ME Labs
Jack Button: Technical Writer
Jack Dibachi: Team Leader
Scott Smith: Technical Analyst

# Experiment 4: Vibrations

## Objectives

This experiment examines the vibration characteristics of a cantilever beam, and uses these characteristics to calibrate an accelerometer under dynamic conditions. Once calibrated, the accelerometer can be used to obtain the vibration characteristics of a spring and a foam block.

## Introduction

In order to fully understand the vibration profile of an object, a strain gauge and accelerometer can be used to collect data on the position and acceleration profile of a point along the object. In this lab, these sensors are used to characterize the vibrational profile of a cantilever beam.

Strain gauges are used to measure the percent deformation or “strain” at that given point, which can further be related to the change in position at that point or the change in position of the end tip of the cantilever. Because the strain gauge changes in resistance with changes in strain, the LabVIEW VI set up in this lab can read this change in property through a voltage output delivered by a wheatstone bridge. This relationship will effectively calibrate the strain gauge such that the voltage output of the strain gauge can be used to gather the profile of the beam over the course of a vibration.

The accelerometer is used to measure the acceleration of the tip of the cantilever beam. However, the accelerometer is a second order with a frequency dependent response, so the phase shift between the actual acceleration of the system and the acceleration that is read by the accelerometer must be gathered. This is why it is not only important to calibrate the accelerometer by relating the voltage output of the accelerometer to acceleration statically, but it is also important to characterize the accelerometer dynamically to understand the response time of the accelerometer.

The cantilever tip displacement behaves as a decaying harmonic oscillator. Since the displacement is a maximum at the point of release, the phase shift of the cosine wave is zero, so the tip has the following behavior over time:

(Equation 1) $$\delta(t) = \delta_{max} e^{-\frac{t}{\tau}} cos(\omega_d t)$$

The acceleration of the tip can be obtained by taking the second derivative of the displacement with respect to time, which gives the following relationship:

(Equation 2) $$a(t) = \delta_{max}(\tau^{-2} - \omega_d{}^2) e^{-\frac{t}{\tau}} cos(\omega_d t) + \frac{2\delta_{max}\omega_d}{\tau} e^{-\frac{t}{\tau}} sin(\omega_d t) = A e^{-\frac{t}{\tau}} cos(\omega_d t + \phi - \frac{\pi}{2})$$

(Equation 3) $$A = \sqrt{(\delta_{max}(\tau^{-2} - \omega_d{}^2))^2 + (\frac{2\delta_{max}\omega_d}{\tau})^2},\ \phi = arctan(\frac{1}{2\omega_d\tau} - \frac{\omega_d}{2\tau})$$

These amplitude and phase shift values will be compared with the amplitude and phase shift resulting from the curve fitted accelerometer data to obtain the dynamic accelerometer calibration parameters.

The effective spring constant can be calculated through the decay rate of the oscillations observed in this lab. This is calculated by finding the undamped natural frequency of the system by the following derivation:

(Equation 4) $$\tau = \frac{1}{\xi\omega_n}$$

(Equation 5) $$\omega_d = \omega_n\sqrt{1 - \xi^2}$$

Combining equation 2 and equation 3:

(Equation 6) $$\tau = \frac{\sqrt{1-\xi^2}}{\omega_d\xi}$$

Isolating the damping ratio produces:

(Equation 7) $$\xi = \sqrt{\frac{1}{\tau^2\omega_d{}^2+1}}$$

Lastly, combining equation 5 with equation 3 and isolating $\omega_n$:

(Equation 8) $$\omega_n = \frac{\omega_d}{\sqrt{1-\frac{1}{\tau^2\omega_d{}^2+1}}}$$

The relationship between the natural frequency of the system and the spring constant of the cantilever beam is related through the equation for calculating natural frequency:

(Equation 9) $$k = \omega_n{}^2 m$$

And further combining equation 6:

$$k = \frac{\omega_d{}^2}{1-\frac{1}{\tau^2\omega_d{}^2+1}} m$$

Simplified

(Equation 10) $$k = \omega_d{}^2 m + \frac{m}{\tau^2}$$

This can also be used to further calculate the Young's modulus of the beam by the following equation.

(Equation 11) $$E = \frac{kL^3}{3I}$$

The effective damping coefficient is found two ways. The first way starts back at Equations 1 with the identity:

(Equation 12) $$\tau = \frac{1}{\xi\omega_n} = \frac{2m}{c}$$

Rearranging this finally gets is the damping coefficient:

(Equation 13) $$c = \frac{2m}{\tau}$$

The first way is to be used when it is easier to find the time constant from curve fitting. If that, however, doesn't work the logarithmic decrement should be used as shown below. Where xo and x1 represent subsequent peaks of the sinusoid.

(Equation 14) $$\zeta = \frac{1}{\sqrt{1+(2\pi\ ln(xo/x1))^2}} \approx \frac{ln(xo/x1)}{2\pi} \text{ For } \zeta \ll 1$$

For the mass drop experiment, the position of the mass over time can be gathered from by numerically integrating the position data. However, there is likely to be inherent offset in the acceleration data which will result in drift over the integration process. On top of this, there is a significant amount of noise in the data which needs to be filtered out.

MatLab built-in filtering function can be set up by gathering the parameters required to build a butterworth filter. The normalized cutoff frequency of the filter is equal to the frequency of the noise divided by the nyquist frequency (½ the sampling rate = 5000Hz). The noise frequency is found by building a single-sided power spectrum.

(Equation 15) $$\frac{f_{noise}}{f_{Nyquist}}$$

A cumulative trapezoidal numerical integration scheme can be set up to integrate the filtered acceleration data. However, as mentioned, this can result in large amounts of drift. If assumed to be linear, this drift can be removed by the use of boundary conditions. The acceleration (minus gravitational acceleration) of the block is known to be 0 m/s2 at the beginning of the experiment and 0 m/s2 at the end of the experiment. This is known to be the same for velocity (0 m/s respectively). For position, the boundary conditions are known to be 0m initially and -H (recorded change in height) m at the end. A correction factor can be built for this.

(Equation 16) $$a_{cf} = \frac{a_f - a_0}{t_f - t_0} * t$$

(Equation 17) $$a_{correct} = a_{recorded} - a_{cf}$$

Integrate numerically to get calculated velocity profile $v_{cf}$.

(Equation 18) $$v_{cf} = \frac{v_f - v_0}{t_f - t_0} * t$$

(Equation 19) $$v_{correct} = v_{recorded} - v_{cf}$$

Integrate numerically to get calculated velocity profile $p_{cf}$.

(Equation 20) $$p_{cf} = \frac{p_f - p_0 + H}{t_f - t_0} * t$$

(Equation 21) $$p_{correct} = p_{recorded} - p_{cf}$$

The limitations of this numerical integration and correction scheme are that it assumes error in data is corrected linearly. This is not always the case as other non-linear errors in acceleration data can accumulate throughout the integration process. This issue will be exemplified in the discussion section when talking about initial erroneous spring drop data.

**Methods**

The first step of this lab is calibrating the strain gauge. In order to do this, the strain gauge is placed at a known distance from the cantilever base lengthwise. Vertical calipers are placed at the end of the beam to measure the relative tip displacement. Measurements are taken at 10 different relative displacements in order to produce a curve which can describe the relationship between strain gauge readout and tip displacement.

After this, the strain gauge calibration data collection is finished and the accelerometer can now be statically calibrated. This is done by attaching the accelerometer to a pivot and collecting data on the change in accelerometer readout with the change in angle. Although the static calibration data is not directly used to analyze the other data, it provides a context for the voltage readout of the accelerometer relative to gravitational acceleration. The accelerometer data is collected over the range of 90 degrees to -90 degrees in intervals of 20 degrees, corresponding to a static gravitational acceleration ranging from ±g.

The next step in the lab is to calibrate the accelerometer dynamically by attaching the accelerometer to the end of the cantilever beam and displacing the tip of the cantilever beam a known distance. After running the LabVIEW VI for recording accelerometer and strain gauge data, the cantilever tip is released from the known distance and the vibrations are recorded. This is repeated with 3 different added mass values (105.9g, 204.1g, 259.4g).

After the accelerometer is fully calibrated, the acceleration of a falling mass impact with a spring can then be recorded. First the accelerometer is attached to a 190 g block, which is free

to slide along a slide. The mass is released from varying heights onto the spring, and trials are repeated with a foam block in place of the spring, and with two masses (383 g) instead of one mass.

With all the accelerometer voltage data collected in LabVIEW for varying heights and mass, the Fast Fourier Transform finds the dominant frequency and the accelerometer calibration parameters corresponding to the oscillation frequency convert to acceleration. The newly converted acceleration data is then integrated numerically with the proper boundary conditions to obtain the object position over time. The position data is used to determine the system damping coefficient, and the spring constant is determined by Hooke's Law.

