Simon Popecki

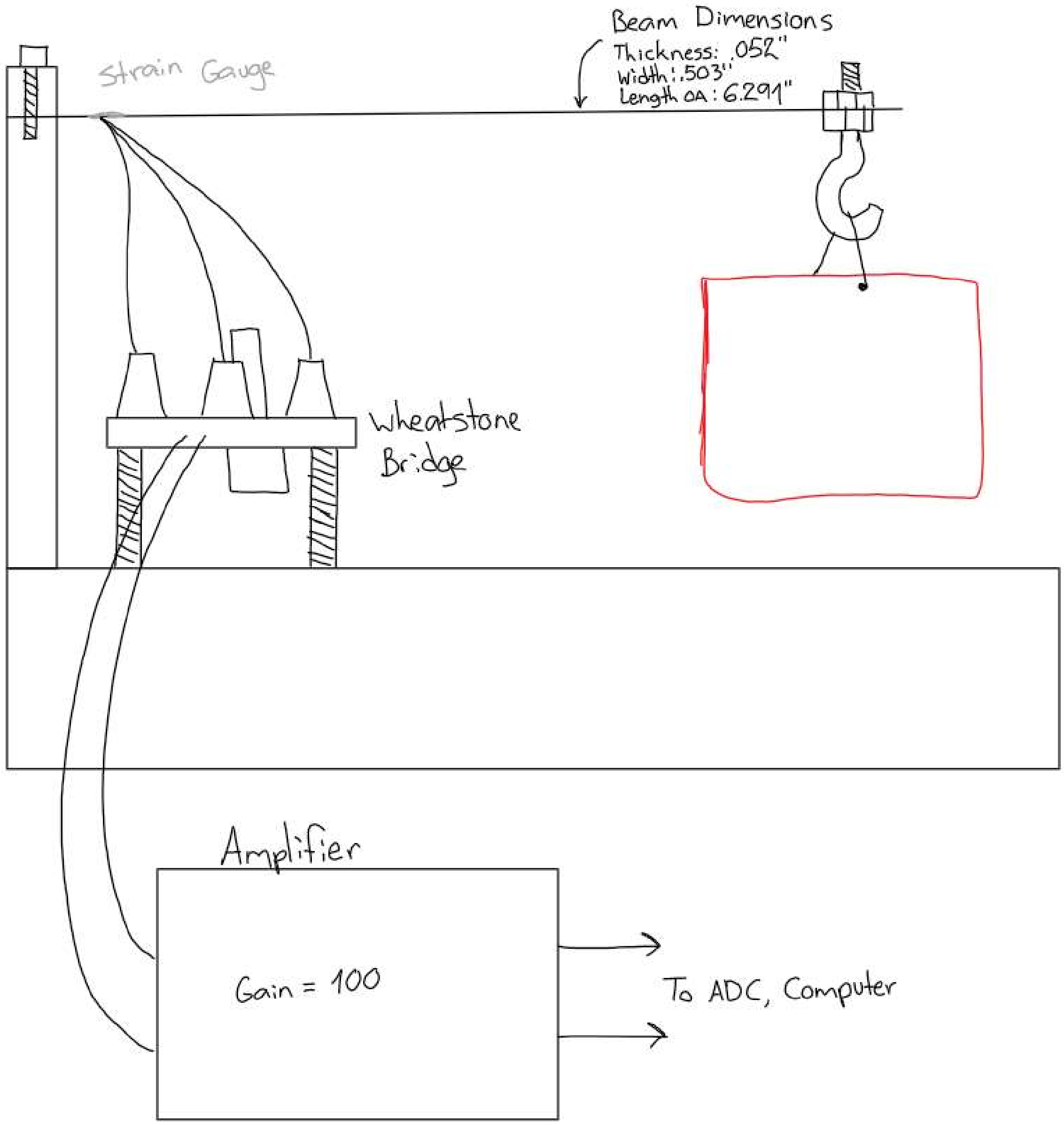
5, March 2017

ME 646

LAB 3 – STRAIN GAUGES AND VIBRATION

BEAM AND STRAIN GAUGE RESPONSE

Diagram of the Physical Setup:



Determination of Bridge Sensitivity Using Shunt Resistors:

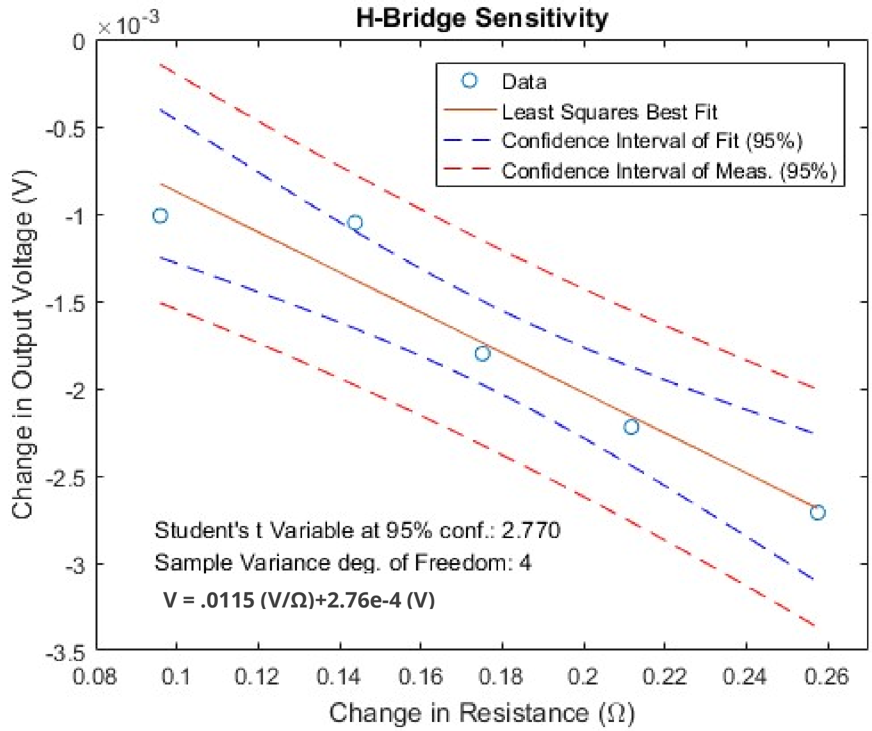


Figure 1 – Experimental response of the Wheatstone bridge. The large confidence interval was caused by an anomalous point, data for subsequent portions of this experiment will be borrowed from Jesse Feng.

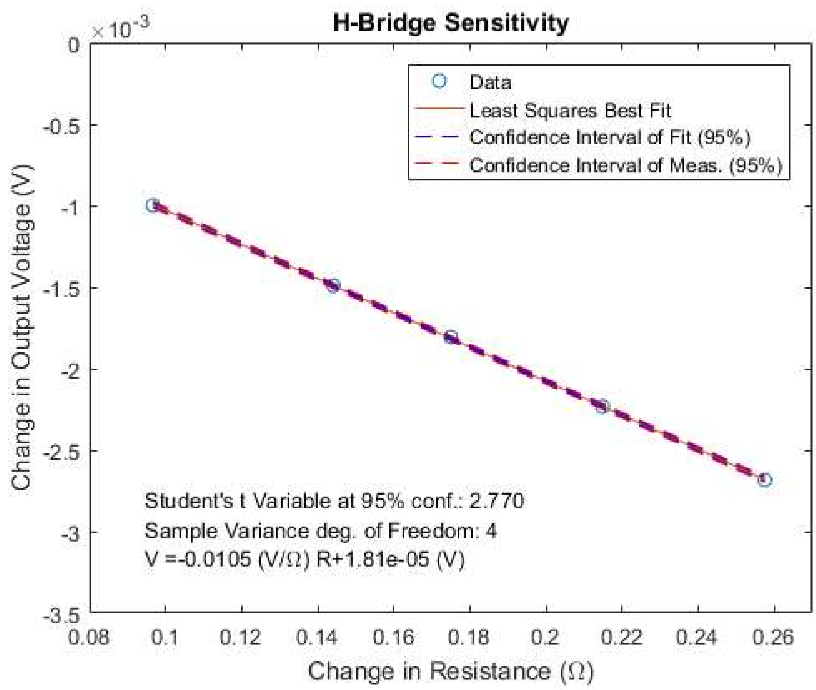


Figure 2 – The data of Figure 1 were replaced with cleaner data.

