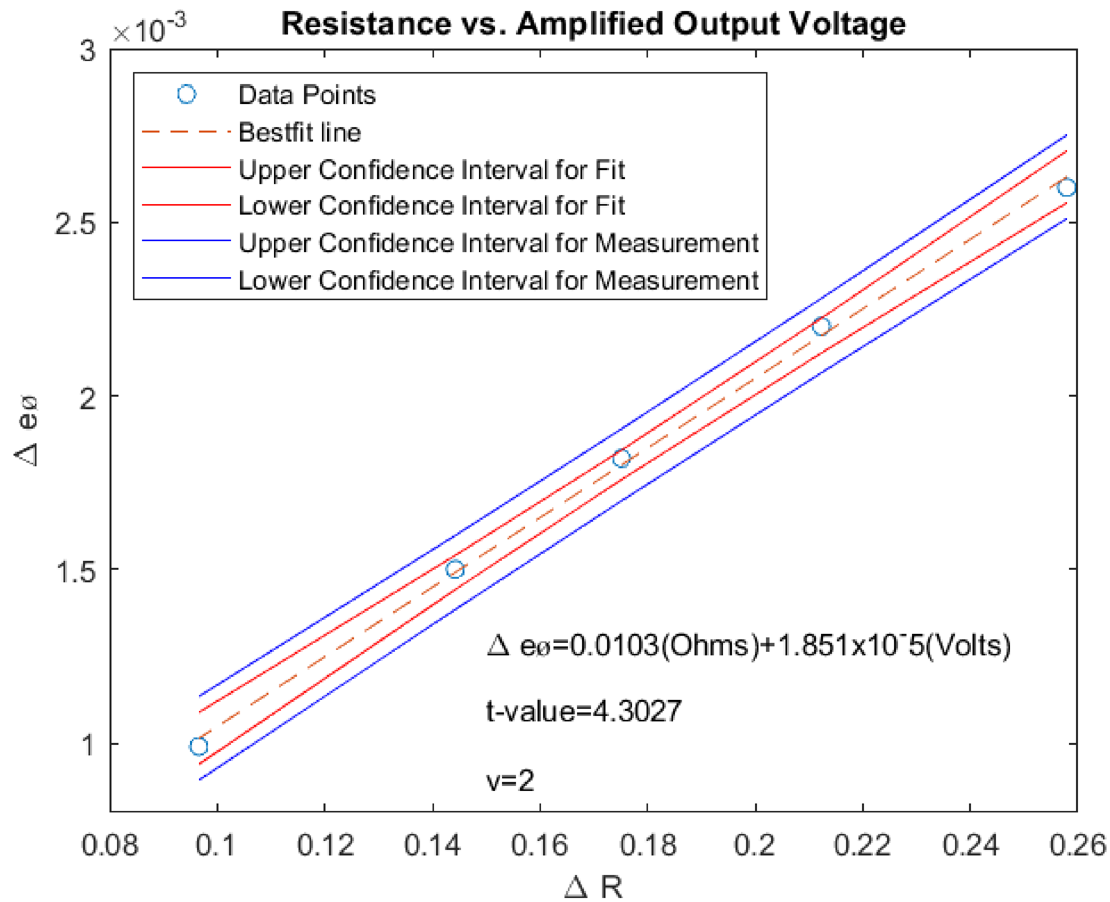


Figure 1 - Beam and strain gauge setup using a Wheatstone bridge to pull in the voltage, an amplifier to increase the readings, and a scope to record the data

1b)



Solving for ΔR in the Wheatstone Bridge

$$\Delta R = \frac{1}{\frac{1}{R1} + \frac{1}{R3\_eff}}$$

$where\ R3\_eff = R3||R_{shunt}$ and $R1 = R_{strain\_gage}$

$$\Delta R = R_{strain\_gage} - \left(\frac{1}{R3} + \frac{1}{R_{shunt}}\right)$$

And so ΔR reduces to

$$\Delta R = R_{strain\_gage} - \frac{R3 * R_{shunt}}{R3 + R_{shunt}}$$

Bridge Sensitivity

Using MATLAB, an observed bridge sensitivity of 0.01 V/Ω was calculated from the data. To calculate the expected bridge sensitivity using primarily the Wheatstone bridge setup, we must derive the equation using a quarter Wheatstone bridge.

$$E_0 = E_i\left(\frac{R1}{R1+R2} - \frac{R3}{R3+R4}\right)$$

Where R2 = R3 = R4 = R and R1 = R + ΔR.

$$E_0 = E_i\left(\frac{2(R+\Delta\mathrm{R}) - (2\mathrm{R}+\Delta\mathrm{R})}{2(2R+\Delta\mathrm{R})}\right)$$

And as ΔR approaches zero, the equation becomes:

$$\frac{\Delta E_0}{E_i} = \frac{\Delta\mathrm{R}}{4R} \ or \ \frac{\Delta E_0}{\Delta\mathrm{R}} = \frac{E_i}{4R}$$

Using this equation, an expected bridge sensitivity of 0.0104 V/Ω was found using an input voltage of 5V and knowing the resistors used were 120 Ω

1c)

The equation that relates the strain the beam experiences, the gage factor, the voltage gain input and the input voltage is derived from the half Wheatstone bridge by

$$\frac{dE_0}{E_i} = \frac{GF}{4}(\mathcal{E}_1 - \mathcal{E}_2 - \mathcal{E}_3 + \mathcal{E}_4)$$

And with the $\mathcal{E}_3 = \mathcal{E}_4 = 0$ and $\mathcal{E}_1 = \mathcal{E}_b + \mathcal{E}_{thermal}$ and $\mathcal{E}_2 = -\mathcal{E}_b + \mathcal{E}_{thermal}$ the equation becomes:

$$\mathcal{E}_b = \frac{2 * dE_0}{E_i * GF}$$

1d)

Using the cantilever beam equations, we see that

$$R_A = F \text{ And } M_{max} = P(L - x)$$

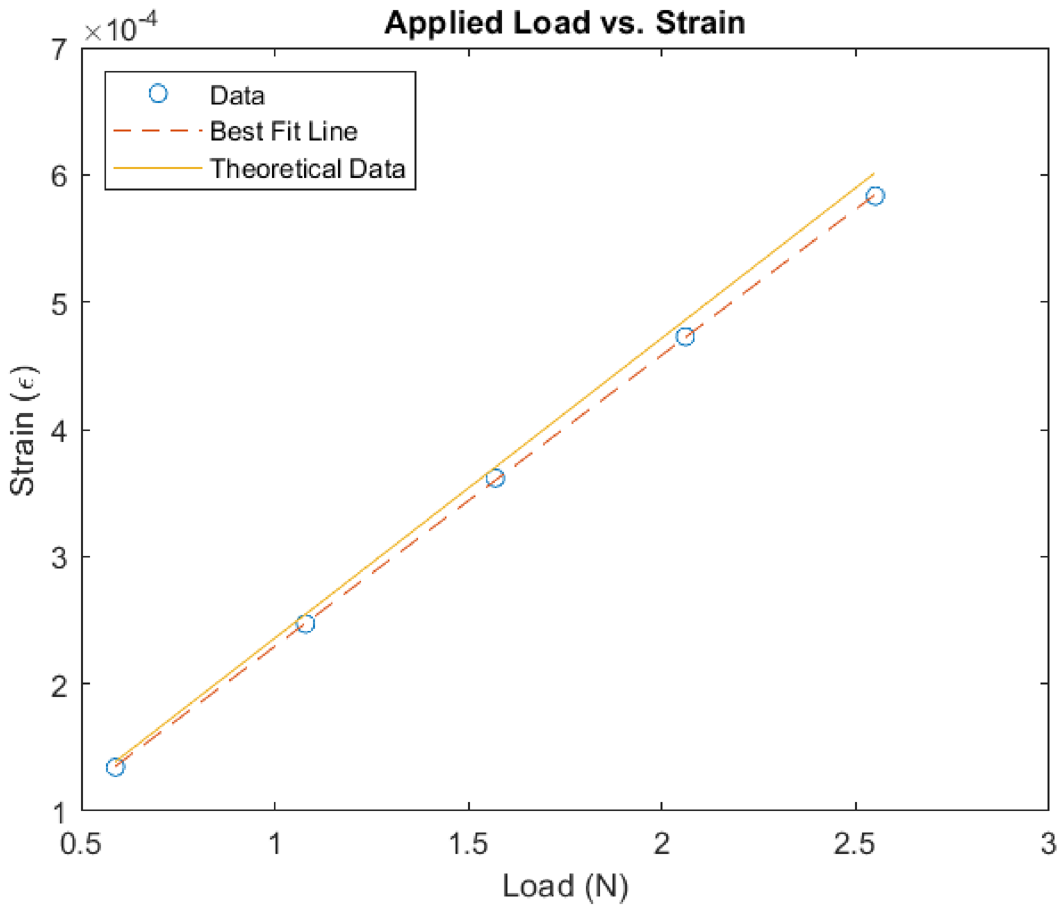
where P is the applied load, L is the length of the beam and x is the position of the strain gages on the beam. Calculating for stress we find that