**Results**

| **Table 1: Linear sensitivity of a cantilever-mounted strain gauge and an accelerometer** | | |
|---|---|---|
| **Device** | **Linear Sensitivity** | **Percent Sensitivity** |
| Strain gauge | 0.0149 ± 0.0001 V/mm | 1.223 ± 0.005% |
| Accelerometer (Y) | -0.192 ± 0.002 V/(m/s$^2$) | 0.55 ± 0.01% |
| Accelerometer (Z) | 0.186 ± 0.004 V/(m/s$^2$) | 0.55 ± 0.01% |

| **Table 2: Beam Dimensions** | | | |
|---|---|---|---|
| **Length (mm)** | **Width (mm)** | **Thickness (mm)** | **Strain Gauge Location (mm)** |
| 254.02 ± 0.01 | 30.02 ± 0.01 | 1.25 ± 0.01 | 25.35 ± 0.01 |

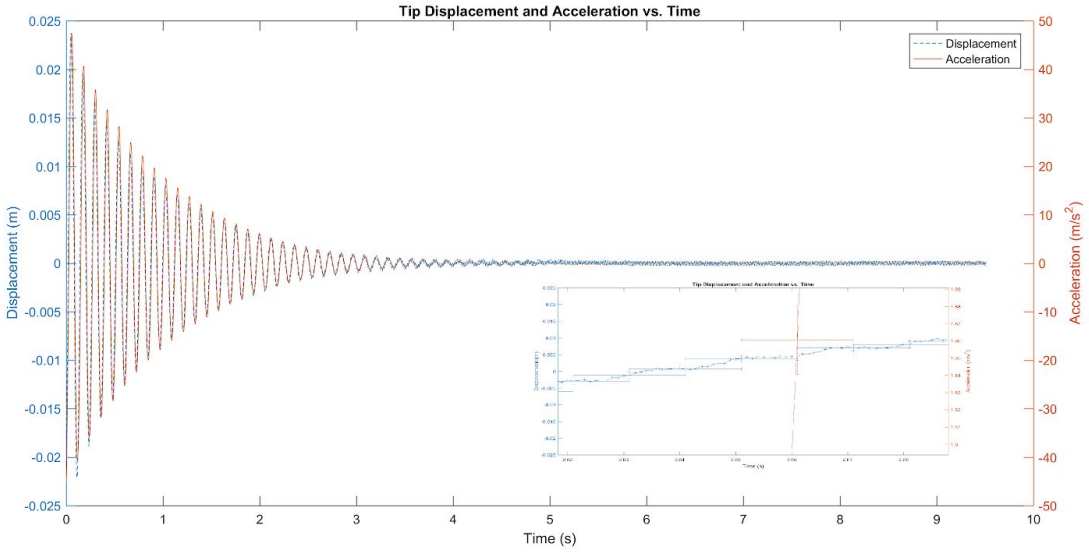



**Figure 1:** Cantilever tip displacement and acceleration over time, in response to an initial deflection of 25.62 mm and a mass of 30.0 g. Error bars are enlarged to show detail.

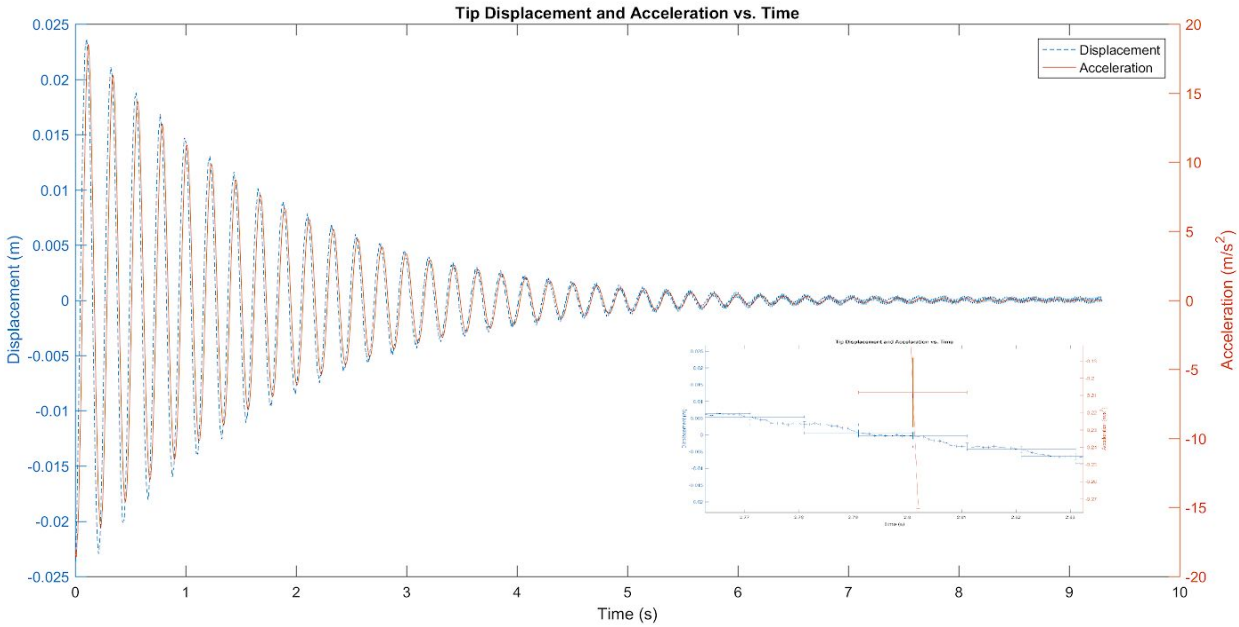



**Figure 2:** Cantilever tip displacement and acceleration over time, in response to an initial deflection of 26.84 mm and a mass of 128.8 g. Error bars are enlarged to show detail.

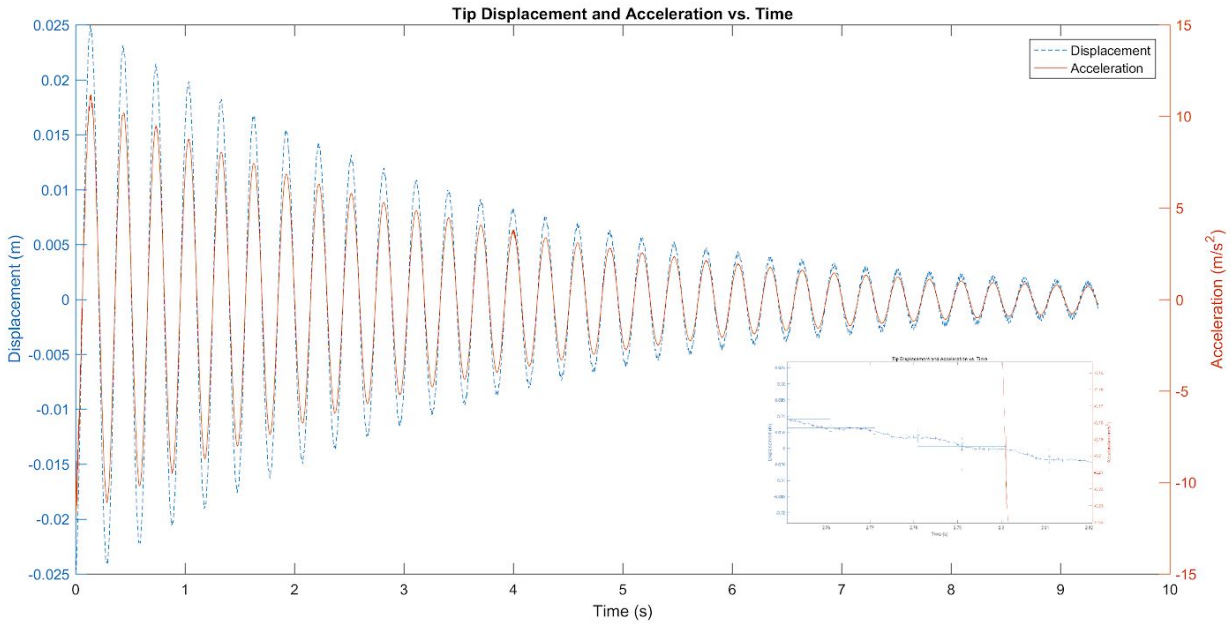



**Figure 3:** Cantilever tip displacement and acceleration over time, in response to an initial deflection of 27.08 mm and a mass of 227.0 g. Error bars are enlarged to show detail.