Comparison of Measured Bridge Sensitivity with Calculated Bridge Sensitivity

The measurements of bridge sensitivity were taken by adding a shunt resistor to one of the legs of the Wheatstone bridge. The calculations of the bridge sensitivity are made using the quarter Wheatstone bridge model. The voltage supplied to the Wheatstone bridge was 5 V – this value is represented by E_i in the equations below. E_o represents the output voltage across the Wheatstone bridge (see equation 1).

$$E_o = E_i\left(\frac{R_1}{R_1+R_2} - \frac{R_3}{R_3+R_4}\right) \quad (1)$$

Since only one resistor is being changed in value, the following is true:

$$R_2 = R_3 = R_4 = R \quad (2) \qquad\qquad R_1 = R + \Delta R \quad (3)$$

With the introduction of changes in voltage and resistance the equation for output voltage of the Wheatstone bridge is shown as:

$$E_o + \Delta E_o = E_i\left(\frac{R+\Delta R}{2R+\Delta R} - \frac{1}{2}\right) = E_i\left(\frac{2(R+\Delta R)-(2R+\Delta R)}{2(2R+\Delta R)}\right) = E_i\left(\frac{\Delta R}{(4R+2\Delta R)}\right) \quad (4)$$

The input voltage is approximated as:

$$E_i = \left(\frac{\Delta R}{4R}\right) \quad (5)$$

If the output voltage is assumed to be zero at the point where there is no change in resistance. Strain is represented by ε, and gauge factor is represented by GF.

$$\frac{\Delta E_o}{E_i} \approx \left(\frac{\Delta R}{4R}\right) = \frac{GF\varepsilon}{4} \quad (6)$$

Solving for the bridge sensitivity by rearranging equation 6:

$$\frac{\Delta E_o}{\Delta R} \approx \frac{E_i}{4R} \rightarrow \frac{5\,V}{4(120\,\Omega)} = .0104\,V/\Omega \quad (7)$$

The experimental bridge sensitivity had an error less than 1% using the data of Jesse Feng. Using the original data, the error percentage was 11%.

Solving for Strain Given Gauge Factor and Amplifier Output

The experimental Wheatstone bridge was a half bridge – there were two strain gauges and two static 120 Ω resistors.

$$\frac{\Delta E_o}{E_i} = \frac{GF}{4}(\varepsilon_1 - \varepsilon_2 - \varepsilon_3 + \varepsilon_4) \tag{8}$$

Strains three and four are equal to zero (since there are only two strain gauges – one on top of the beam and one below the beam). Strains one and two are equal to the bending strain plus the thermal strain (which is comparatively small) strain one is roughly equal in magnitude to strain two if they are on opposing sides of the beam. Equation 8 reduces to:

$$\frac{\Delta E_o}{E_i} = \frac{GF\varepsilon}{2} \tag{9}$$

Rearranging equation 9, and accounting for gain allows for the isolation of strain, which can now be solved for given a change in output voltage.

$$\varepsilon_b = \frac{\Delta E_o * 2 * Gain}{E_i * GF} \tag{10}$$

Determining Strain vs. Load and Strain vs. Displacement

The strain in a beam can be calculated by equation 11.

$$\varepsilon = \frac{F * length * (\frac{Thickness}{2})}{E(\frac{Width * (Thickness)^3}{12})} \tag{11}$$

Where 'E' is the Young's modulus, 'Thickness' is the thickness of the beam (smallest dimension measured), and 'Width' is the width of the beam (second smallest dimension measured). Length is the distance between the hook and strain gauges. Force can be determined either through equation 12 (mass) or through equation 13 (displacement).

$$F = mg \tag{12}$$

$$F = \frac{(deflection)3E(\frac{Width * (Thickness)^3}{12})}{Length^3} \tag{13}$$

Where m is the mass suspended by the beam, g is the gravitational acceleration constant on Earth, and deflection is the amount the beam was displaced by the micrometer.

Substituting equation 13 into equation 11 causes the moment of inertia and Young's modulus to cancel, yielding:

$$\varepsilon = \frac{3(Deflection)(Thickness}{2(Length)^2} \tag{14}$$

The comparison between the values recorded by the strain gauges and the predictions derived from Mass (Figure 3) and from displacement (Figure 4) is shown below. Figure 5 shows the linear least squares 95% confidence intervals of measurement and fit.

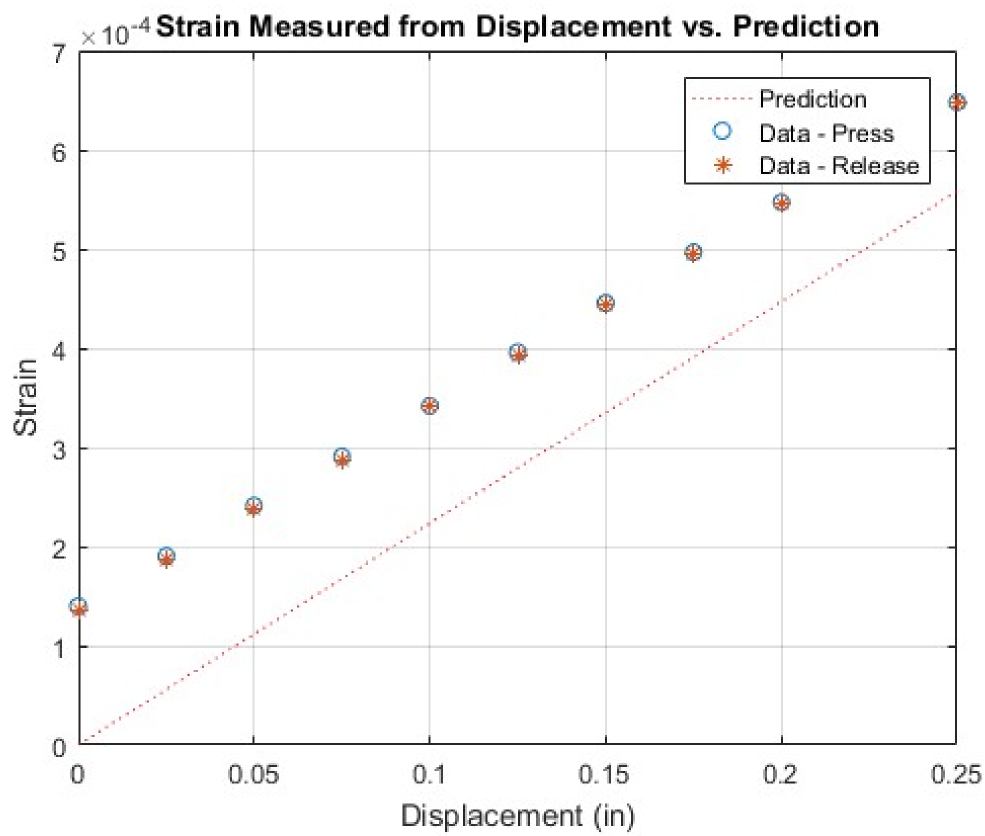


Figure 3 – Strain calculated from displacement compared to data recorded by the strain gauges