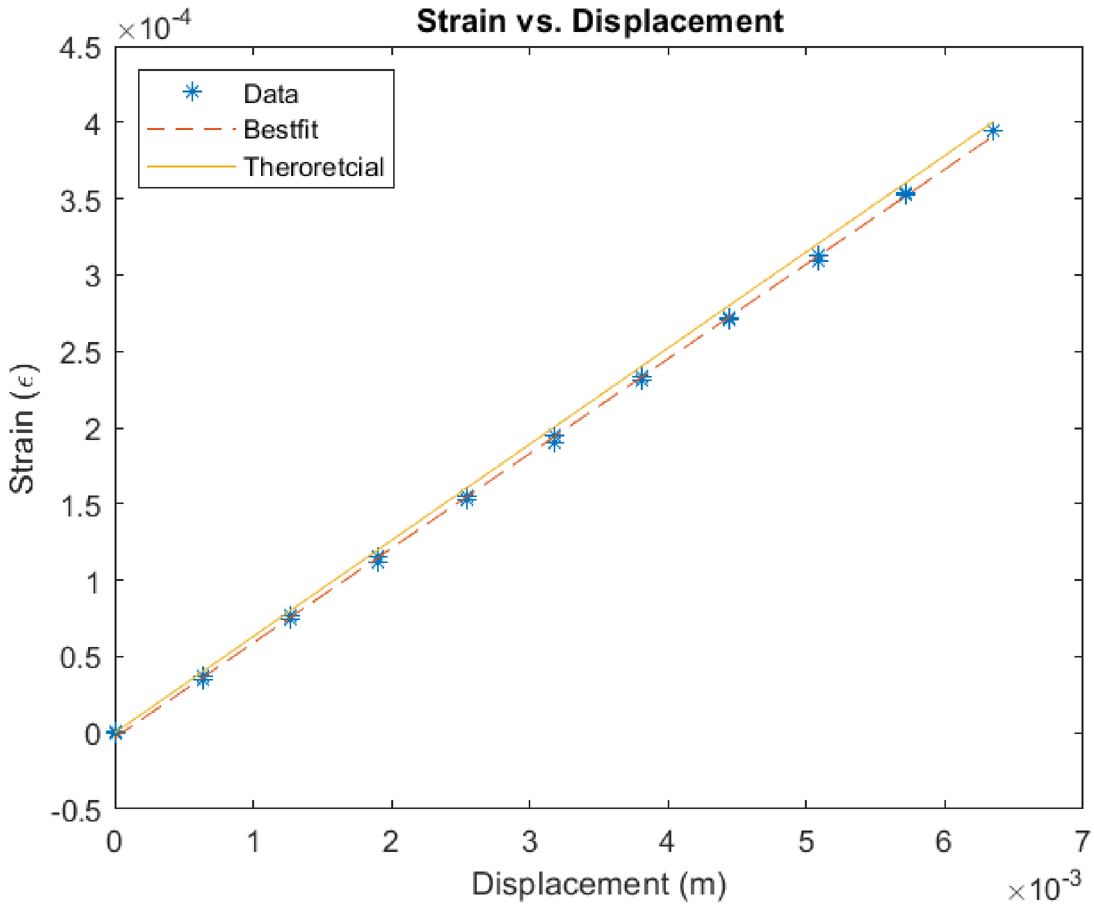
$$\sigma = \frac{My}{I} = E\varepsilon$$

Where $I = \frac{bh^3}{12}$ and $y = \frac{h}{2}$ on the surface. With these equations, we can integrate and solve for the maximum displacement, or vice versa and get the fundamental equation for a cantilever beam

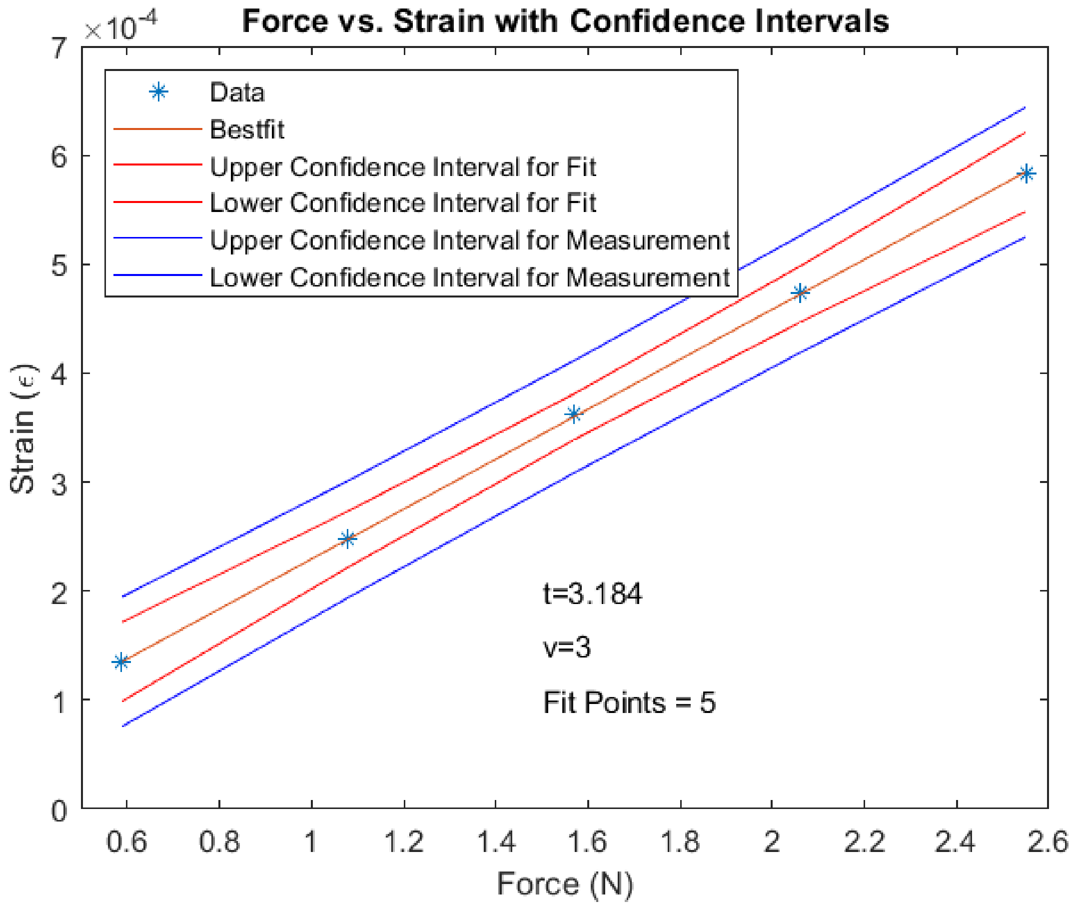
$$\delta_{max} = \frac{PL^3}{3EI}$$

By using these formulas, strain values can then be found by the applied force and displacement of the entire beam. The two figures below show these calculation from a various force data set applied to the beam.