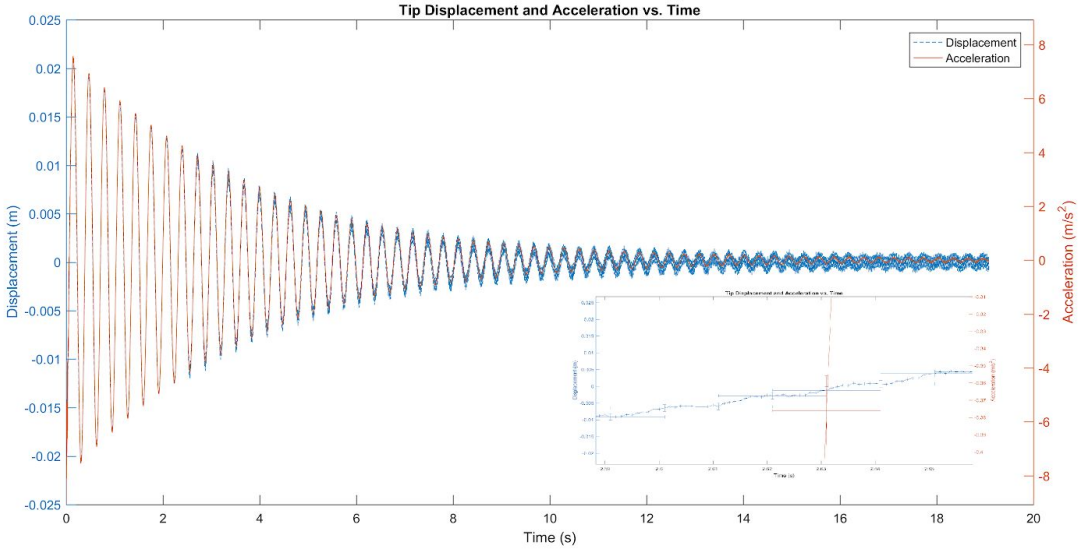



**Figure 4:** Cantilever tip displacement and acceleration over time, in response to an initial deflection of 22.69 mm and a mass of 282.3 g. Error bars are enlarged to show detail.

| Table 3: Oscillation parameters and derived beam characteristics | | | | |
|---|---|---|---|---|
| **Mass (g)** | **Damped Natural Frequency (rad/s)** | **Decay Constant (s)** | **Spring Constant (N/m)** | **Young's Modulus (GPa)** |

| | | | | |
|---|---|---|---|---|
| 30.0 ± 0.1 | 53.204 ± 0.001 | 0.481 ± 0.001 | 85.05 ± 0.28 | 107.54 ± 5.48 |
| 30.0 ± 0.1 | 51.821 ± 0.002 | 0.945 ± 0.005 | 80.60 ± 0.27 | 101.91 ± 5.20 |
| 128.8 ± 0.1 | 28.424 ± 0.002 | 1.623 ± 0.004 | 104.11 ± 0.08 | 131.64 ± 6.70 |
| 128.8 ± 0.1 | 28.452 ± 0.001 | 1.540 ± 0.010 | 104.32 ± 0.08 | 131.91 ± 6.71 |
| 227.0 ± 0.1 | 21.212 ± 0.001 | 3.350 ± 0.010 | 102.16 ± 0.05 | 129.17 ± 6.57 |
| 227.0 ± 0.1 | 21.213 ± 0.001 | 3.220 ± 0.010 | 102.17 ± 0.05 | 129.19 ± 6.57 |
| 282.3 ± 0.1 | 19.667 ± 0.001 | 3.211 ± 0.006 | 109.22 ± 0.04 | 138.10 ± 7.03 |
| 282.3 ± 0.1 | 19.677 ± 0.001 | 3.026 ± 0.005 | 109.33 ± 0.04 | 138.24 ± 7.03 |
| **Average (30g Excluded Outlier):** | | | 105.22 ± 0.06 | 133.04 ± 6.77 |

**Table 3 Note:** Tabulated spring constant and Young's modulus calculated from amplitude and phase shift values found in Appendix I. The 30g measurements are treated as an outlier within the total average, as explained in the discussion section.

| **Table 4: Averaged Vibration Profile Values** | | | | |
|---|---|---|---|---|
| **Test** | **30g Mass** | **128.8g Mass** | **227.9g Mass** | **282.3g Mass** |
| Decay Constant Average (s) | 0.713 ± 0.003 | 1.58 ± 0.01 | 3.29 ± 0.01 | 3.118 ± 0.006 |
| Spring Constant (N/m) | 82.82 ± 0.04 | 104.21 ± 0.08 | 102.16 ± 0.05 | 109.28 ± 0.04 |
| Young's Modulus (GPa) | 104 ± 5 | 131 ± 6 | 129 ± 6 | 138 ± 7 |

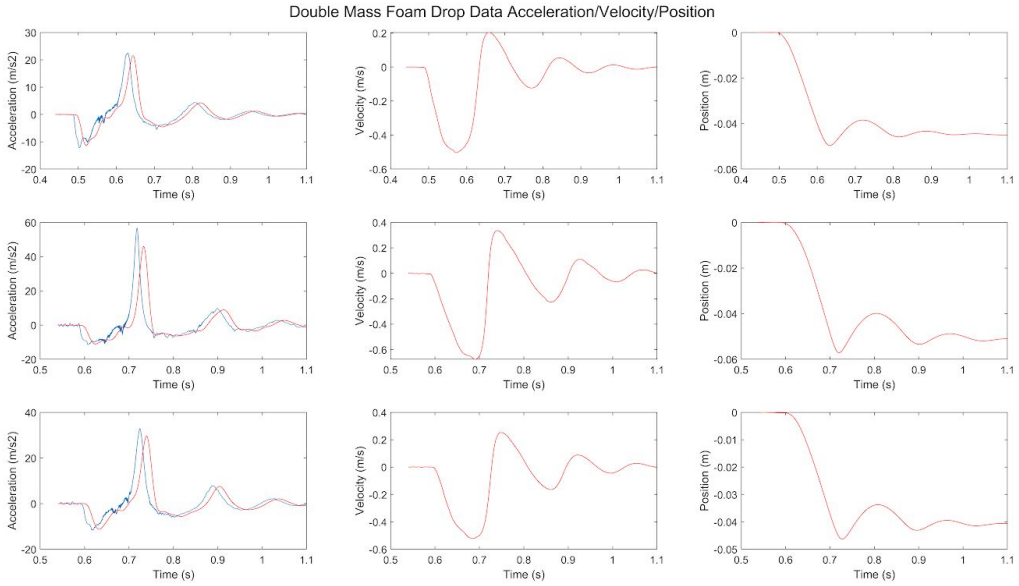



**Figure 5:** Plots of Acceleration, Velocity, and Position for Drops 14-16. These drops are of 383g mass and onto a foam block.

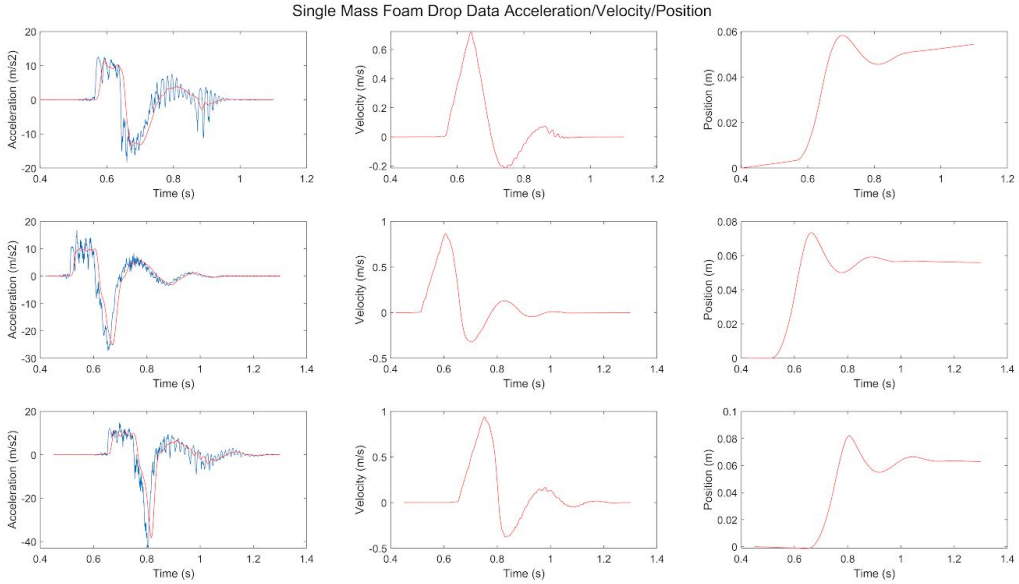



**Figure 6:** Plots of Acceleration, Velocity, and Position for Drops 17-19. These drops are of 190g mass and onto a foam block.

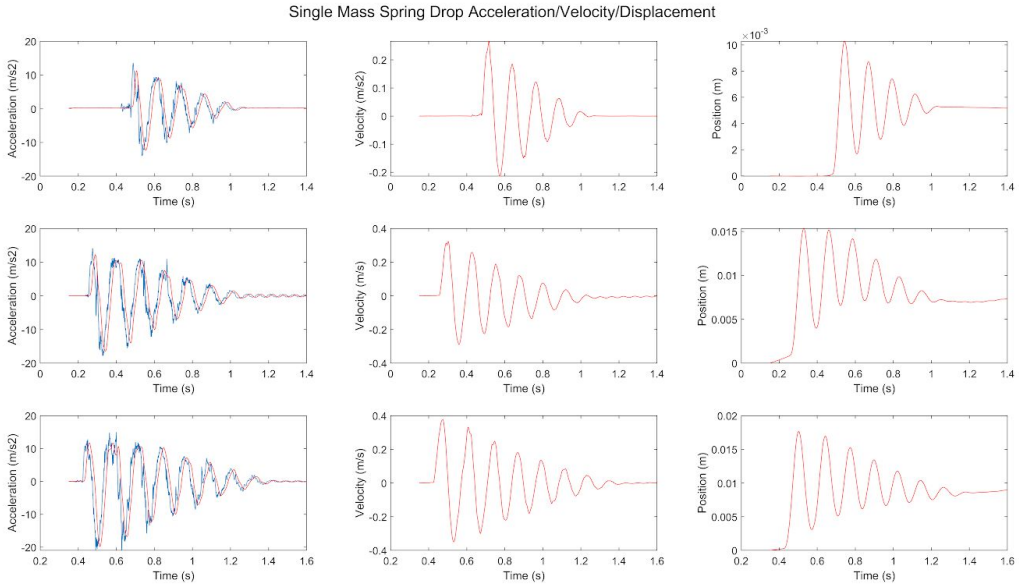



**Figure 7:** Plots of Acceleration, Velocity, and Position for Drops 21-23. These drops are of 190g gram mass and onto a spring.