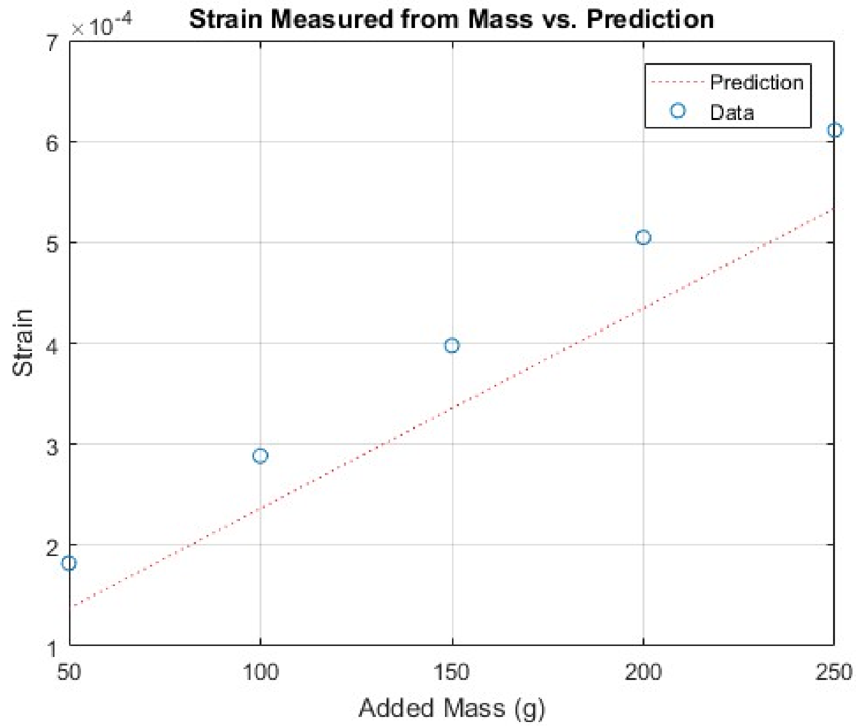


Figure 4 – Strain measured by adding mass to the end of the beam vs. strain gauge measurement

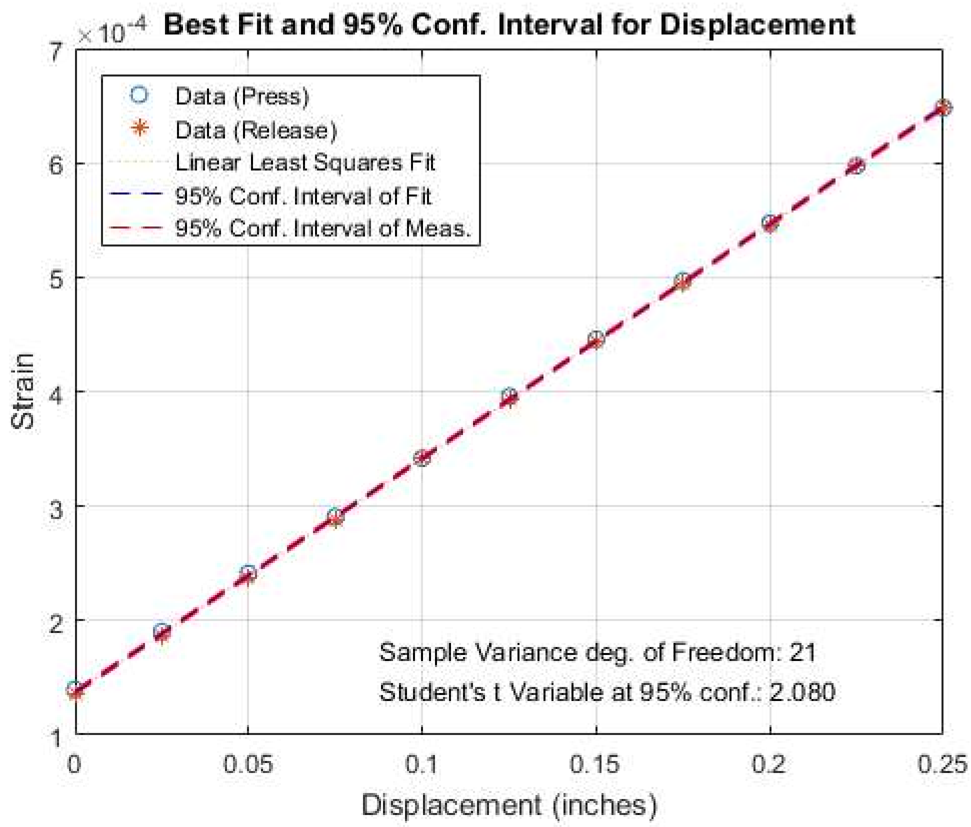


Figure 5 – The data above show a relatively narrow 95% confidence interval.

Determination of Hysteresis for the Displacement Measurements

Hysteresis is the difference between the y values f(x) for any function of x. In this lab the hysteresis will be the largest value measured from the strain during beam depression (downwards) by the micrometer, minus the strain at the corresponding displacement from the beam's return to its starting position. Table 1 shows the difference between each *point* corresponding to a specific displacement, as well as the total hysteresis.

Total Hysteresis (ε):	5.122 x10^{-4}	
Displacement (inches):	Press – Release (ε x10^{-4}):	Hysteresis (ε x10^{-4}):
0	1.398 – 6.482	-5.084
.025	1.904 – 5.978	-4.074
.050	2.412 – 5.454	-3.042
.075	2.908 – 4.942	-2.034
.100	3.420 – 4.434	-1.014
.125	3.958 – 3.930	.0280
.150	4.456 – 3.416	1.040
.175	4.968 – 2.872	2.096
.200	5.472 – 2.370	3.102
.225	5.974 – 1.868	4.106
.250	6.482 – 1.360	5.122

Table 1

Harmonics of the Beam:

The beam was pulled back and let go- allowed to resonate. The subsequent oscillations were recorded, the data were trimmed such to start at 0 seconds, and normalized such that the zero volts are produced at the resting position of the beam. The waveform can be seen in Figure 6 (below).