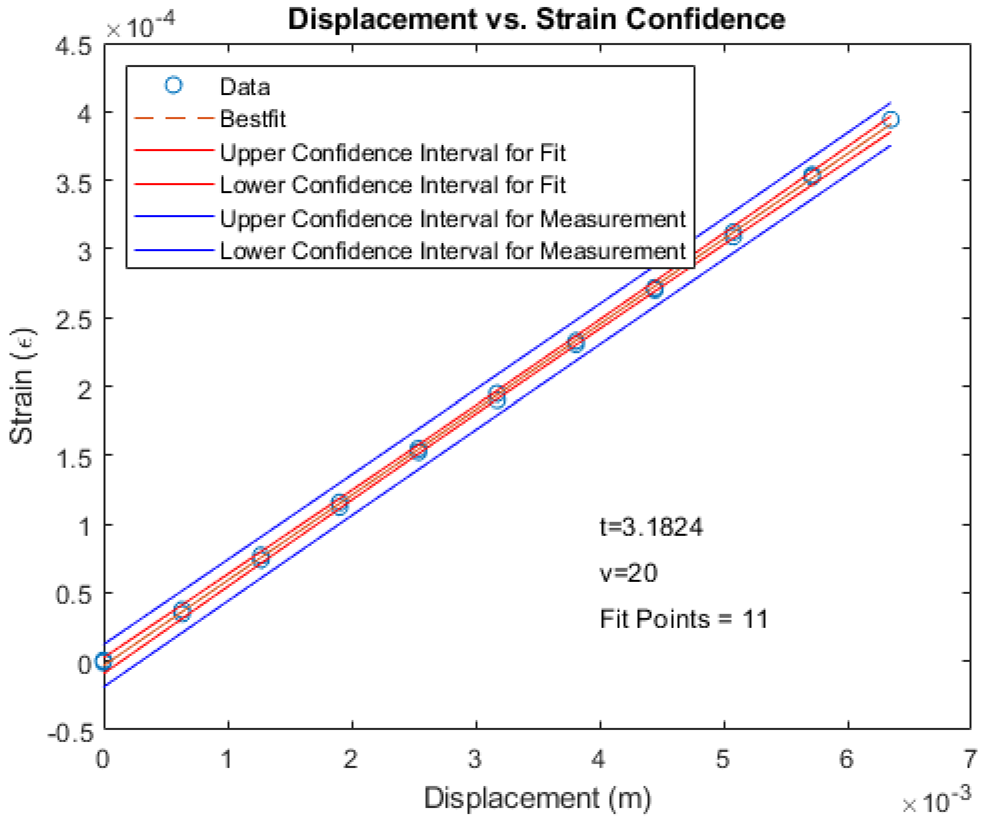




A confidence interval of 95% was then applied to the measurements yielding the strain vs displacement and strain vs. force plots.



Force vs. Strain with Confidence Intervals
×10^-4
Data
Bestfit
Upper Confidence Interval for Fit
Lower Confidence Interval for Fit
Upper Confidence Interval for Measurement
Lower Confidence Interval for Measurement
t=3.184
v=3
Fit Points = 5
Strain (ε)
Force (N)

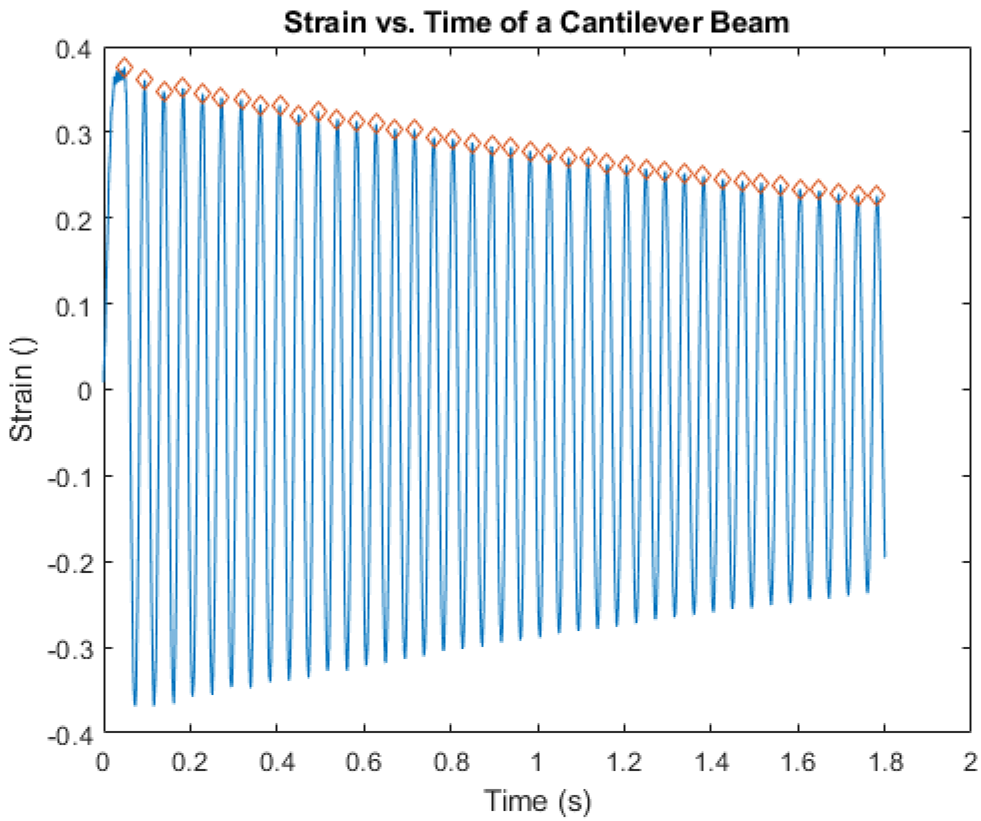


The hysteresis for the data was calculated to have a maximum value of 1.21% for the voltage measurement on the beam using different load values increasing then decreasing. The number was determined by calculating the difference between the corresponding loads as it was measured when the load was increasing to when it was decreasing. That difference was then divided by the total voltage difference from no load to the largest load. Please adhere to the table below that displays the data used for this calculation when increasing and decreasing load.

Displacement	Voltage	% Hysteresis
-0.025	0.0004	0.38
-0.05	0.0197	0.72
-0.075	0.0405	0.82
-0.1	0.0608	0.87
-0.125	0.0813	0.62
-0.15	0.1023	1.21
-0.175	0.1223	0.62
-0.2	0.1427	0.43
-0.225	0.164	0.87
-0.25	0.1859	0.43
0.225	0.1622	
-0.2	0.1418	
-0.175	0.121	
-0.15	0.0998	
0.125	0.08	
-0.1	0.059	
-0.075	0.0388	
-0.05	0.0182	
-0.025	-0.0004	
0	-0.001	

Part 2 – Cantilever Beam Vibration

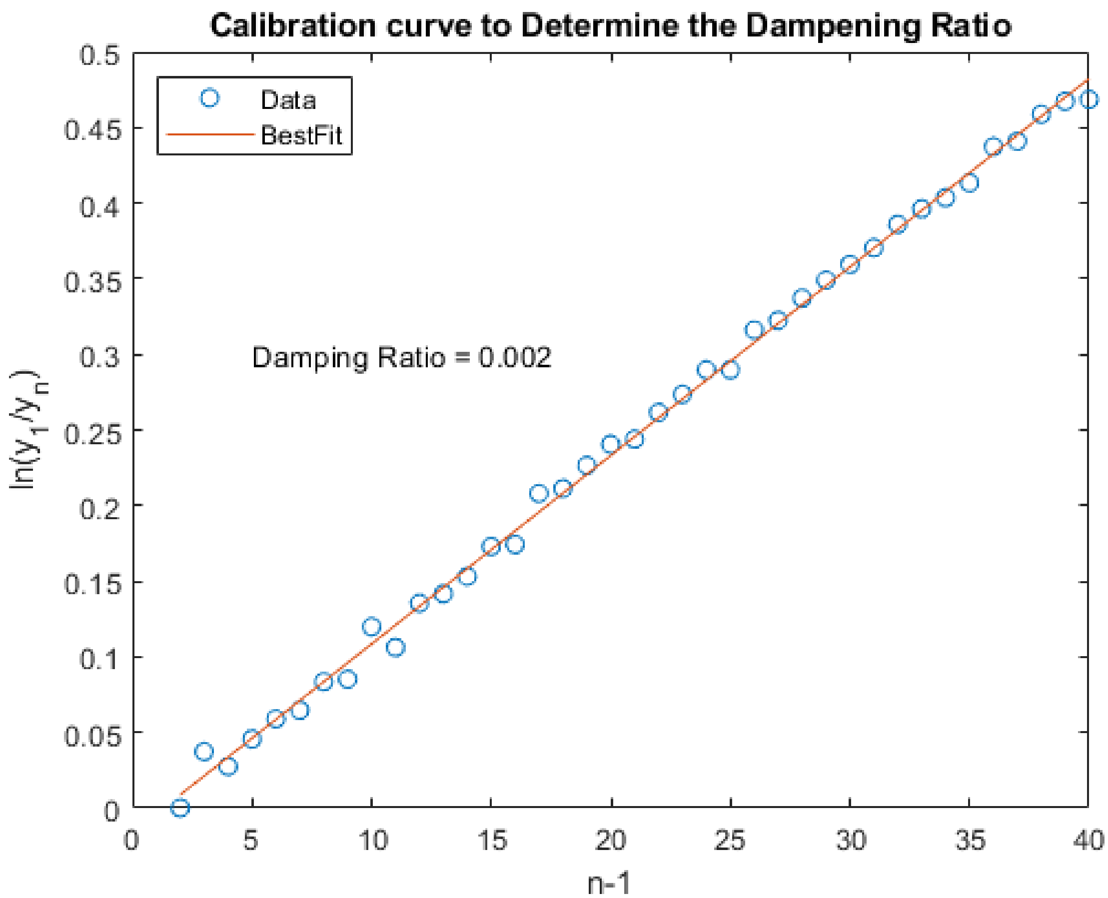
2a)