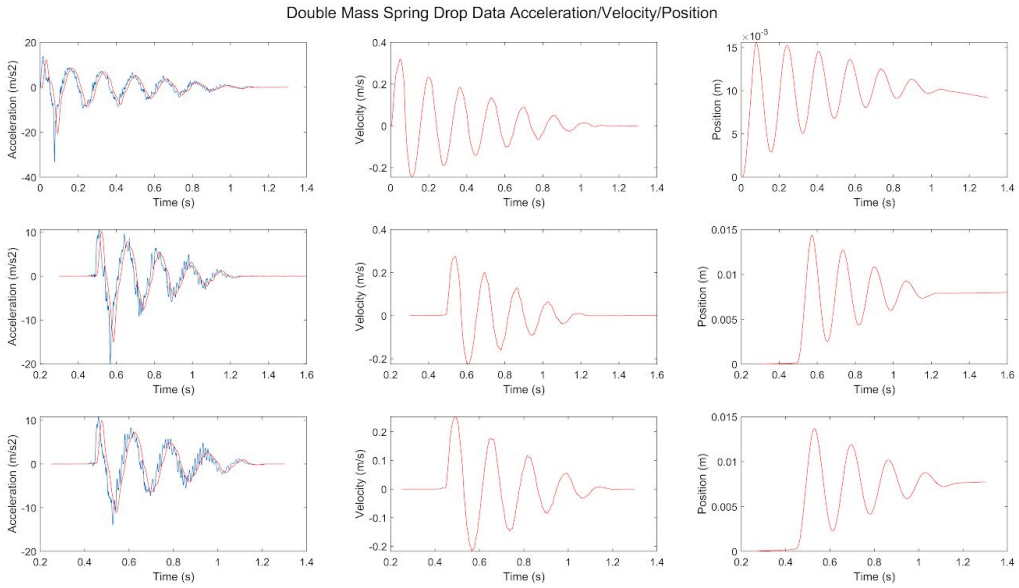



**Figure 8:** Plots of Acceleration, Velocity, and Position for Drops 24-26. These drops are of 383g mass and onto a spring.

| Table 5: Maximum penetration depth from equilibrium (mm) | | | |
|---|---|---|---|
| | **Height 1** | **Height 2** | **Height 3** |
| **190g Mass - Foam** | 4.6 | 6.2 | 5.7 |
| **383g Mass - Foam** | 54.4 | 56.0 | 63.9 |
| **190g Mass - Spring** | 5.2 | 7.4 | 9.0 |
| **383g Mass - Spring** | 5.2 | 7.4 | 9.0 |

| Table 6: Foam Block and Spring Calculated Spring Constant (N/m) | | | |
|---|---|---|---|
| | **Height 1** | **Height 2** | **Height 3** |
| **190g Mass - Foam** | 63.28 ± 0.04 | 66.67 ± 0.04 | 61.51 ± 0.04 |
| **383g Mass - Foam** | 88.03 ± 0.03 | 88.23± 0.03 | 88.89 ± 0.02 |
| **190g Mass - Spring** | 507.4 ± 0.2 | 524.0 ± 0.2 | 509.1 ± 0.2 |
| **383g Mass - Spring** | 586.1 ± 0.2 | 535.6 ± 0.2 | 558.3 ± 0.2 |

| Table 7: Foam Block and Spring Calculated Damping Coefficient (Ns/m) | | | |
|---|---|---|---|
| | **Height 1** | **Height 2** | **Height 3** |
| **190g Mass - Foam** | 7.21 ± 0.03 | 5.92 ± 0.03 | 3.55 ± 0.02 |
| **383g Mass - Foam** | 6.4 ± 0.04 | 5.32 ± 0.04 | 9.85 ± 0.04 |
| **190g Mass - Spring** | 4.5 ± 0.2 | 1.0± 0.2 | 0.9± 0.2 |
| **383g Mass - Spring** | 1.9± 0.2 | 13.2± 0.2 | ± 0.2 |

## Discussion

### Static Sensors

As demonstrated in Appendix I Figure I, the strain gauge behaves linearly with incremented cantilever tip deflections. Although the output is subject to noise of ± 0.01 V, which is on the order of the strain gauge absolute sensitivity in Table 1, the gauge has a wide operating range of voltages across the domain of tip deflections. As a result, the

strain gauge may not be a good linear transducer for deflections close to a millimeter, but is a good linear transducer for deflections on the order of centimeters.

When compared to the orientation dependent gravitational acceleration , the accelerometer behavior is linear from -90 degrees to 90 degrees. Although the accelerometer sensitivity is higher than that of the strain gauge, its suitability as a linear transduced is dependent of the device it is affixed to. The accelerometer can only be used as a linear transducer if the direction of actuation is parallel with the gravitational acceleration, and if the angle the accelerometer experiences can be effectively related to its location in space.

**Dynamic Accelerometer Calibration**

The amplitude adjustment and phase shift required to match the accelerometer voltage output to the actual acceleration of the cantilever tip does not vary much across the range of frequencies tested. However, the range of frequencies feasible with the lab equipment and with a cantilever beam is somewhat narrow, varying by only about 33 rad/s. The accelerometer demonstrates an increase in the phase adjustment and amplitude adjustment factor required as frequency increases, which is a result of the second order response of the accelerometer to a forced oscillation. As the frequency increases, the amplitude of oscillations inside the accelerometer decreases, thus the adjustment factor required to bring the accelerometer up to the true acceleration amplitude increases. This also forces a delay in the accelerometer response, which manifests as a phase shift. As the accelerometer changes direction faster, the internal components take longer to track the accelerometer body motion, creating a shift in the waveform phase. If a larger range of masses are available, or if a stiffer cantilever beam is available, a wider range of oscillation frequencies can be tested to better determine the dependency of the calibration factors on frequency.

**Vibration Profile and Beam Characteristics**

The data from Table 3 shows the calculated Young's modulus and spring constant values found through adjusted accelerometer data. Because the spring constant is a value intrinsic to the oscillating beam, these values should remain constant throughout each test. This was very clearly observed to be the case for all tests containing a mass, even though their masses, frequencies of oscillation, and decay constants were far different from each other.

This effect was, however, not observed in the experiments which only involve the mass of the accelerometer (30g). This is likely because of the fact that the mass of the beam is not negligible in the test. Throughout the experiment, the mass of the beam is assumed to be negligible in each test, due to the fact that it is completely overwhelmed by the effect of the mass at the tip of the cantilever. The effective mass of the cantilever (within the mass spring system in this lab), would be 1/3rd of its total mass, by Rayleigh's method. Adding mass to the beam is calculated (and observed) to significantly lower the damped oscillating frequency. This would mean that the unaccounted for beam mass would increase the mass of the system from 30g to 30g+

$\frac{1}{3}m_{beam}$ and would further decrease the frequency of oscillation. The frequency of oscillation is quadratically related to the calculated spring constant and thus, the spring constant decreases quadratically with a decrease in frequency. This suggests that, even though the mass of the beam is assumed to be very small, with very little added mass, it can have a significant impact on the calculated spring constant and Young's modulus of the beam.

It is unlikely that any discrepancy this large is due to fitting parameters or test setup. The test setups remained consistent throughout each iteration and the fits/adjustments made to acceleration data produced high R-squared values and were visibly consistent.

**Position Derivation from Accelerometer Data**

In this experiment, the amplitude and phase adjustments (dynamic calibration) for the accelerometer become extremely important. The mass drop produces an oscillating signal from the accelerometer that is similar in frequency to the oscillations produced by the cantilever beam system in the dynamic calibration, so the amplitude and phase adjustments can be similarly applied in this scenario in order to find the true acceleration of the mass at different times.

The noise in the signal was rather high, but could easily be reduced by producing a butterworth lowpass filter that would signal out noise frequencies. A Fast Fourier Transform would produce a frequency spectrum containing both the system oscillation and noise frequency peaks. After normalizing the noise signal to the Nyquist frequency, MatLab built-in functions could filter out exactly the noise frequencies and minimize the loss of the true-signal to filtering. While this worked extremely well, there were certainly limitations inherent to the accelerometer data initially that not even a perfect filter could fix. It was found that found that the accelerometer could not handle accelerations larger than 5g, as shown in the figures from *Appendix IV - Faulty Spring Mass-Drop Data*. This was known going into the lab, but it was still extremely difficult to ensure that accelerometer stayed within this threshold without stopping to analyse the data. Though the data for the foam block drop is clear, the spring drop acceleration was clearly past the accelerometer operating limit. This was corrected using revised drop heights to obtain a new batch of data.