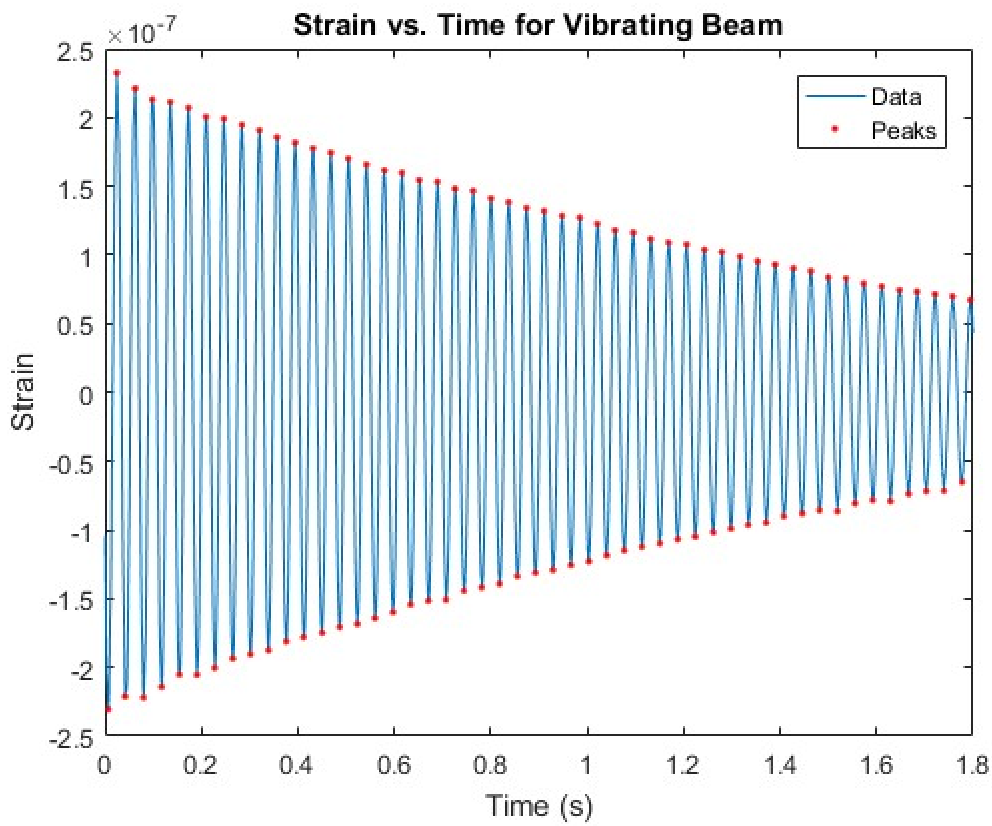


Figure 6 – The waveform of the beam vibration

The damping ratio of the beam was calculated using two different methods. The first method used least squares to determine a single damping ratio value ξ, which was found to be .0041. The least squares method is derived from equation 15.

$$\ln\left(\frac{y_1}{y_n}\right) = \left(\frac{2\pi\xi}{\sqrt{1-\xi^2}}\right)(n-1) \quad (15)$$

The second method used equation 16, and produced an array of values.

$$\xi = \frac{\frac{1}{n-1}\ln\left(\frac{y_1}{y_n}\right)}{\sqrt{4\pi^2+\left[\frac{1}{n-1}\ln\left(\frac{y_1}{y_n}\right)\right]^2}} \quad (16)$$

The second method's results are shown below in Table 2. The damped natural frequency was found to be 165.3470 rad/s. The undamped natural frequency was found to be 165.3483 rad/s. These values are very close – the only damping present was the viscosity of air and the internal friction of the steel bar.

Average ξ : .0040	
Standard Deviation of ξ : 9.49×10^{-4}	
ξ Values (Peaks in ascending order from top left to bottom right – e.g. first peak is 0 last peak is .0041⋯)**:**	
0	0.00377858
0.00792578	0.00369839
0.00694509	0.00377963
0.00507497	0.00386316
0.00460476	0.00380683
0.00472383	0.0039008
0.00411706	0.0038961
0.00404618	0.00383685
0.00393465	0.00389408
0.00398214	0.00385965
0.00392523	0.0038971
0.00390827	0.00394379
0.00381773	0.00394874
0.00383253	0.00396374
0.003854	0.00396093
0.00387075	0.00407046
0.00373783	0.00401506
0.00383384	0.00409199
0.00369097	0.00409366
0.00377688	0.0041298
0.00367551	0.00409081
0.00378715	0.00409292
0.0037558	0.00408467
0.00380678	0.00411787
0.00376281	

Table 2

Comparing Calculated and Experimental Beam Stiffness

The stiffness of the beam was calculated knowing stiffness is equal to the force on the beam divided by beam displacement. The equation for beam stiffness (equation 19) was derived from equations 17 and 18 and the previously stated relationship. Where I is the moment of inertia of the rectangular beam, l is the length of the beam, E is the Young's modulus of steel, and Y is the distance from the center of the beam to the top of the beam.

$$\sigma = \frac{My}{I} = E\varepsilon \tag{17}$$

$$\delta = \frac{Fl^3}{3EI} \tag{18}$$

$$\frac{F}{\delta} = K = \frac{3EI}{l^3} = \frac{3*200\ GPa*(\frac{.0128*.0013^2}{12})}{(.1501)^3} \tag{19}$$

The calculated stiffness was found to be 435.35 N/m using equation 19. The stiffness was also calculated experimentally – the voltage outputs of the force (mass) driven measurements were compared to the voltage outputs of the displacement driven measurements. Linearly interpolated arrays were used to increase the resolution of the force driven voltage measurements. Using the relationship between force and voltage, and voltage and displacement, a point of interest could be solved for giving the force applied for a given displacement. The stiffness figure from the interpolated array method was 535.65 N/m, a figure within 19% error of the calculated value.

Comparison Between Stiffness, Effective Mass, and the Natural Frequency

The mass of the steel beam was estimated by multiplying the volume by the density of steel. The mass of the beam was calculated to be 20.4 grams. The mass of the hook was given as 7.4 grams. The effective mass is one quarter of the total mass. The effective mass and calculated/experimental stiffnesses are substituted into equation 20.

$$\omega_n = \sqrt{\frac{K}{m_{eff}}} \tag{20}$$

The natural frequency derived from the experimental value for K (535.65 N/m) was found to be **139.8 rad/s**. The natural frequency using the calculated value for K (435.35) was found to be **126.1 rad/s**. The expected damped/undamped natural frequency (found by analyzing the waveform) was **165.3 rad/s**. The error percentages for the experimental and calculated stiffness were 15% and 24% respectively. The unaccounted for damping of the system was likely caused by air resistance or the internal friction of the beam.

Fast Fourier Transformations Using Function Generator

During the lab period a sine wave and triangle wave were recorded – the frequency of both waves was set at 500 Hz using, and generated using an oscilloscope. Screen shots from the National Instruments software are shown below (Figures 7, 8).

Figure 5 – A 500 Hz sine wave and corresponding Fourier Transform. The Fourier peak is at 500 Hz,