2b)

Damped Natural Frequency = 145 rad/s

2c)



2d)

	Peak 1	Peak 2	Peak 3	Peak 4	Peak 5	Peak 6	Peak 7	Peak 8	Peak 9	Peak 10
Damping Ratio	0	.003	.0015	.0018	.0018	.0019	.0017	.0019	.0017	.0021
	Peak 11	Peak 12	Peak 13	Peak 14	Peak 15	Peak 16	Peak 17	Peak 18	Peak 19	Peak 20
Damping Ratio	.0017	.0020	.0019	.0019	.0020	.0018	.0021	.0020	.0020	.0019
	Peak 21	Peak 22	Peak 23	Peak 24	Peak 25	Peak 26	Peak 27	Peak 28	Peak 29	Peak 30
Damping Ratio	.0019	.0020	.0020	.0020	.0019	.0020	.0020	.0020	.0020	.0020
	Peak 31	Peak 32	Peak 33	Peak 34	Peak 35	Peak 36	Peak 37	Peak 38	Peak 39	Peak 40
Damping Ratio	.0020	.0020	.0020	.0019	.0019	.0020	.0019	.0020	.0020	.0019

Average Damping Ratio = 0.0019

Standard Deviation of the Damping Ratio: 0.0002

Comparing the two methods to calculate the damping ratio, we can see that the difference between the two methods yield very similar results, only differing by around 5%.

2e)

The theoretical beam stiffness was calculated to be 267 N/m. The observed beam stiffness was 271 N/m. These values differ by only 2%.

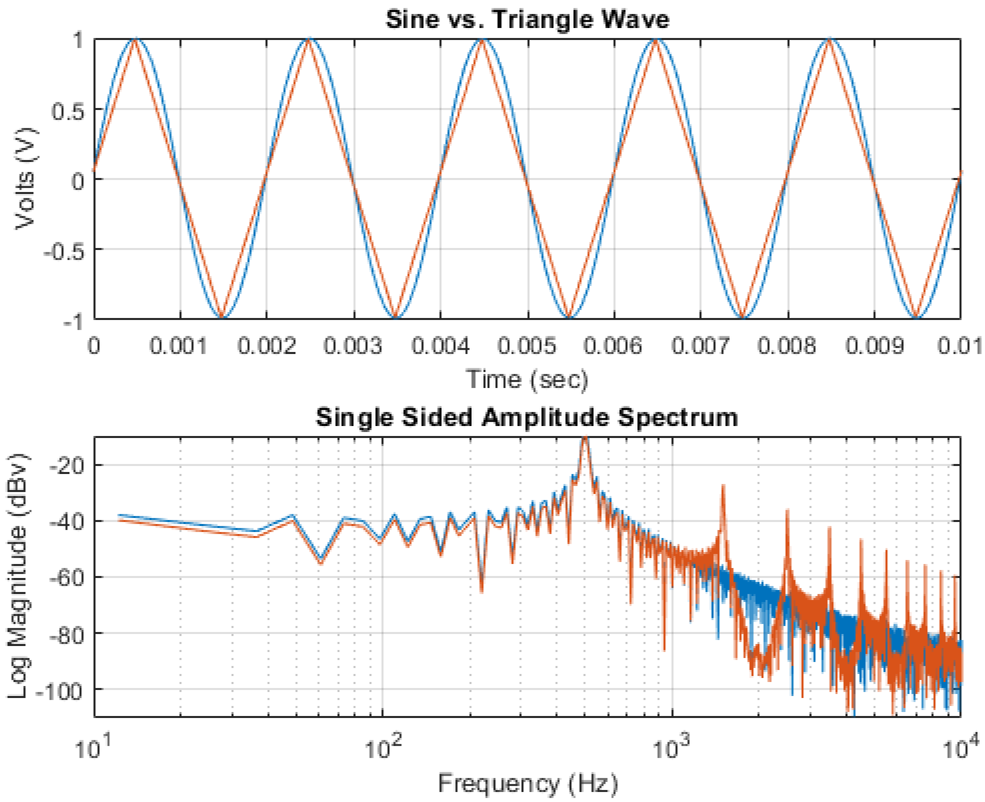
2f)

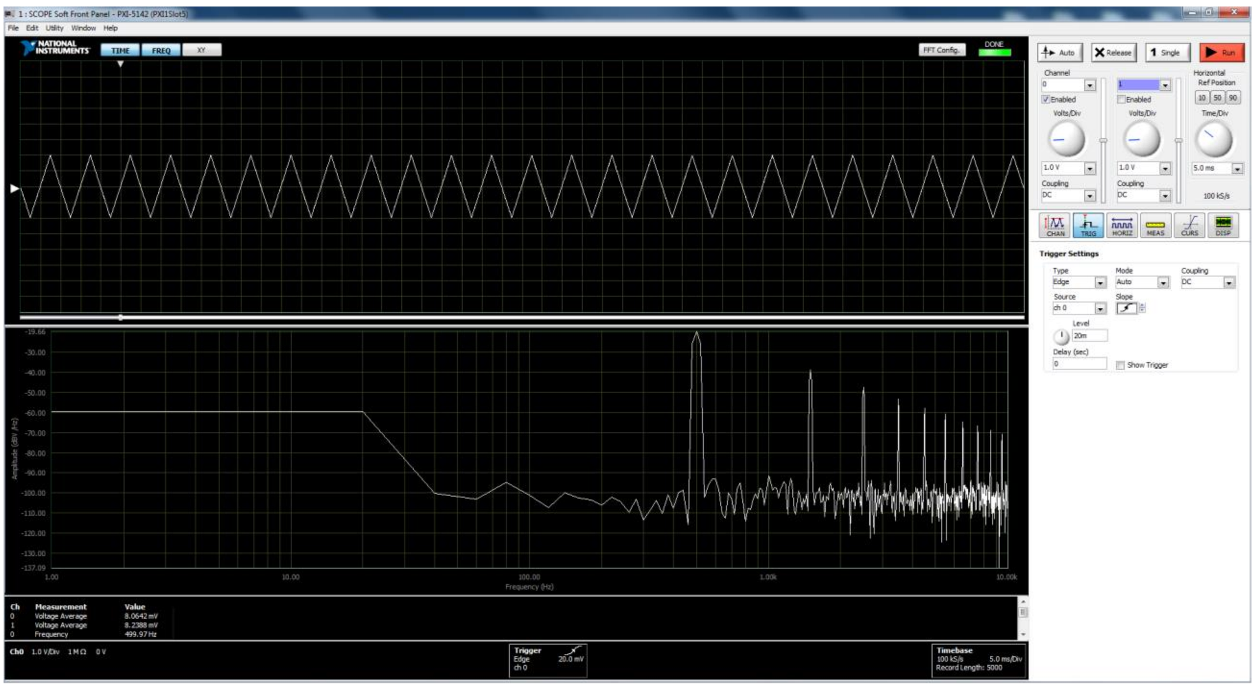
Using the beam equation to calculate the stiffness and using experimental data, a calculation of ω_n and ω_d can be calculated and are assumed to be nearly equal as the damping coefficient is low. The natural frequency from theoretical equations were calculated to be 145.8/s while the natural frequency obtained from observed data was 147/s. These values only differ by less than 1%!

2g)

The beam vibration has inherent damping in the system by just existing within the real world at sea level. The energy is dissipated as thermal energy and the natural change in momentum from air resistance with the speed of the beam against the atmospheric air.