The effects of this overextension of the accelerometer are most clearly recognizable in the position data for the spring drops. As described in the introduction, a cumulative trapezoidal integration function in MatLab was applied to this data set (after removing drift with correction factors from equations 16-21). For foam block drops, this data is remarkably accurate with minimal drift. For spring drops, the first batch of this data followed an erroneous arc pattern, even after removing drift. This is because of the "bottoming out" effect previously described and, as explained in the introduction, these non-constant errors in the acceleration data produce nonlinear drift when integrated. This is not accounted for by the attempt at drift removal because it assumes that drift is constant in the acceleration data.

After collecting data which doesn't breach the acceleration threshold, this method was extremely reliable in correcting the drift of the velocity and position data. This is apparent in the low drift/straightness of the non-oscillating regions of the drop position profiles. If this correction factor were not valid, these regions would clearly show high drift. This indicates reliable position data.

**Spring Constant and Damping Coefficient for Mass-Drop**

The spring constants derived from the initial and final displacements proved to be very consistent, however, there is a discrepancy in the spring constant values of the foam block. This is thought to be because of the fact that, as mass increased, the stress in the foam block at its lowest displacement is high enough to where the linear spring constant assumption breaks down. Foam is not linear in all stress regions[1]. Thus the spring constant would start to increase as the maximum deflection within the nonlinear spring constant region is breached. Even before this region, this value is only semi-linear. Thus, in order to fully characterize the foam block, drops would need to take place at several different heights. A simple compression test with a pull-test fixture would likely be much more well-suited for this application though.

The values derived for the damping coefficient of the spring and foam block are significantly less consistent than that of the spring constant. This is because the derivation for the damping coefficient relies on an exponential fit for time decay constant calculation. This linear fit is only semi-reliable and becomes virtually unusable with the double mass foam block drop. These drops contain only 2 clear peaks which make finding a time decay constant rather difficult. This inconsistency is clearly visible in the damping coefficient data kept in Table 7. This was mitigated by performing logarithmic decrement calculations for these particular drops. The data produced from these calculations appear consistent with those produced from a time decay constant fit.

**Conclusion**

Though the accelerometer had a response and amplitude lag/adjustment to be made, when correlated to strain gauge data, it proved to be a very effective way to acquire acceleration data with high repeatability and frequency.

The mass of the cantilever can have a serious impact on the vibrational profile of the beam. However, with enough added mass, this lab setup is a very reliable way to gather information on the vibrational characteristics of a system. With the exception of the 30g mass data, error in measurements and data collected was low enough to produce reliable and consistent data sets.

Within the operating range of ±4g's of acceleration, the accelerometer is well suited for this experiment. The resultant data is subject to less noise in static and dynamic use than acceleration data obtained from the strain gauge, and provides high resolution readings within the operating band of ±4g's. While this accelerometer is ideal for small scale mass drop

[1] Giampiero Pampolini, Michel Raous. Non linear elasticity, viscosity and damage in open-cell polymeric foams. Archive of Applied Mechanics, Springer Verlag.

experiments, it is not suitable for mass drops with large masses, high drop heights, or stiff springs. If future experiments constitute more trials with these characteristics, an accelerometer with an operating range of ±10g's is more suitable.

**Appendix I: Calibration**

**Table I:** Strain gauge calibration data.

| Tip Deflection (mm) | Sensor Response (V) |
|---|---|
| -38.00 ± 0.01 | -0.61 ± 0.01 |
| -28.00 ± 0.01 | -0.46 ± 0.01 |
| -18.00 ± 0.01 | -0.30 ± 0.01 |
| -8.00 ± 0.01 | -0.15 ± 0.01 |
| 0.00 ± 0.01 | -0.035 ± 0.01 |
| 4.00 ± 0.01 | 0.02 ± 0.01 |
| 14.00 ± 0.01 | 0.17 ± 0.01 |
| 24.00 ± 0.01 | 0.31 ± 0.01 |
| 34.00 ± 0.01 | 0.47 ± 0.01 |
| 44.00 ± 0.01 | 0.61 ± 0.01 |

**Figure I:** Cantilever-mounted strain gauge calibration data with linear fit. $R^2 = 0.9998$

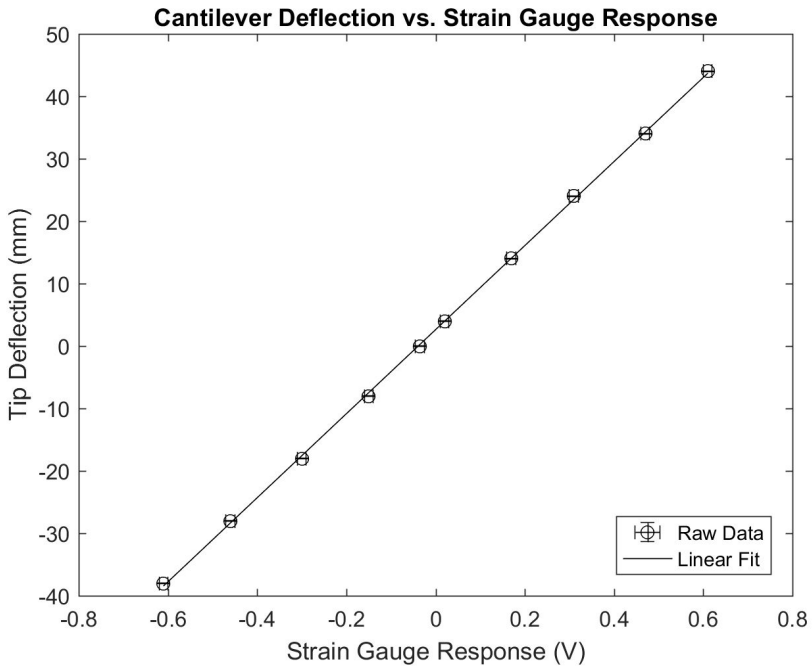
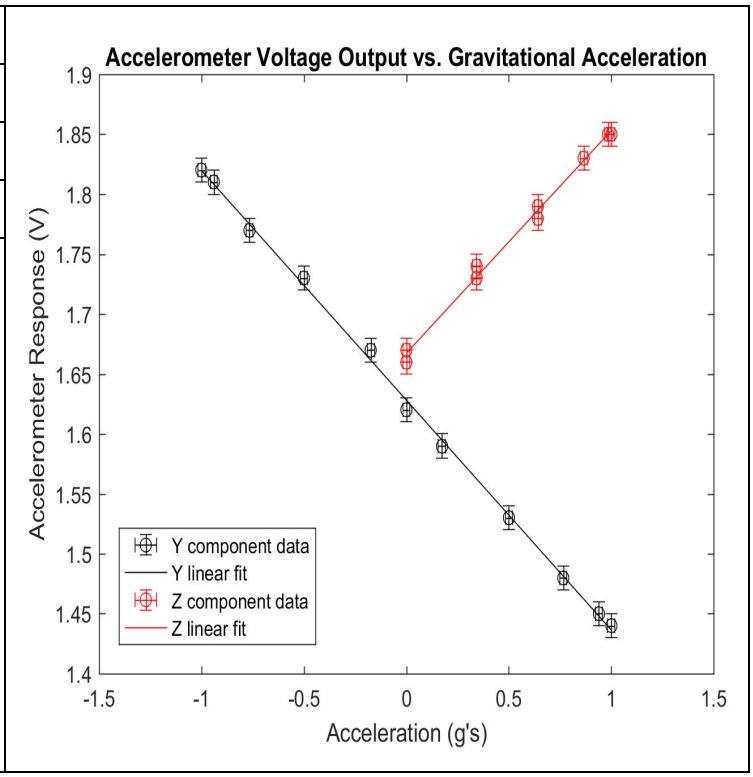



**Table II:** Static accelerometer calibration data.

| Angle (degrees) | Y Axis Response (V) | Z Axis Response (V) |
|---|---|---|
| 0.00 ± 0.02 | 1.44 ± 0.01 | 1.66 ± 0.01 |
| 0.34 ± 0.02 | 1.45 ± 0.01 | 1.73 ± 0.01 |
| 0.64 ± 0.02 | 1.48 ± 0.01 | 1.78 ± 0.01 |
| 0.87 ± 0.02 | 1.53 ± 0.01 | 1.83 ± 0.01 |
| 0.98 ± 0.02 | 1.59 ± 0.01 | 1.85 ± 0.01 |
| 1.00 ± 0.02 | 1.62 ± 0.01 | 1.85 ± 0.01 |

**Figure II:** Static accelerometer calibration data with linear fits. $R_y^2 = 0.9988$, $R_z^2 = 0.9957$

| | | | |
|---|---|---|---|
| 0.98 ± 0.02 | 1.67 ± 0.01 | 1.85 ± 0.01 |  |
| 0.87 ± 0.02 | 1.73 ± 0.01 | 1.83 ± 0.01 | |
| 0.64 ± 0.02 | 1.77 ± 0.01 | 1.79 ± 0.01 | |
| 0.32 ± 0.02 | 1.81 ± 0.01 | 1.74 ± 0.01 | |
| 0.00 ± 0.02 | 1.82 ± 0.01 | 1.67 ± 0.01 | |