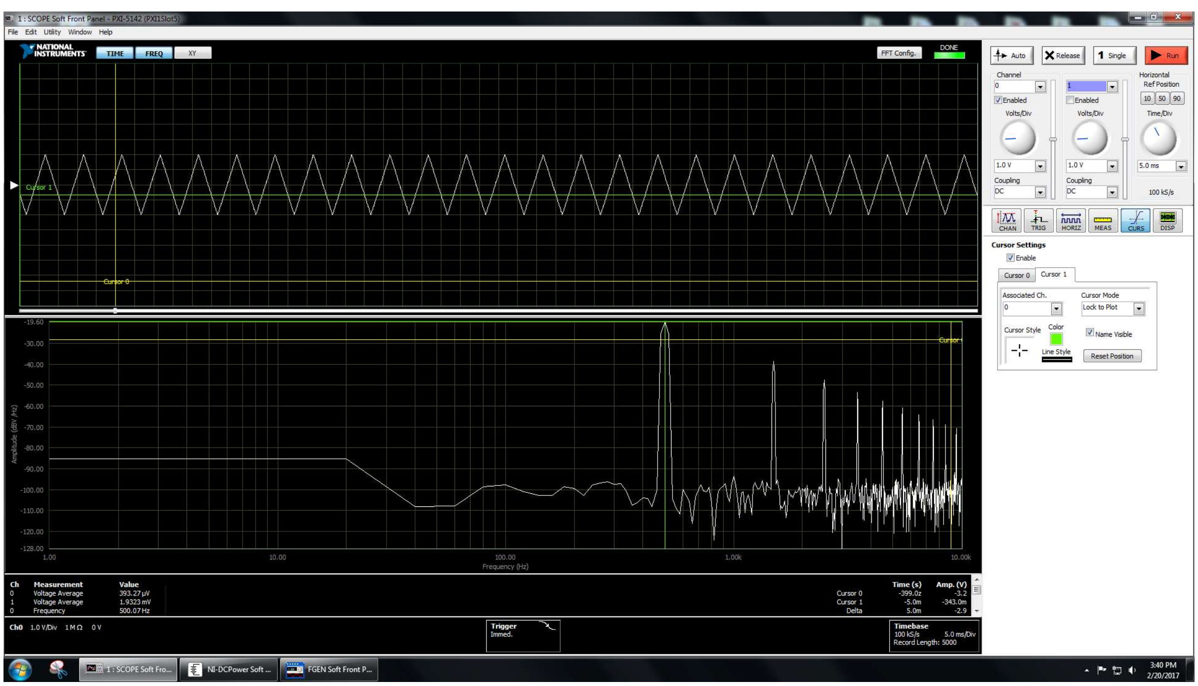


Figure 8 – A 500 Hz triangle wave and corresponding Fourier transform. The Fourier peak is at 500 Hz.

Figure 9 (below) shows a MATLAB Fourier transform of the same data. The sine wave is plotted in blue and the triangle wave is plotted in orange.

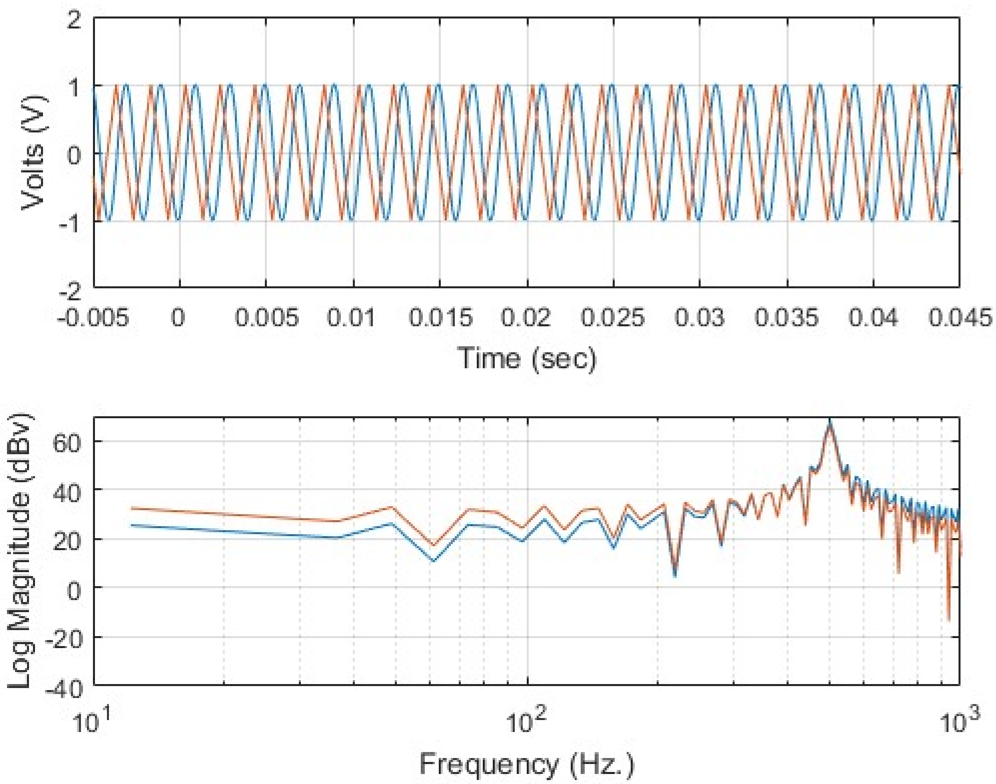


Figure 9 – The MATLAB Fast Fourier Transform of the two waveforms shown on the previous page. The peaks match each other in frequency – both waves' FFT frequencies have peaks at 500 Hz as expected.

5,000 data points were collected for each waveform. The sampling rate was 10 microseconds. The frequency resolution is the inverse of the product of the sampling rate and the number of data points – this value was calculated to be 20 Hz for the experimental setup used. The minimum detectable frequency is equal to the frequency resolution, any lower frequency resolution would lead to aliasing. The maximum determinable frequency is equal to the inverse of twice the sample interval. This value was calculated to be 50 kHz for the data recorded. Frequency resolution could be improved by sampling at a higher rate – that is to say, decreasing the time between each sample. Alternatively, frequency resolution could be increased by decreasing the number of sample points. The most effective method to improve frequency resolution would be to sample at a higher rate, however, oscilloscopes get more expensive with capability.

String Vibration

The tension in the string was determined using the relationship in equation 20.

$$T = \frac{V_{out} F_{FS}}{S_{LC} E_{ex} G} \tag{20}$$

Where T is the tension in the string, V_{out} is the amplitude of the string vibration waveform (.3 V), F_{FS} is the fullscale force (100 lbs), S_{LC} is the sensitivity of the load cell (3.4 mV/V). Converting units to Système international and substituting into equation 20, tension was found to be 78.5 N.

Equation 21 was used to solve for the natural frequency.

$$f_n = \frac{1}{2l}\sqrt{\frac{T}{\rho}}\, n \tag{21}$$

Where ρ is the mass of the string per unit length, T is tension, l is the length of the string, and n is the frequency order (harmonic). Converting units to Système international and substituting into equation 21, with n being an array of the first 4 harmonics. The values used to find the natural frequencies are shown in Table 3 (below).

G = 100	ρ = 6.8 g/m
E_{ex} = 5 V	T = 78.5 N
V_{out} = .3 V	L = .7223 m
F_{FS} = 100 lb	N = [1 2 3 4]
S_{LC} = 3.4 mV/V	

Table 3

Amplitude and Frequency Response

The waveforms for both loadcells were analyzed using MATLAB FFT. Both load cells are plotted with their respective FFTs in Figure 10 (below).