Part 3 – Demonstrate Understanding of FFT





The above figures Illustrate the difference the sine and triangle wave forms obtained displayed. The frequency plotted together shows that the peaks are at the same frequencies!

3b)

Using the equation $\delta f = \frac{1}{N\delta t}$, a frequency resolution of 20/s is calculated. This also yields a minimal and maximum detectable frequency of 50,000/s and 20/s, respectively, using the equation $f_k = k\delta f$, with k being 1 and 2,500.

3c)

To adjust the frequency resolution to reach smaller frequencies, you can increase your time between data points. To increase the frequency you can detect, do the exact opposite. The number of data points collected can be manipulated as well with the same strategies and end results.

Part 4 – String Vibration

4a)

$F_{loadcell,max}$	S_{sens}	Gain	V_{out1}	V_{out2}	Length	Linear Density	N	V_{ex}
100 lbf	3,400 mv/V	1000	3 V	6 V	.73 m	6.8 g/m	1 to 4	5 V

From our setup, the above values are constants that are used to calculate the measured force the load cell recorded from its voltage output. The equation for this value is

$$F_{in} = \frac{V_{out} F_{loadcell,max}}{S_{sens} V_{ex} G}$$

With these values, we were then able to calculate the first resonate frequency of each string using the equation

$$Resonate\ Frequency\ (1\ to\ 4) = \frac{1}{2}L * \sqrt{\frac{F_{in}}{Linear\ Density}} * N$$

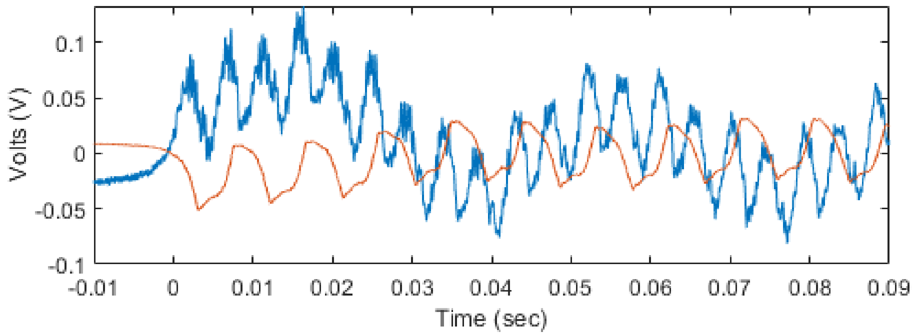
Using that equation, the first resonate frequency for the 3 Volt and 6 Volt output are

Number	Resonate Frequency 3V	Resonate Frequency 6V
1	.0194/s	.0275/s
2	.0388/s	.0549/s
3	.0583/s	.0824/s
4	.0777/s	.1099/s

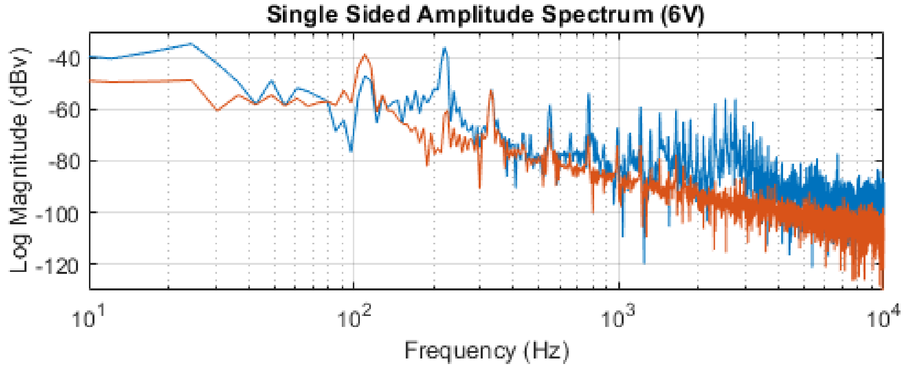
4b)

The below graphs are the amplitude vs frequency for each tension string with different time per division and amount of voltage input used.

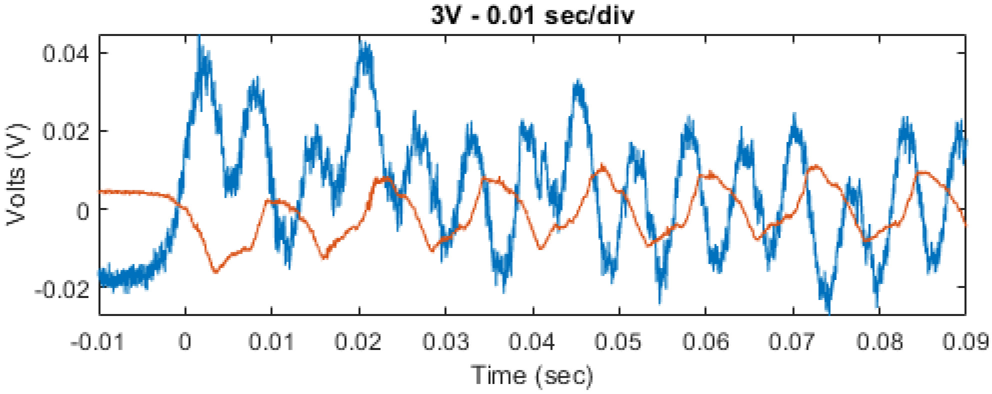
6V - 0.01 sec/div



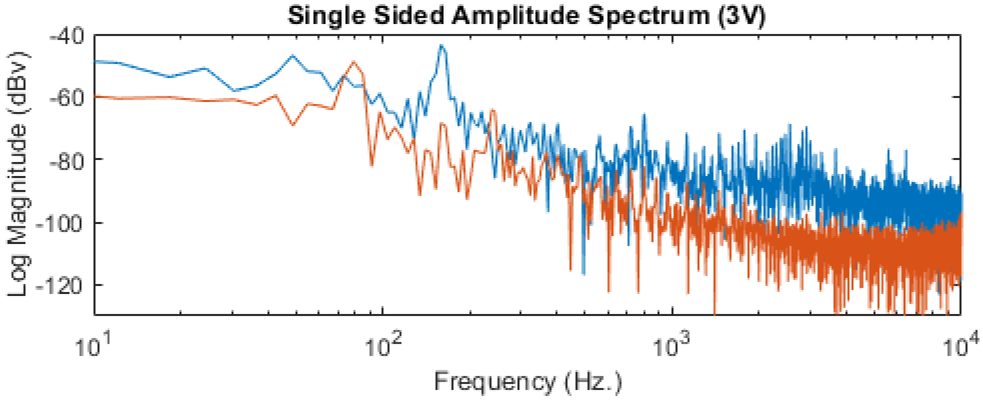
Single Sided Amplitude Spectrum (6V)