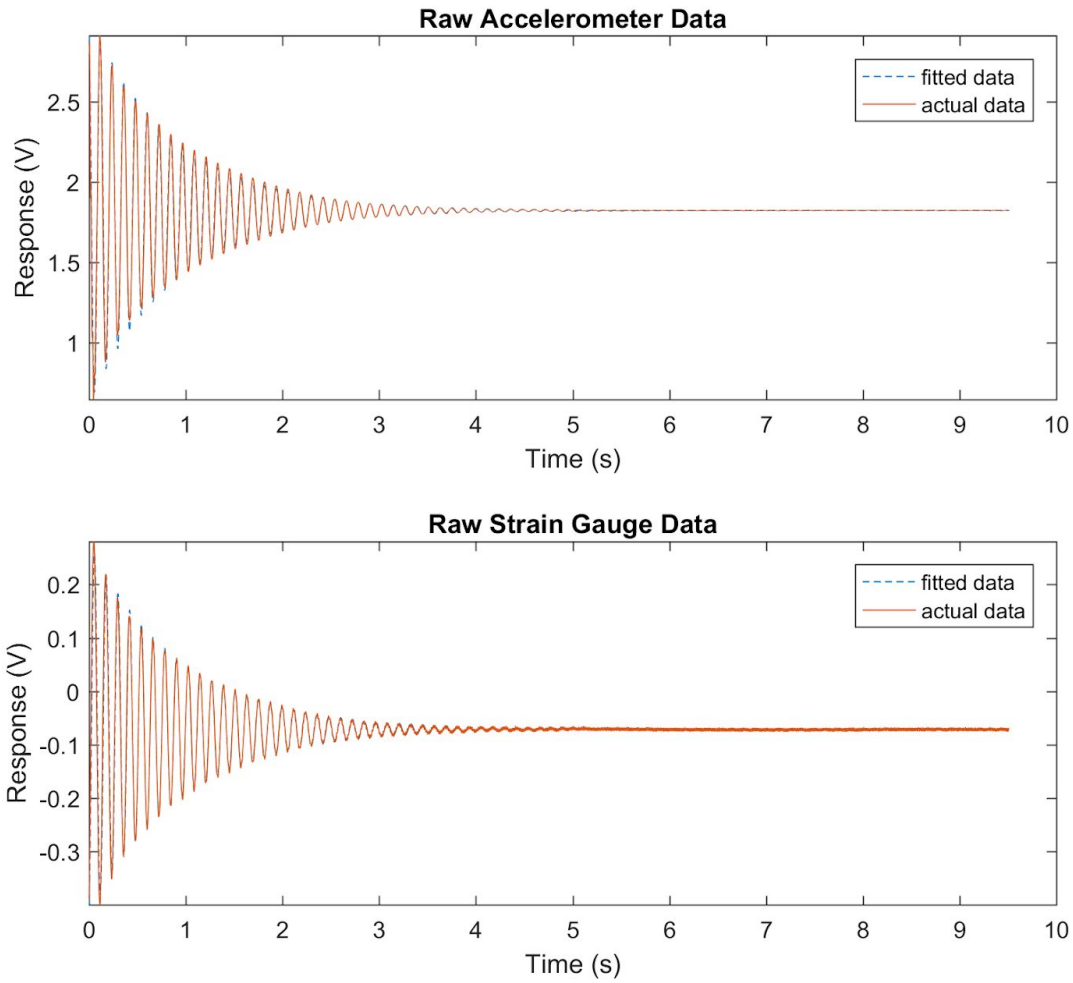



**Figure IIIA-B:** Accelerometer (**A**) and strain gauge (**B**) responses to a 25.62 mm tip deflection with a tip mass of 30.0 g.

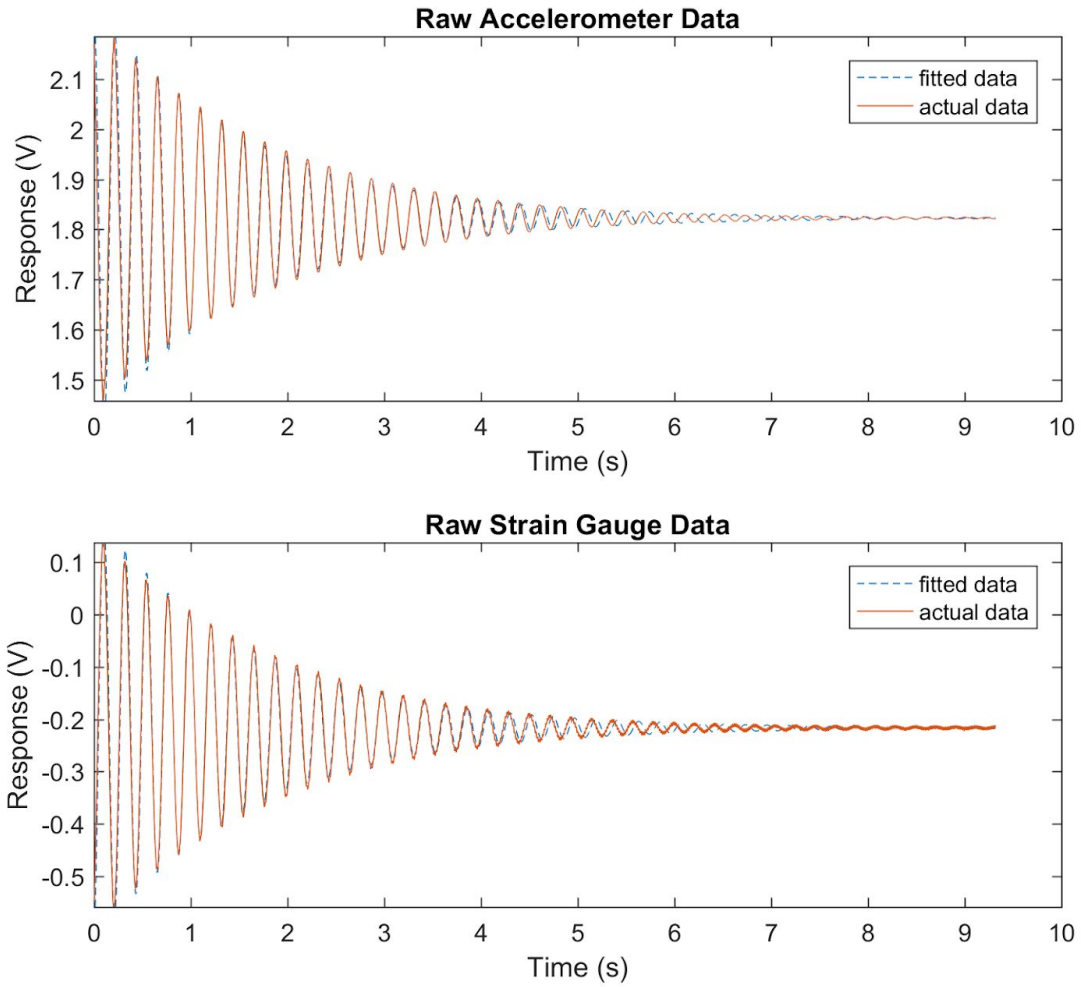



**Figure IVA-B:** Accelerometer (**A**) and strain gauge (**B**) responses to a 26.84 mm tip deflection with a tip mass of 128.8 g.