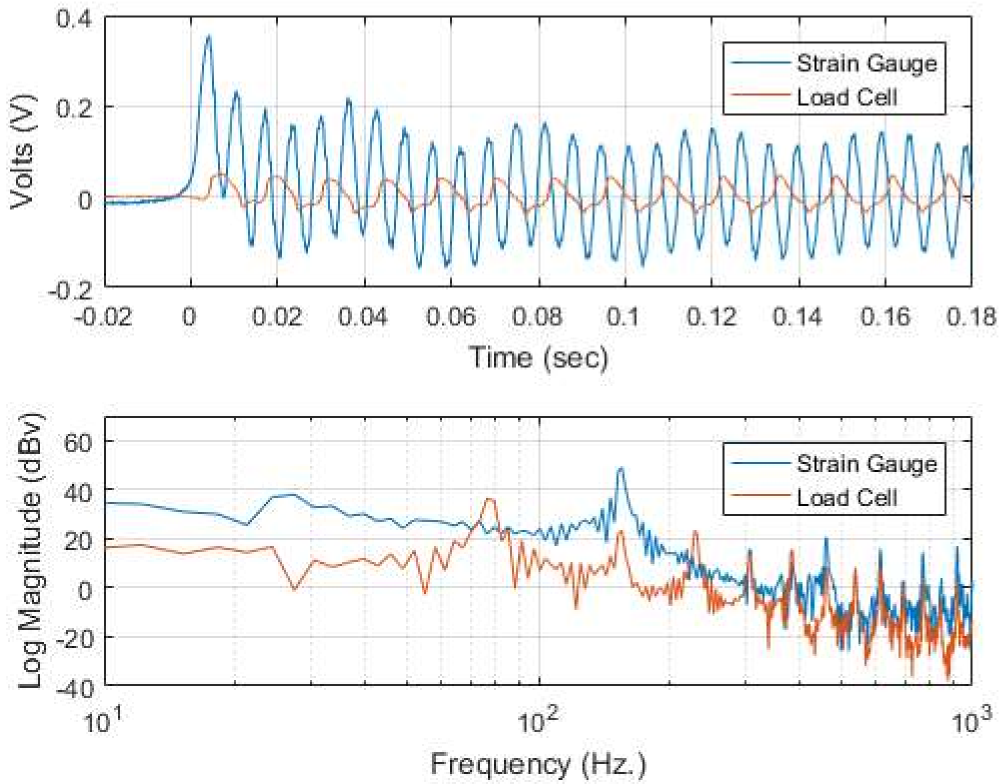


Figure 10 – FFT of both load cell signals. Load cell in the legend refers to the piezoelectric load cell.

Table 4 (below) compares the predicted natural frequency harmonics with the experimental values. Predicted values were found using equations 20 and 21.

Harmonic (n)	Predicted Frequency (Hz)	Experimental Load Cell Frequency (Hz)	Experimental Strain Gauge Frequency (Hz)
1	74.35	76.29	No Spike
2	148.69	152.6	152.6
3	223.04	228.9	No Spike
4	297.39	305.2	308.2

Table 4

The load cells provide the same response with the exception of missing spikes in the response of the strain gauge load cell. This is because one load cell measures a force twice per period – once when the string is at maximum height, and once again when the string is at the minimum height. The other load cell only measures once per period