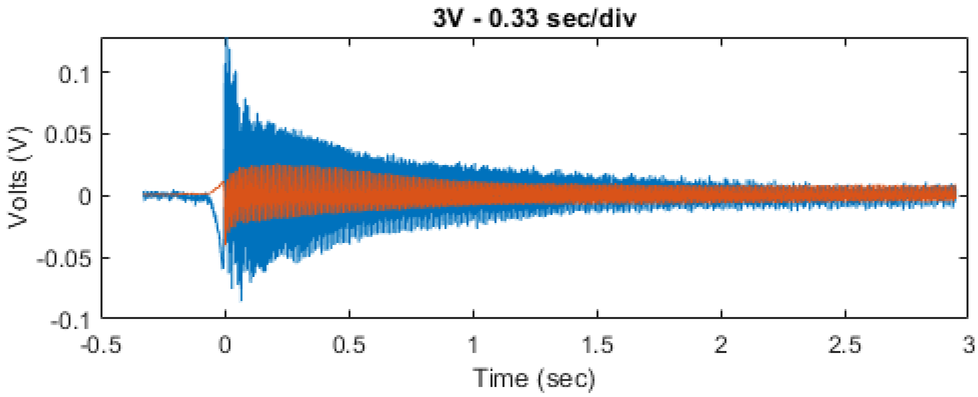


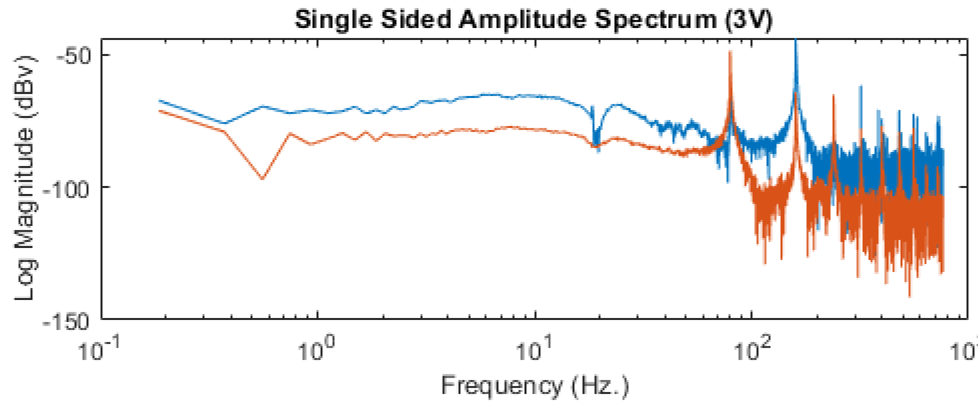
3V - 0.01 sec/div

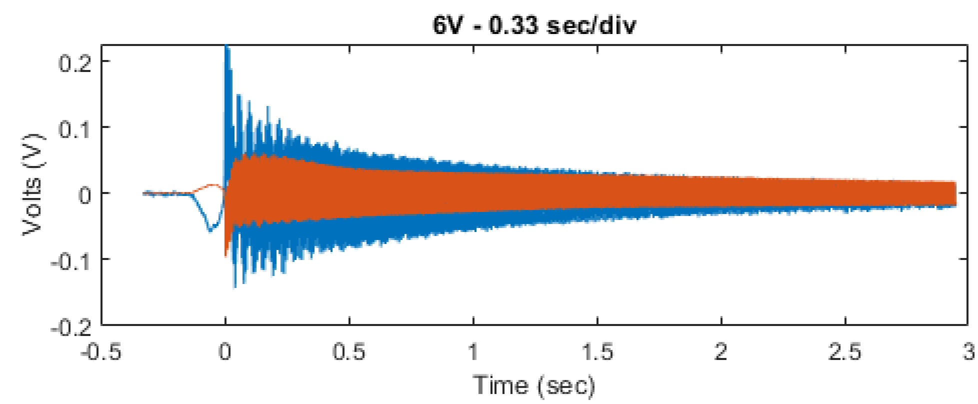


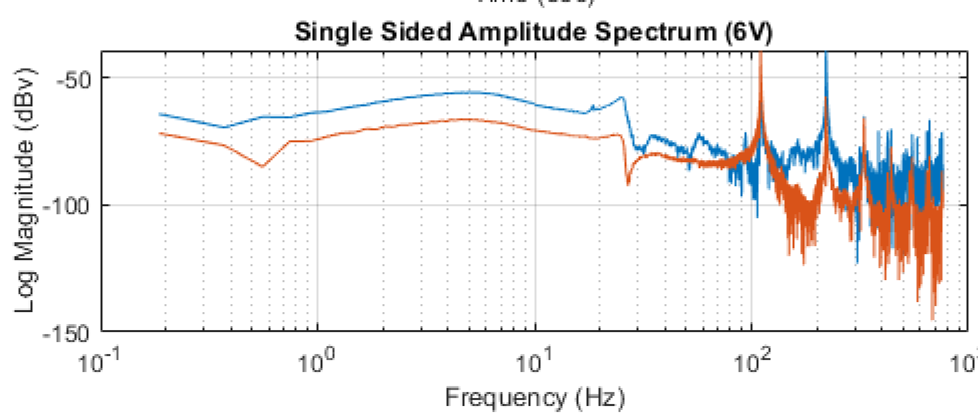
Single Sided Amplitude Spectrum (3V)




3V - 0.33 sec/div
Volts (V)
Time (sec)


Single Sided Amplitude Spectrum (3V)
Log Magnitude (dBv)
Frequency (Hz.)


6V - 0.33 sec/div
Volts (V)
Time (sec)


Single Sided Amplitude Spectrum (6V)
Log Magnitude (dBv)
Frequency (Hz)

4c)

The first four resonate frequencies for both different string tension are displayed in the tables below. The fundamental frequency is the one with the largest peak from imposing initial string displacement and releasing.

Observed Natural Frequencies: 3V

N	Piezoelectric	Strain-Gauge
1	79.7	152
2	160	298
3	238	322
4	319	445

Observed Natural Frequencies: 6V

N	Piezoelectric	Strain-Gauge
1	110	226
2	219	310
3	329	528
4	420	637

Calculated Natural Frequencies: 3V

N	Frequency
1	72.2
2	144.3
3	216.5
4	288.7

Calculated Natural Frequencies: 6V

N	Frequency
1	102.1
2	204.1
3	306.2
4	408.3

4d)

The two load cells used have different fundamental frequencies attached to them because when the string is moved from its equilibrium position, the piezoelectric sensor records the minimum frequency of the test because of its placement. The other load cells record purely the maximum. One different condition is if the string does not only oscillate sideways, but downwards, they both would experience and record maximum frequencies from the compressive force the cells would experience.