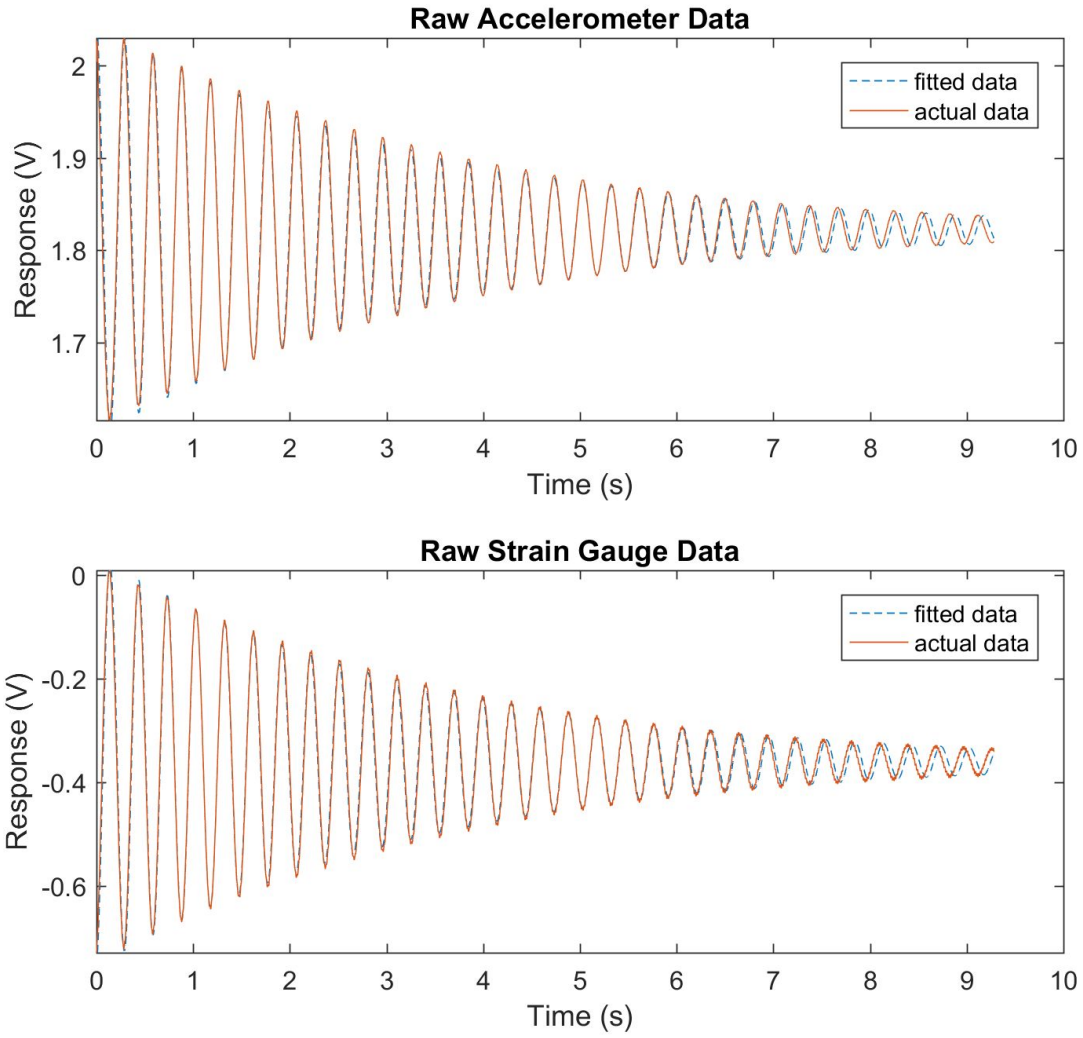



**Figure VA-B:** Accelerometer (**A**) and strain gauge (**B**) responses to a 27.08 mm tip deflection with a tip mass of 227.0 g.

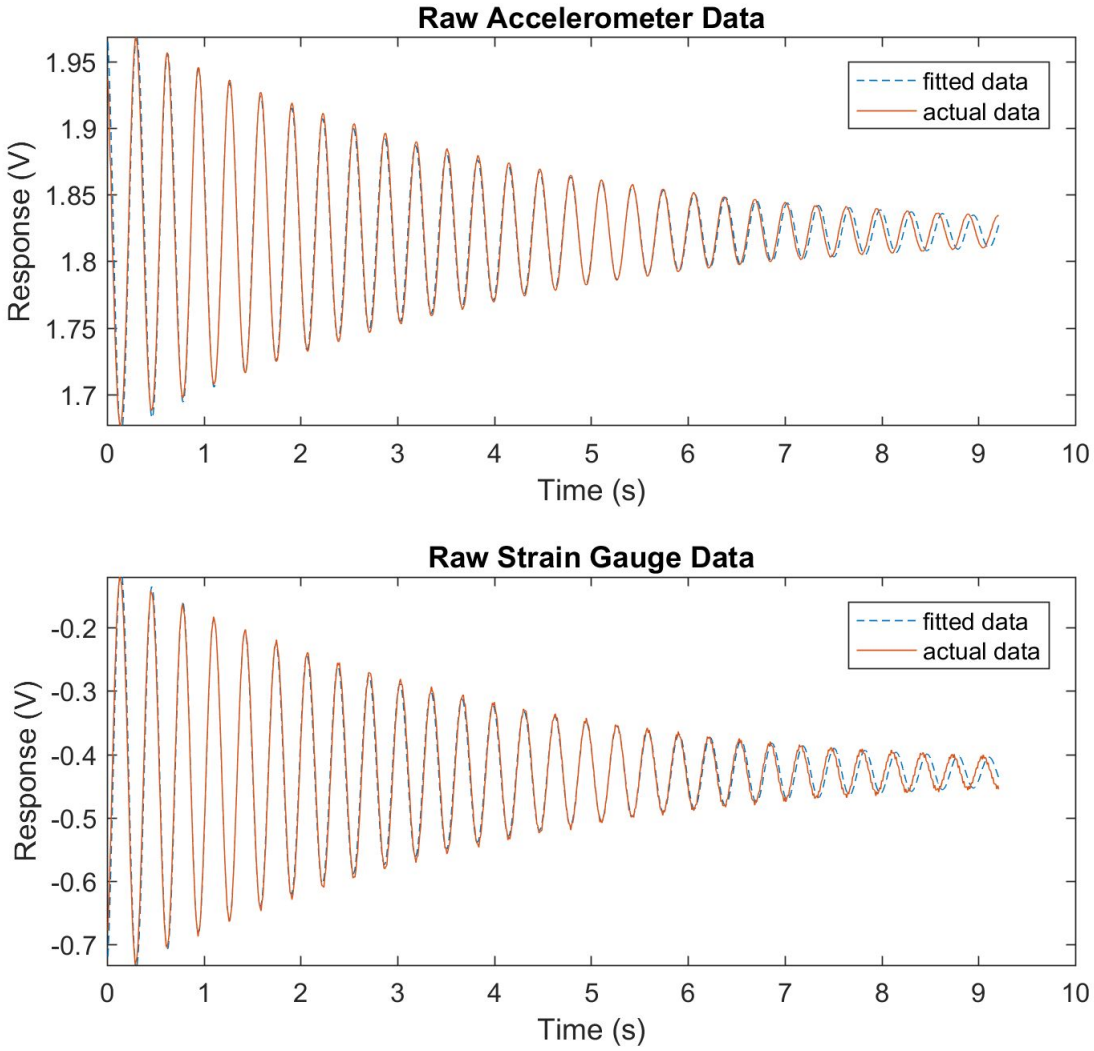



**Figure VIA-B:** Accelerometer (**A**) and strain gauge (**B**) responses to a 22.69 mm tip deflection with a tip mass of 282.3 g.

| **Table III:** Amplitude and phase shift adjustment factors | | | | | |
|---|---|---|---|---|---|
| **Mass (g)** | **Displacement (mm)** | **$\omega_d$ (rad/s)** | **$\tau$ (s)** | **Amplitude factor (m/(Vs^2))** | **Phase adjustment (rad)** |
| 30.0 ± 0.1 | 25.62 ± 0.01 | 53.204 ± 0.001 | 0.481 ± 0.001 | 59.24 ± 0.01 | 0.23 ± 0.03 |
| 30.0 ± 0.1 | 25.62 ± 0.01 | 51.821 ± 0.002 | 0.945 ± 0.005 | 52.41 ± 0.06 | 0.31 ± 0.03 |
| 128.8 ± 0.1 | 26.84 ± 0.01 | 28.424 ± 0.002 | 1.623 ± 0.004 | 52.78 ± 0.1 | 0.20 ± 0.01 |
| 128.8 ± 0.1 | 26.84 ± 0.01 | 28.452 ± 0.001 | 1.540 ± 0.010 | 52.3 ± 0.1 | 0.19 ± 0.01 |
| 227.0 ± 0.1 | 27.08 ± 0.01 | 21.212 ± 0.001 | 3.350 ± 0.010 | 53.4 ± 0.1 | 0.13 ± 0.01 |
| 227.0 ± 0.1 | 27.08 ± 0.01 | 21.213 ± 0.001 | 3.220 ± 0.010 | 53.3 ± 0.1 | 0.15 ± 0.01 |
| 282.3 ± 0.1 | 22.69 ± 0.01 | 19.667 ± 0.001 | 3.211 ± 0.006 | 50.90 ± 0.06 | 0.15 ± 0.01 |
| 282.3 ± 0.1 | 22.69 ± 0.01 | 19.677 ± 0.001 | 3.026 ± 0.005 | 49.80 ± 0.06 | 0.14 ± 0.03 |

## Appendix II: Sources of Uncertainty

### Caliper Uncertainty

The resolution uncertainty of the caliper is ± 0.01mm

### Scale Uncertainty

The resolution uncertainty of the scale is ± 0.1g

## Appendix III: Propagated Uncertainties

### Cantilever Spring Constant

$$\Delta k_{beam} = \sqrt{(2m\Delta\omega_d)^2 + (\Delta m(\omega_d{}^2 + \tfrac{1}{\tau^2}))^2 + (\tfrac{2m}{\tau^3}\Delta\tau)^2}$$

### Moment Area of Inertia

$$\Delta I = \sqrt{(\tfrac{1}{12}h^3\Delta b)^2 + (\tfrac{1}{4}h^2 b\Delta h)^2}$$

### Young's Modulus

$$\Delta E = \sqrt{(\tfrac{L^3}{3I}\Delta k)^2 + (\tfrac{kL^2}{I}\Delta L)^2 + (\tfrac{2kL^3}{3I^2}\Delta I)^2}$$

### Mass Drop Spring Constant

$$\frac{\Delta k}{k} = \sqrt{\left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta g}{g}\right)^2}$$

**Damping coefficient via logarithmic decrement**

$$\frac{\Delta c}{c} = \sqrt{\left(\frac{\Delta k}{2k}\right)^2 + \left(\frac{\Delta m}{2m}\right)^2 + \left(\frac{x1 * \Delta xo}{xo * ln(x0/x1)}\right)^2 + \left(\left(\frac{\Delta x1}{xo * x1 * ln(x0/x1)}\right)^2\right.}$$

**Damping coefficient via time constant**

$$\frac{\Delta c}{c} = \sqrt{\left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta \tau}{\tau}\right)^2}$$