CODE

```
function [output] = confInt(x,y)

[~,BF] = leastSquares(x,y);
m = BF(1); %slope
b = BF(2); %y intercept

% yfffff = (ThermocoupleVoltage-betaHat(1))/betaHat(2); %° C
% Part3BF = polyfit(ThermistorTemperature,ThermocoupleTemperature,1);
bfy = m*x+b; %Generating y values for the best fit

Yc = bfy; %The value of y predicted by the least squares equation for a given value
of x
tvp = 2.77; %For N = 10, 95% confidence
yiyci = (y-bfy).^2;
sumyiyci = sum(yiyci);
Syx = (sumyiyci/(length(y)-1))^.5; %standard error of the fit
SampleMeanValue = (sum(x))/length(x);

unsummedDen = zeros(1,length(x)); %pre-allocating for faster for loop performance

for i = 1:1:length(x)
   unsummedDen(i) = (x(i)-SampleMeanValue)^2;
end
Den = sum(unsummedDen);

CIofFitPOS = Yc+tvp.*Syx.*(1./length(y)+((x-SampleMeanValue).^2./(Den))).^.5;
CIofFitNEG = Yc-tvp.*Syx.*(1./length(y)+((x-SampleMeanValue).^2./(Den))).^.5;

CIofMeasurementPOS = Yc+tvp.*Syx.*(1+1./length(y)+((x-
SampleMeanValue).^2./(Den))).^.5;
```

```
CIofMeasurementNEG = Yc-tvp.*Syx.*(1+1./length(y)+((x-
SampleMeanValue).^2./(Den))).^.5;

xq = linspace(x(1),x(end));
cifp = interp1(x,CIofFitPOS,xq,'spline');
cifn = interp1(x,CIofFitNEG,xq,'spline');
cimp = interp1(x,CIofMeasurementPOS,xq,'spline');
cimn = interp1(x,CIofMeasurementNEG,xq,'spline');

outputrows = [cifp;cifn;cimp;cimn];
output = outputrows';



end

function [output] = confInt2(x,y)

[~,BF] = leastSquares(x,y);
m = BF(1); %slope
b = BF(2); %y intercept

% yfffff = (ThermocoupleVoltage-betaHat(1))/betaHat(2); %°  C
% Part3BF = polyfit(ThermistorTemperature,ThermocoupleTemperature,1);
bfy = m*x+b; %Generating y values for the best fit

Yc = bfy; %The value of y predicted by the least squares equation for a given value
of x
tvp = 2.080; %For N = 22, 95% confidence
yiyci = (y-bfy).^2;
sumyiyci = sum(yiyci);
Syx = (sumyiyci/(length(y)-1))^.5; %standard error of the fit
SampleMeanValue = (sum(x))/length(x);
```

```
unsummedDen = zeros(1,length(x)); %pre-allocating for faster for loop performance

for i = 1:1:length(x)
    unsummedDen(i) = (x(i)-SampleMeanValue)^2;
end
Den = sum(unsummedDen);

CIofFitPOS = Yc+tvp.*Syx.*(1./length(y)+((x-SampleMeanValue).^2./(Den))).^.5;
CIofFitNEG = Yc-tvp.*Syx.*(1./length(y)+((x-SampleMeanValue).^2./(Den))).^.5;

CIofMeasurementPOS = Yc+tvp.*Syx.*(1+1./length(y)+((x-
SampleMeanValue).^2./(Den))).^.5;
CIofMeasurementNEG = Yc-tvp.*Syx.*(1+1./length(y)+((x-
SampleMeanValue).^2./(Den))).^.5;

xq = linspace(x(1),x(end));
cifp = interp1(x,CIofFitPOS,xq,'spline');
cifn = interp1(x,CIofFitNEG,xq,'spline');
cimp = interp1(x,CIofMeasurementPOS,xq,'spline');
cimn = interp1(x,CIofMeasurementNEG,xq,'spline');

outputrows = [cifp;cifn;cimp;cimn];
output = outputrows';


end

function [output] = dispStrain(deflection)
%this function calculates strain given displacement
deflection = deflection*.0254; %converting inches to meters
thickness = .052*.0254; %meters
```

```
length = (6.291-.382)*.0254; %meters
output = (3.*deflection.*thickness)/(2.*length.^2);

end

function [dampingRatio] = dRatio1(pks)
%this function uses the fist method, with pks as an input

Y = pks; %renaming it for convenience
Y1 = Y(1);

for n = 2:length(Y)
    alpha = (log(Y1/Y(n)))/(n-1);
end

dampingRatio = alpha./((4.*pi.^2+alpha.^2).^.5);

end

function [dampingRatio] = dRatio2(pks)
%where pks is an array of the magnitudes of the peaks i.e. the Ys

Y = pks; %renaming pks to Y, not strictly necessary
Y1 = Y(1);

dampingRatiof = zeros(1,length(Y)); %preallocating the array
for n = 2:1:length(Y)
    dampingRatiof(n) = ((1./(n-1)).*log(Y1./Y(n)))./((4.*pi.^2)+((1./(n-
1)).*log(Y1./Y(n))).^2).^.5;
end

dampingRatio = dampingRatiof';
```

```
end

%% Header
%Simon Popecki
%3. March 2017
%ME 646

%lab3.mat information
%BEAMPRESS: the deflection of the beam from 0 inches to .250 inches
%[deflection (inches) , voltage output (mV)]

%BEAMRELEASE: the return of the beam from .250 inches to 0 inches
%[deflection (inches) , voltage output (mV)]

%DER: the comparison between shunt resistor resistance and the
%corresponding change in voltage [voltage (mV) , resistance (kOhm)]

%ZEROVMASSLOADV: the zero voltage, mass applied to the beam, and the
%voltage outputted by the strain gauage [mV,g,mV]

%BUCKETWEIGHT: the mass of the bucket used to weigh samples in
%ZEROVMASSLOADV, grams

%% Beam/Strain Gauge Response
clear all;
close all;
load lab3.mat
%DER - (:,1) is the output voltage (IN MILLIVOLTS) and (:,2) is
%the resistance for the given shunt resistor(K OHMS)

%The resistance is the independent variable
```

```
RSG = 120; %Ohms
Rshunt = (DER(:,2)*1000); %converting to Ohms
deltaR = RSG-((RSG.*Rshunt)./(RSG+Rshunt));
deltaE = (DER(:,1)/1000)/100; %converting to volts, dividing by gain of 100

[REbestfit,REbestfitmb] = leastSquares(deltaR,deltaE); %the slope and y intercept
of the best fit line
REbfx = REbestfit(:,1);
REbfy = REbestfit(:,2);

%making all the y values for the confidence intervals
conInts = confInt(deltaR,deltaE); %for an N of 5, v of 4
cifp = conInts(:,1);
cifn = conInts(:,2);
cimp = conInts(:,3);
cimn = conInts(:,4);
cix = linspace(deltaR(1),deltaR(end));

m = num2str(REbestfitmb(1),3);
b = num2str(REbestfitmb(2),3);
txt = strcat('V =',m,' (V/¥Omega) R+ ',b,' (V)');
txt2 = 'Sample Variance deg. of Freedom: 4';
txt3 = 'Student''s t Variable at 95% conf.: 2.770';

%Plotting the deltaR vs. deltaE and respective confidence intervals
figure
plot(deltaR,deltaE,'o',REbfx,REbfy,cix,cifp,'b--',cix,cimp,'r--',cix,cifn,'b--',cix,cimn,'r--
')
title('H-Bridge Sensitivity')
ylabel('Change in Output Voltage (V)')
xlabel('Change in Resistance (¥Omega)')
xmin = .08;
```

```
xmax = .27;
ymin = -.0035;
ymax = 0;
axis ([xmin xmax ymin ymax])
text(.35*xmax,.9*ymin,txt)
text(.35*xmax,.85*ymin,txt2)
text(.35*xmax,.8*ymin,txt3)
legend('Data','Least Squares Best Fit','Confidence Interval of Fit (95%)','Confidence
Interval of Meas. (95%)','location','northeast')
grid on

% Part 1.d
Gain = 100;
Marray = ZEROVMASSV(:,2);
Varray = ZEROVMASSV(:,3);

MV = ZEROVMASSV(:,3)-ZEROVMASSV(:,1); %taking voltages from the
displacement calculation for comparison
VstrainM = abs(strain(MV,Gain)); %strain as calculated from the strain gauge
(taking absolute value)


DP =  BEAMPRESS(:,1); %inches
DVP = BEAMPRESS(:,2); %voltages for pressing on the beam (mV)
VstrainDP = abs(strain(DVP,Gain)); %strain as calculated from the strain gauge
(taking absolute value)
DPstrain = dispStrain(DP);

DR = BEAMRELEASE(:,1); %inches
DVR = BEAMRELEASE(:,2); %inches

VstrainDR = abs(strain(DVR,Gain));
```

```
DRstrain = dispStrain(DR);

Mstrain = massStrain(Marray);
MVstrain = strain(Varray,Gain);

%Best fit line for part 1.d.iii
BEAM = (BEAMPRESS+flipud(BEAMRELEASE))/2;
BEAMx = BEAM(:,1);
BEAMy = BEAM(:,2);
BEAMystrain = strain(BEAMy,Gain);
BEAMfitV = leastSquares(BEAMx,BEAMy);
BEAMfitVolts = BEAMfitV(:,2);
BEAMfit = strain(BEAMfitVolts,Gain);
% beamm = BEAMfitmb(1);
% beamb = BEAMfitmb(2);
conInts2 = confInt2(BEAMx,BEAMystrain);
cifp2 = conInts2(:,1);
cifn2 = conInts2(:,2);
cimp2 = conInts2(:,3);
cimn2 = conInts2(:,4);
cix2 = linspace(BEAMx(1),BEAMx(end));

%Hysteresis
BPstrain = strain(BEAMPRESS(:,2),Gain);
BRstrain = strain(BEAMRELEASE(:,2),Gain);
BeamDiff = BPstrain-BRstrain;

figure
plot(DP,DPstrain,'r:',DP,VstrainDP,'o',DR,VstrainDR,'*')
title('Strain Measured from Displacement vs. Prediction')
xlabel('Displacement (in)')
ylabel('Strain')
```

```
legend('Prediction','Data - Press','Data - Release','locaton','northwest')
grid on

figure
plot(Marray,Mstrain,'r:',Marray,MVstrain,'o')
title('Strain Measured from Mass vs. Prediction')
xlabel('Added Mass (g)')
ylabel('Strain')
legend('Prediction','Data')
grid on

txt4 = 'Sample Variance deg. of Freedom: 21';
txt5 = 'Student''s t Variable at 95% conf.: 2.080';
figure
plot(DP,VstrainDP,'o',DR,VstrainDR,'*',BEAMx,BEAMfit,':',cix2,cifp2,'b--
',cix2,cimp2,'r--',cix2,cifn2,'b--',cix2,cimn2,'r--')
title('Best Fit and 95% Conf. Interval for Displacement')
xlabel('Displacement (inches)')
ylabel('Strain')
legend('Data (Press)','Data (Release)','Linear Least Squares Fit','95% Conf. Interval
of Fit','95% Conf. Interval of Meas.','location','northwest')
grid on
xmin = 0;