Appendix

```
% Lab 3
clear all
close all

%% Part 1

R_Shunt = [55700,67700,82100,99700,149000]; % Ohms
V_Shunt = [.260,.220,.182,.150,.099]; % Volts
R_Strain = 120; % Ohms
V_Gain = V_Shunt./100; % Accounting for Amplifier

% Constant for Calculating Delta R
A = R_Strain.*R_Shunt;
B = R_Strain + R_Shunt;

Delta_R = R_Strain - (A./B); % Equation for the change in R

[p1,s]=polyfit(Delta_R,V_Gain,1);
Fit_Data = p1(1)*Delta_R+p1(2);

Nu = length(Delta_R)-2; % First order
t_95 = tinv(0.975,Nu); % P = 95%
x_Bar = sum(Delta_R)/length(Delta_R); % Mean
Denom = sum((Delta_R-x_Bar).^2);
S_yx = (sum((V_Gain-Fit_Data).^2)/Nu).^(0.5); % Standard Error of Fit
Con_Fit = t_95*S_yx*(1/length(Delta_R)+(Delta_R-x_Bar).^2/Denom).^(0.5);
Con_Measure = t_95*S_yx*(1+1/length(Delta_R)+(Delta_R-
x_Bar).^2/Denom).^(0.5);

% Plotting
figure(1)
plot(Delta_R,V_Gain,'o',Delta_R,Fit_Data,'--
',Delta_R,Fit_Data+Con_Fit,'r',Delta_R,Fit_Data-
Con_Fit,'r',Delta_R,Fit_Data+Con_Measure,'b',Delta_R,Fit_Data-
Con_Measure,'b');
title('Resistance vs. Amplified Output Voltage')
xlabel('\Delta R')
ylabel('\Delta e\o')
ylim([0.8*10^-3,3*10^-3])
legend('Data Points','Bestfit line','Upper Confidence Interval for
Fit','Lower Confidence Interval for Fit','Upper Confidence Interval for
Measurement','Lower Confidence Interval for
Measurement','location','northwest')
text(.15,1.3e-03,strcat('\Delta e\o','  ','=','  ','0.0103(Ohms)+1.851x10^-
5(Volts)'))
text(.15,1.1e-03,'t-value=4.3027')
text(.15,.9e-03,'v=2')
```

```
E_i = 5; %Volts
G_F = 2.1; %Strain gauge factor
St = 2.*V_Gain./G_F*E_i;

% Strain vs. Load

Load = [.060,.110,.160,.210,.260] .* 9.81; % N
Load_Volts=[.0712-.000600,.130-.000250,.190-
.000150,.248+.000200,.306+.000300]; % Volts accounting for the Zero shifting
Beam_Length = 6.58*.0254; %Length of beam in meters
Thickness = 0.050*.0254; % Thickness of beam in meters
Width = .497*.0254; % meters
E = 1.93e11; % Elastic Modulus

Strain_P=(2.*Load_Volts./100)./(G_F*E_i);
Load_Theo=(12.*Load.*(Beam_Length-
Width)*(Thickness/2))./(E*(Width*Thickness^3));
[h1,s]=polyfit(Load,Strain_P,1);
Best_Fit_1=h1(1)*Load+h1(2);

figure(2)
plot(Load,Strain_P,'o',Load,Best_Fit_1,'--',Load,Load_Theo)
xlabel('Load (N)')
ylabel('Strain (\epsilon)')
title('Applied Load vs. Strain')
legend('Data','Best Fit Line','Theoretical Data','Location','Northwest')

% Statistic Calculations
Nu_1 = length(Load)-2; % First order
t_95_1 = tinv(0.975,Nu_1); % P = 95%
X_Bar_1 = sum(Load)/length(Load); % Mean
Denom_1 = sum((Load-X_Bar_1).^2);
s_yx_1 = (sum((Load_Theo-Best_Fit_1).^2)/Nu_1).^(0.5); % Standard Error
Calculation of the Fit
Con_Fit_1 = t_95_1*s_yx_1*(1/length(Load)+(Load-X_Bar_1).^2/Denom_1).^.5;
Con_Measure_1 = t_95_1*s_yx_1*(1+1/length(Load)+(Load-
X_Bar_1).^2/Denom_1).^.5;

% Plotting with Statistical Lines
figure(3)
plot(Load,Strain_P,'*',Load,Best_Fit_1,Load,Best_Fit_1+Con_Fit_1,'r',Load,Bes
t_Fit_1-Con_Fit_1,'r',Load,Best_Fit_1+Con_Measure_1,'b',Load,Best_Fit_1-
Con_Measure_1,'b')
xlabel('Force (N)')
ylabel('Strain (\epsilon)')
xlim([0.5,2.6])
title('Force vs. Strain with Confidence Intervals')
legend('Data','Bestfit','Upper Confidence Interval for Fit','Lower Confidence
Interval for Fit','Upper Confidence Interval for Measurement','Lower
Confidence Interval for Measurement','Location','Northwest')
text(1.5,2e-4,strcat('t',' ', '=' ,' ', '3.184'))
text(1.5,1.5e-4,'v=3')
text(1.5,1e-4,'Fit Points = 5')

% Strain vs. Displacement
```

```
Disp =
[0,.025,.050,.075,.100,.125,.150,.175,.200,.225,.250,.250,.225,.200,.175,.150
,.125,.100,.075,.050,.025,0]*0.0254; % meters
Disp_Volts=[.0004,.0197,.0405,.0608,.0813,.1023,.1223,.1427,.1640,.1859,.2070
,.2070,.1850,.1622,.1418,.1210,.0998,.0800,.0590,.0388,.0182,-.0004];
St_2=(2*Disp_Volts./100)./(G_F*E_i); % Accounting for amplifier
Disp_Theo=(3.*Disp.*(Thickness/2).*(Beam_Length-Width)./(Beam_Length^3));
[p2,s] = polyfit(Disp,St_2,1);
Best_Fit_2 = p2(1)*Disp+p2(2);




figure(4)
plot(Disp,St_2,'*',Disp,Best_Fit_2,'--',Disp,Disp_Theo)
title('Strain vs. Displacement')
xlabel('Displacement (m)')
ylabel('Strain (\epsilon)')
legend('Data','Bestfit','Theroretcial','Location','Northwest')

% Statistics
Nu_2 = length(Disp)-2; % First order
t_95_2 = tinv(0.975,Nu_2); %t value for P=95%
x_Bar_2 = sum(Disp)/length(Disp); % Mean
Denom_2 = sum((Disp-x_Bar_2).^2);
s_yx_2 = (sum((Disp_Theo-Best_Fit_2).^2)/Nu_2).^.5; %standard error of the
fit
Con_Fit_2 = (t_95_2*s_yx_2*(1/length(Disp)+(Disp-x_Bar_2).^2/Denom_2).^.5);
Con_Measure_2 = (t_95_2*s_yx_2*(1+1/length(Disp)+(Disp-
x_Bar_2).^2/Denom_2).^.5);

figure(5)
plot(Disp,St_2,'o',Disp,Best_Fit_2,'--
',Disp,Best_Fit_2+Con_Fit_2,'r',Disp,Best_Fit_2-
Con_Fit_2,'r',Disp,Best_Fit_2+Con_Measure_2,'b',Disp,Best_Fit_2-
Con_Measure_2,'b')
xlabel('Displacement (m)')
ylabel('Strain (\epsilon)')
title('Displacement vs. Strain Confidence')
legend('Data','Bestfit','Upper Confidence Interval for Fit','Lower Confidence
Interval for Fit','Upper Confidence Interval for Measurement','Lower
Confidence Interval for Measurement','Location','Northwest')
text(4e-03,1e-4,strcat('t' , '  ','=','  ','3.1824'))
text(4e-03,.66e-4,'v=20')
text(4e-03,.33e-4,'Fit Points = 11')


% Hysteresis Calculations
Increasing =
[0.0004,.0197,.0405,.0608,.0813,.1023,.1223,.1427,.1640,.1859,.2070];
Decreasing = [-
.0004,.0182,.0388,.0590,.0800,.0998,.1210,.1418,.1622,.1850,.2070];
Y_Range = Increasing(end)-Increasing(1);
Difference = Increasing-Decreasing;
```

```
Hysteresis_Max = (max(abs(Difference))./(Y_Range))*100;
Hyst = (Difference/Y_Range)*100



%% Part 2


% 2a

Header=29;
Data = importdata('Beam3.lvm','\t',Header);
Col1=Data.data(:,1);
Col2=Data.data(:,2);
for i=1:length(Col1)
    if Col1(i)>-0.005
        Base=i;
        break
    end
end

Base_Line = mean(Col2(1:Base));
Dev = std(Col2(1:Base));
Threshold = 5*Dev;

for i=1:length(Col1)
    if(abs(Col2(i)-Base_Line)>Threshold)
        Start_Time=i;
        break
    end
end

New_Time=Col1(Start_Time:length(Col1))-Col1(Start_Time);
New_Strain=Col2(Start_Time:length(Col1));
Constant=0.015;
[Locations,Values]=peakfinder(New_Strain,Constant);

figure(6)
plot(New_Time,New_Strain,New_Time(Locations),Values,'d')
xlabel('Time (s)')
ylabel('Strain ()')
title('Strain vs. Time of a Cantilever Beam')


% 2b
K = 3*E*Width*Thickness^3;
Denom_3=12*Beam_Length^3;
K_Beam = K/Denom_3;
Density_Steel = 7700; % kg/m^3
Mass_Hook = 0.0074; % kg
Volume_Beam=Beam_Length*Thickness*Width;
Mass_Beam = Density_Steel*Volume_Beam;
Mass_Effective = Mass_Beam/4 + Mass_Hook;
Damped_Natural_Frequency = sqrt(K_Beam/Mass_Effective); % rad/s
```

```matlab
% 2c
Log = log(Values(2)./Values(2:length(Values)));
Con = transpose(2:length(Values));
[mb,Time_Wave] = polyfit(Con,Log,1);
Best_Fit_3 = mb(1)*Con+mb(2);
alpha = mb(1);
Damp_Ratio = (alpha/sqrt(4*pi^2+alpha^2));


figure(7)
plot(Con,Log,'o',Con,Best_Fit_3)