xmax = .25;
ymin = .0001;
ymax = .0007;
axis ([xmin xmax ymin ymax])
text(.35*xmax,.25*ymax,txt4)
text(.35*xmax,.2*ymax,txt5)

%% Cantilever Beam Vibration - Part 2
```

```
%clearing old variables and loading in data from lab book
load lab3.mat

%loading data and processing it into usefull arrays (only working with
%test 3)
nheaderlines=29; %data starts on line 30
% test1struct = importdata('Test1.lvm','¥t',nheaderlines); %importing three
waveforms, at least one should be good
% test2struct = importdata('Test2.lvm','¥t',nheaderlines);
test3struct = importdata('Test3.lvm','¥t',nheaderlines);
% test1 = test1struct.data; %pulling useful data out of the structured array
% test2 = test2struct.data;
test3 = test3struct.data;
% test1t = test1(:,1); %the time array for test 1 - units are presumed to be seconds
% test2t = test2(:,1); %the time array for test 2
test3traw = test3(:,1); %the time array for test 3
% test1V = test1(:,2); %the voltage array for test 1
% test2V = test2(:,2); %the voltage array for test 2
test3Vraw = test3(:,2); %the voltage array for test 3

%plotting the data together to see what looks best
% figure
% plot(test1t,test1V,test2t,test2V,test3traw,test3Vraw)
% title('Test Plot - Which Data Set is Best?')
% xlabel('Time (s)')
% ylabel('Voltage (V)')
% legend('Test 1','Test 2','Test 3')
%test 2 is the biggest, test 1 is the smallest, and test 3 in inbetween
%going with test 3 data for no particular reason....

%removing the part of the data before 0 seconds
test3t = test3traw(test3traw >=0);
```

```
test3V = test3Vraw(end-(length(test3t))+1:end);
%plotting to see how it looks (not normalized about zero yet)
% figure
% plot(test3t,test3V)
%looks good!

%normalizing the data about 0 V
meantest3V = mean(test3V);
test3Vn = test3V-meantest3V; %creating the normalized waveform which gives us
deltaE
%plotting to check
% figure
% plot(test3t,test3Vn)
%looks good!

%converting the normalized waveform to strain
Gain = 100; %the gain is probably 100, larger gains make strain smaller
test3s = strain(test3Vn,Gain);

%plotting to check
% figure
% plot(test3t,test3s)
% grid on
%all values between .00000005 and .00000022
th = .00000005;
[pks,dep,pidx,didx] = peakdet(test3s,th);
peakLocations = test3t(pidx);
depLocations = test3t(didx);

%plotting the strain vs. time waveform with peaks and depressions
figure
plot(test3t,test3s,peakLocations,pks,'r.',depLocations,dep,'r.')
```

```
title('Strain vs. Time for Vibrating Beam')
xlabel('Time (s)')
ylabel('Strain')
legend('Data','Peaks')

%finding the damping ratio using method 1
DR1 = dRatio1(pks);

%Finding the damping ratio using method 2
DR2 = dRatio2(pks); %creates an array of damping ratios for the second method
(put this in a table)
avgDR2 = mean(DR2); %finding the arithmetic mean of the damping ratios from
method 2 (results in single DR number)
stdDR2 = std(DR2); %standard deviation of the DR values from method 2


%finding the damped natural frequency
Td = peakLocations(2)-peakLocations(1); %distance between the first and second
peak
omegaD = (2*pi)./Td; %this is the damped natural frequency

%finding the undamped natural frequency
omegaN = omegaD./((1-(avgDR2.^2)).^.5); %using the damping ratio found by the
second method (they are basically the same)


%calculating beam stiffness
E = 200000000000; %Pa
width = .503*.0254; %m
thickness = .052*.0254; %m
I = ((width*(thickness^3))/12);
stiffCalc = (3*E*I)/((((6.291-.382)*.0254)^3); %from the cantilever beam equation
```

```
%finding beam stiffness from data
Farray = (ZEROVMASSV(:,2)./1000*9.81); %getting force in newtons from the force test data
VFarray = (ZEROVMASSV(:,3)-ZEROVMASSV(:,1)); %getting deltaE from the force data
intarray = linspace(Farray(1),Farray(end),2000); %newtons
interpVF = interp1(Farray,VFarray,intarray);

Darray = (BEAMPRESS(:,1).*.0254); %getting displacement in meters from displacement test data
VDarray = (BEAMPRESS(:,2)-BEAMPRESS(1)); %getting the deltaE for displacement

POIarray = find(interpVF >= VDarray(6));
POIindex = POIarray(1); %position in the interpVF array where displacement is a known value
DPOI = Darray(6); %the displacement in meters at the point of interest
FPOI = intarray(POIindex); %the force in Netwons at the point of interest

stiffExp = FPOI/DPOI;

%finding beam volume
Vbeam = ((6.291-.382)*.0254)*width*thickness;
rhoSteel = 8050; %kg/m^3
MassofBeam = Vbeam*rhoSteel;

%% String Vibration - Part 4
clear all;
close all;

%Loading and plotting the data, finding the FFT
```

```
nheaderlines=32;
wave = importdata('p3.lvm','¥t',nheaderlines); %You may have to adjust this
%the lvm file has four columns
%wave2 = importdata('Part2_1_triangle.lvm','¥t',nheaderlines);

t = wave.data(:,1);   %time vector
v1 = wave.data(:,2);  %waveform 1 - strain gauge
v2 = wave.data(:,4);  %waveform 2 - load cell
subplot(2,1,1)
plot (t,v1,t,v2); grid
xlabel ('Time (sec)')
ylabel ('Volts (V)')
legend('Strain Gauge','Load Cell')
T = t(2)-t(1);         % Time per sample
Fs = 1/T;              % Sampling frequency
L = size(v1);          % Length of signal - # of points
NFFT = 2^nextpow2(L(1)); % Next power of 2 from length of y - need for FFT
Y1 = fft(v1,NFFT)./L(2); % this is a vector with complex number elements
Y2 = fft(v2,NFFT)./L(2); % this is a vector with complex number elements
% % Y(w)=A(w)-iB(w)=Integral of [y(t)*exp(-iwt)dt]
%Discrete Fourier Transform output
%                      N
%       Y(k) =       sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
%                     n=1
% The spacing is determined from 1/Ndt where dt is the time spacing and N
% is the number of points.
% Need to divide by L because the function leaves that out of the output

f = Fs/2*linspace(0,1,NFFT/2+1); % linspace generates linearly spaced points
% % to the maximum frequency determined by sampling rate
%
% % f is the frequency vector
```

```
% % Factor of 2 is because it is a two sided FT
%
% % Plot single-sided amplitude spectrum.
subplot(2,1,2)
AY1=20*log10(abs(Y1(1:NFFT/2+1)));
AY2=20*log10(abs(Y2(1:NFFT/2+1)));
semilogx(f,AY1,f,AY2);grid  % abs(Y) = (Re(Y)^2 + Im(Y)^2)^1/2
axis([10 1000 -40 70])
xlabel ('Frequency (Hz.)')
ylabel ('Log Magnitude (dBv)')
legend('Strain Gauge','Load Cell')

%Finding Tension
Vout = .3; %Volts
G = 100; %gain
Eex = 5; %Volts
Ffs = 100*4.44822; %force, converted from pounds to N
Slc = 3.4/1000; %Slc - V/V (converted from mV/v)

T = (Vout*Ffs)/(Slc*Eex*G); %tension in Newtons

%Finding the natural frequency
rho = .006805; %kg/m - the mass per unit length of the string
stringLengthInches = 28+7/16; %string length in inches
l = stringLengthInches*.0254; %converting the string length to meters
n = [1 2 3 4]; %The harmonics we are interested in

fn = (1./(2.*l)).*((T/rho).^.5).*n; %finding natural frequency
```

```
function [output1, output2] = leastSquares(x,y)
%this function produces a least squares best fit line from the given x and
%y arrays, there are 100 points in the best fit array



input = [x,y]; %input line
pts = length(input);            % number of points
X = [ones(pts,1), input(:,1)];   % forming X of X beta = y
y = input(:,2);               % forming y of X beta = y
betaHat = (X' * X) ¥ X' * y;   % computing projection of matrix X on y, giving beta
%disp(betaHat);
% plot the best fit line
%xx = linspace(0,100);
%yy = betaHat(1) + betaHat(2)*xx; %betaHat(1) is the Y-intercept, and betaHat(2)
is the slope
% plot the points (data) for which we found the best fit
m = betaHat(2);
b = betaHat(1);

xx = x;
yy = betaHat(1) + betaHat(2)*xx;

output1 = [xx,yy]; %the actual line
%output1 = output1';
output2 = [m,b]; %slope of the line and y intercept
```

```
end

function [output] = massStrain(M)
%this function determines strain in a steel bar given mass added to a
%bucket
M = M/1000; %converting from grams to kilograms
thickness = .052*.0254; %meters
width = .503*.0254; %meters
length = (6.291-.382)*.0254; %meters
E = 200000000000; %young's modulas for steel, Pa
g = 9.81; %m/s^2
BUCKETMASS =.0119; %kg
HOOKMASS = .0074; %kg
M = M+BUCKETMASS+HOOKMASS; %accounting for the additional mass from the
hook and the bucket

I = (width*(thickness^3))/12;
output = (M.*g.*length).*(thickness./2)/(E.*I);

end
```