xlabel('n-1')
ylabel('ln(y_1/y_n)')
title('Calibration curve to Determine the Dampening Ratio')
text(5,.3,strcat('Damping Ratio = 0.002'))
legend('Data','BestFit','Location','Northwest')




% 2d
% Using Large Equation
for n=3:length(Values)
    Num = (1/(n-1))*log(Values(2)./Values(n));
    Damp(n) = Num./sqrt(4*pi^2+Num^2);
end

Damp_Mean = sum(Damp(3:end)./length(Damp(2:end)));
Damp_Std = std(Damp(3:length(Damp)));



%2e
k_Theo = (3*E*((Width*Thickness^3)/12))/Beam_Length^3;
k_Exp = p2(1)/h1(1)

%2f
Natural_Freq_Theo = sqrt(k_Theo/Mass_Effective);
Natural_Freq_Actu = sqrt(k_Exp/Mass_Effective);



%% Part 3

Header=29;
Sin_Wave = importdata('sinAvsF.lvm','\t',Header);
Tri_Wave = importdata('triAvsF.lvm','\t',Header);

Time_Wave = Sin_Wave.data(:,1);   % Shared Time Vector
Sin_Voltage = Sin_Wave.data(:,2);
Tri_Voltage = Tri_Wave.data(:,2);
```

```
% Plotting


T = Time_Wave(2)-Time_Wave(1); % Calculated time interval between data points
Freq = 1/T; % Sampling Frequency
Length = size(Sin_Voltage); % Number of points in vector
Length_2 = length(Sin_Voltage);
Power_2 = 2^nextpow2(Length(1)); % power of 2 from length y
Sin_FFT = fft(Sin_Voltage,Power_2)./Length(1); % fft bull
Tri_FFT = fft(Tri_Voltage,Power_2)./Length(1); % fft bull
Spaced_Points = Freq/2*linspace(0,1,Power_2/2+1); % spaced point vector using
length
Sin_FFT_A = 20*log10(abs(Sin_FFT(1:Power_2/2+1))); % amplitude
Tri_FFT_A = 20*log10(abs(Tri_FFT(1:Power_2/2+1))); % amplitude

figure(8)

subplot(2,1,1)
plot (Time_Wave,Sin_Voltage,Time_Wave,Tri_Voltage); grid
title('Sine vs. Triangle Wave')
xlabel ('Time (sec)')
ylabel ('Volts (V)')
xlim([0,.01])

subplot(2,1,2)
semilogx(Spaced_Points,Sin_FFT_A,Spaced_Points,Tri_FFT_A);grid  % abs(Y) =
(Re(Y)^2 + Im(Y)^2)^1/2
axis([10 10000 -110 -10])
title('Single Sided Amplitude Spectrum')
xlabel ('Frequency (Hz)')
ylabel ('Log Magnitude (dBv)')



% 3b
Freq_Reso = 1/(Length_2*T);


%% Part 4

Header = 32;
Guitar_Data_3v = importdata('guitar1.lvm','\t',Header);
Guitar_Data_6v = importdata('guitar6v1.lvm','\t',Header);
Time_Guitar = Guitar_Data_3v.data(:,1);
y1 = Guitar_Data_3v.data(:,2);
y2 = Guitar_Data_3v.data(:,4);
y3=Guitar_Data_6v.data(:,2);
y4=Guitar_Data_6v.data(:,4);

T1 = Time_Guitar(2)-Time_Guitar(1);         % Time per sample
Freq1 = 1/T1;                 % Sampling frequency
Length1 = size(y1);            % Length of signal - # of points
Power_2_1 = 2^nextpow2(Length1(1)); % Next power of 2 from length of y - need
for FFT
```

```
Guitar_FFT_3_1 = fft(y1,Power_2_1)./Length1(1); % this is a vector with
complex number elements
Guitar_FFT_3_2 = fft(y2,Power_2_1)./Length1(1); % this is a vector with
complex number elements
Space_Points_1 = Freq1/2*linspace(0,1,Power_2_1/2+1); % linspace generates
linearly spaced points
Guitar_3_1_A = 20*log10(abs(Guitar_FFT_3_1(1:Power_2_1/2+1)));
Guitar_3_2_A = 20*log10(abs(Guitar_FFT_3_2(1:Power_2_1/2+1)));

figure(9)

subplot(2,1,1)
plot(Time_Guitar,y1,Time_Guitar,y2)
title('3V - 0.01 sec/div')
xlabel ('Time (sec)')
ylabel ('Volts (V)')

subplot(2,1,2)
semilogx(Space_Points_1,Guitar_3_1_A,Space_Points_1,Guitar_3_2_A);
axis([10 10000 -130 -40])
title('Single Sided Amplitude Spectrum (3V)')
xlabel ('Frequency (Hz.)')
ylabel ('Log Magnitude (dBv)')



Length2 = size(y3);           % Length of signal - # of points
Power_2_2 = 2^nextpow2(Length2(1)); % Next power of 2 from length of y - need
for FFT
Guitar_FFT_6_1 = fft(y3,Power_2_2)./Length2(1); % this is a vector with
complex number elements
Guitar_FFT_6_2 = fft(y4,Power_2_2)./Length2(1); % this is a vector with
complex number elements
Space_Points_2 = Freq1/2*linspace(0,1,Power_2_2/2+1); % linspace generates
linearly spaced points
Guitar_6_1_A = 20*log10(abs(Guitar_FFT_6_1(1:Power_2_2/2+1)));
Guitar_6_2_A = 20*log10(abs(Guitar_FFT_6_2(1:Power_2_2/2+1)));

figure(10)
subplot(2,1,1)
plot(Time_Guitar,y3,Time_Guitar,y4)
title('6V - 0.01 sec/div')
xlabel ('Time (sec)')
ylabel ('Volts (V)')

subplot(2,1,2)
semilogx(Space_Points_2,Guitar_6_1_A,Space_Points_2,Guitar_6_2_A);grid  %
abs(Y) = (Re(Y)^2 + Im(Y)^2)^1/2
axis([10 10000 -130 -30])
title('Single Sided Amplitude Spectrum (6V)')
xlabel ('Frequency (Hz)')
ylabel ('Log Magnitude (dBv)')
```

```
%% Part 4 2

Header = 32;
Guitar_Data_3v = importdata('guitar2.lvm','\t',Header);
Guitar_Data_6v = importdata('guitar6v2.lvm','\t',Header);
Time_Guitar = Guitar_Data_3v.data(:,1);
y1 = Guitar_Data_3v.data(:,2);
y2 = Guitar_Data_3v.data(:,4);
y3=Guitar_Data_6v.data(:,2);
y4=Guitar_Data_6v.data(:,4);

T1 = Time_Guitar(2)-Time_Guitar(1);          % Time per sample
Freq1 = 1/T1;                 % Sampling frequency
Length1 = size(y1);           % Length of signal - # of points
Power_2_1 = 2^nextpow2(Length1(1)); % Next power of 2 from length of y - need
for FFT
Guitar_FFT_3_1 = fft(y1,Power_2_1)./Length1(1); % this is a vector with
complex number elements
Guitar_FFT_3_2 = fft(y2,Power_2_1)./Length1(1); % this is a vector with
complex number elements
Space_Points_1 = Freq1/2*linspace(0,1,Power_2_1/2+1); % linspace generates
linearly spaced points
Guitar_3_1_A = 20*log10(abs(Guitar_FFT_3_1(1:Power_2_1/2+1)));
Guitar_3_2_A = 20*log10(abs(Guitar_FFT_3_2(1:Power_2_1/2+1)));

figure(11)

subplot(2,1,1)
plot(Time_Guitar,y1,Time_Guitar,y2)

title('3V - 0.33 sec/div')
xlabel ('Time (sec)')
ylabel ('Volts (V)')

subplot(2,1,2)
semilogx(Space_Points_1,Guitar_3_1_A,Space_Points_1,Guitar_3_2_A);

title('Single Sided Amplitude Spectrum (3V)')
xlabel ('Frequency (Hz.)')
ylabel ('Log Magnitude (dBv)')



Length2 = size(y3);           % Length of signal - # of points
Power_2_2 = 2^nextpow2(Length2(1)); % Next power of 2 from length of y - need
for FFT
Guitar_FFT_6_1 = fft(y3,Power_2_2)./Length2(1); % this is a vector with
complex number elements
Guitar_FFT_6_2 = fft(y4,Power_2_2)./Length2(1); % this is a vector with
complex number elements
Space_Points_2 = Freq1/2*linspace(0,1,Power_2_2/2+1); % linspace generates
linearly spaced points
Guitar_6_1_A = 20*log10(abs(Guitar_FFT_6_1(1:Power_2_2/2+1)));
Guitar_6_2_A = 20*log10(abs(Guitar_FFT_6_2(1:Power_2_2/2+1)));
```

```
figure(12)
subplot(2,1,1)
plot(Time_Guitar,y3,Time_Guitar,y4)

title('6V - 0.33 sec/div')
xlabel ('Time (sec)')
ylabel ('Volts (V)')

subplot(2,1,2)
semilogx(Space_Points_2,Guitar_6_1_A,Space_Points_2,Guitar_6_2_A);grid  %
abs(Y) = (Re(Y)^2 + Im(Y)^2)^1/2

title('Single Sided Amplitude Spectrum (6V)')
xlabel ('Frequency (Hz)')
ylabel ('Log Magnitude (dBv)')




%% ye
Linear_Density=.00608; %g/m
L=.79;
Ffs = 448; %N 100 lb load cell
Gain = 1000;
mv_V = 3.4; %mV/V
vout3 = 3000;
vout6 = 6000;
N=[1:1:4];

F_in_3V=(vout3*Ffs)/(mv_V*5*Gain);  %tension in string
F_in_6V=(vout6*Ffs)/(mv_V*5*Gain);  %tension in string



Rfreq1=(1/(2*L))*(sqrt(F_in_3V/Linear_Density)).*N;
Rfreq2=(1/(2*L))*(sqrt(F_in_6V/Linear_Density)).*N;
```