## Appendix IV: Faulty Spring Mass-Drop Data

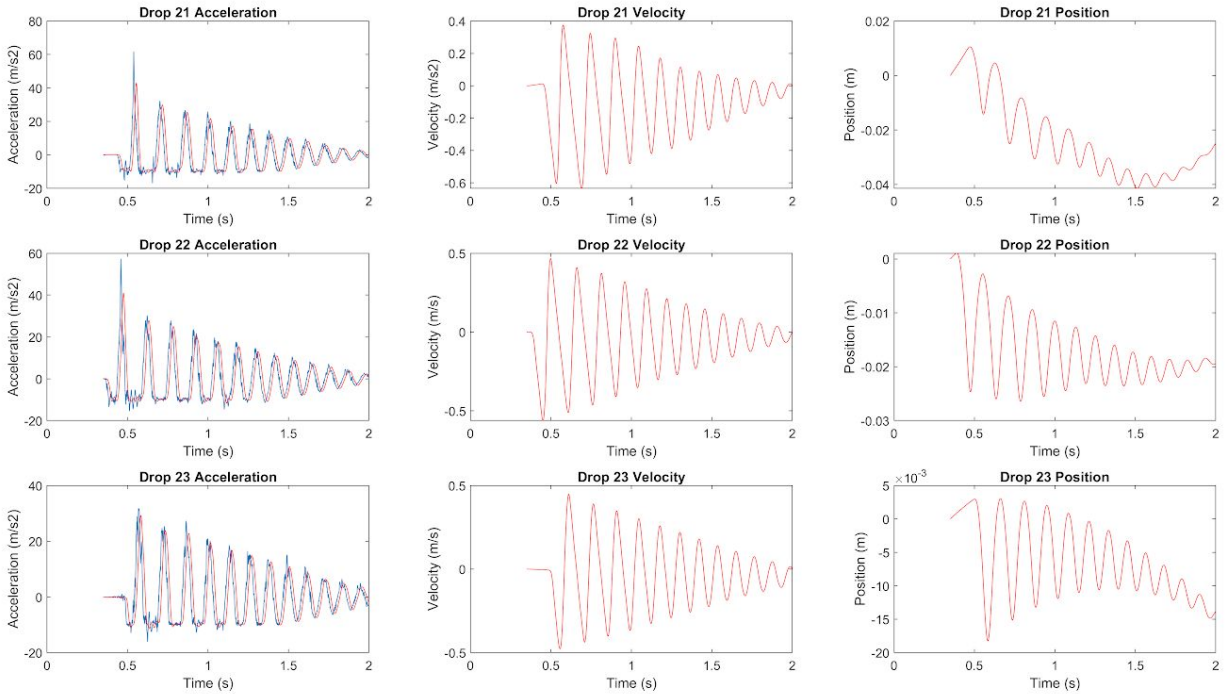



**Figure VII:** Plots of Acceleration, Velocity, and Position for Drops 21-23. These drops are of 190g gram mass and onto a spring.

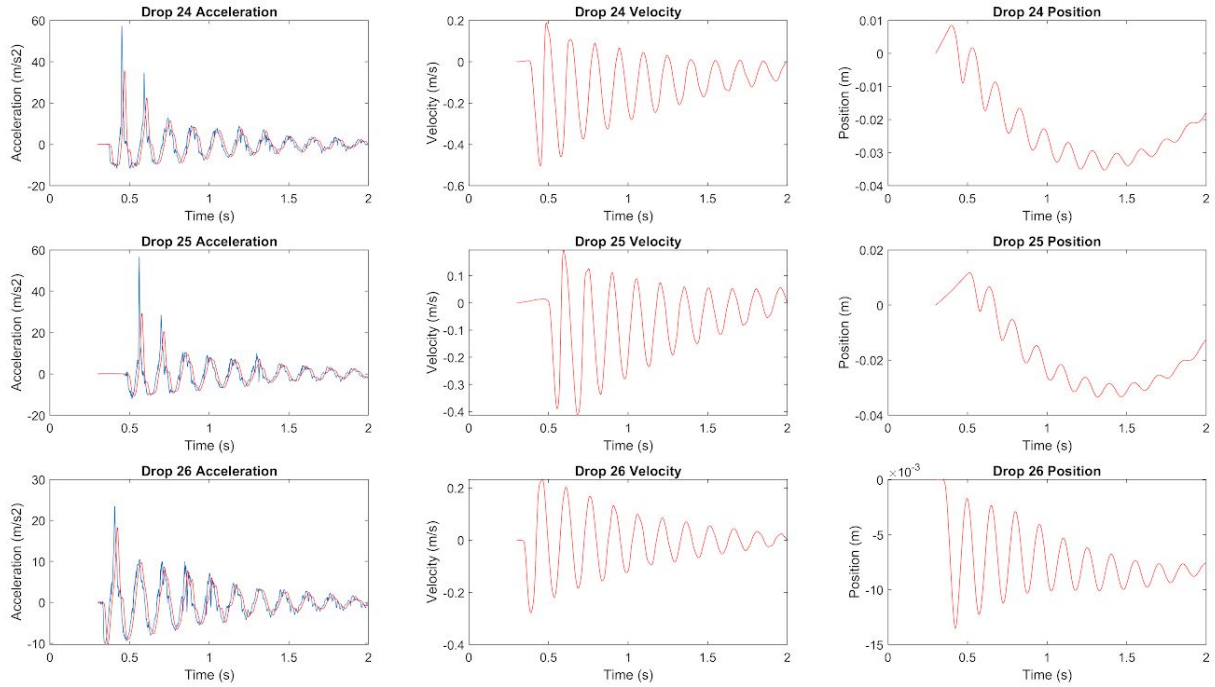



**Figure VIII:** Plots of Acceleration, Velocity, and Position for Drops 24-26. These drops are of 383g mass and onto a spring.

## Appendix V: Cited Sources


I. Giampiero Pampolini, Michel Raous. Non linear elasticity, viscosity and damage in open-cell polymeric foams. Archive of Applied Mechanics, Springer Verlag, 2014, 84 (12), pp.1861-1881. ff10.1007/s00419- 014-0891-5ff. ffhal-01353394f


## Appendix VI: MatLab Integration Code

```
noise = 30;
[b,a] = butter(4,noise/5000,'low');

VCF  = @(Vn)
((Vn(end)-Vn(1))/numel(Vn))*linspace(0,numel(Vn),numel(Vn))+Vn(1);
PCF  = @(Pn,H)
((Pn(end)-Pn(1))/numel(Pn))*linspace(0,numel(Pn),numel(Pn))-(H/numel(Pn))*
linspace(0,numel(Pn),numel(Pn));

%DOUBLE MASS SPRING%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

drop24 = load('drop24.csv');
t24 = drop24(3000:20000,1);
a24 = drop24(3000:20000,3)-mean(drop24(3000:20000,3));
```

```
data24 = [t24 a24];
H24 = -0.01779;

drop25 = load('drop25.csv');
t25 = drop25(3000:20000,1);
a25 = drop25(3000:20000,3)-mean(drop25(3000:20000,3));
data25 = [t25 a25];
H25 = -0.01265;

drop26 = load('drop26.csv');
t26 = drop26(3000:20000,1);
a26 = drop26(3000:20000,3)-mean(drop26(3000:20000,3));
data26 = [t26 a26];
H26 = -0.00758;

%SINGLE MASS SPRING%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

drop21 = load('drop21.csv');
t21 = drop21(3500:20000,1);
a21 = drop21(3500:20000,3)-mean(drop21(3500:20000,3));
data21 = [t21 a21];
H21 = -0.02483;

drop22 = load('drop22.csv');
t22 = drop22(3500:20000,1);
a22 = drop22(3500:20000,3)-mean(drop22(3500:20000,3));
data22 = [t22 a22];
H22 = -0.01934;

drop23 = load('drop23.csv');
t23 = drop23(3500:20000,1);
a23 = drop23(3500:20000,3)-mean(drop23(3500:20000,3));
data23 = [t23 a23];
H23 = -0.01387;

%DOUBLE MASS FOAM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

drop14 = load('drop14.csv');
t14 = drop14(4400:13000,1);
a14 = drop14(4400:13000,3)-mean(drop14(4400:13000,3));
data14 = [t14 a14];
H14 = -0.045;

drop15 = load('drop15.csv');
t15 = drop15(5400:14000,1);
```

```
a15 = drop15(5400:14000,3)-mean(drop15(5400:14000,3));
data15 = [t15 a15];
H15 = -0.051;

drop16 = load('drop16.csv');
t16 = drop16(5400:12500,1);
a16 = drop16(5400:12500,3)-mean(drop16(5400:12500,3));
data16 = [t16 a16];
H16 = -0.0406;

%SINGLE MASS FOAM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

drop17 = load('drop17.csv');
t17 = drop17(4000:8300,1);
a17 = drop17(4000:8300,3)-mean(drop17(4000:8300,3));
data17 = [t17 a17];
H17 = -0.02934;

drop18 = load('drop18.csv');
t18 = drop18(6200:11000,1);
a18 = drop18(6200:11000,3)-mean(drop18(6200:11000,3));
data18 = [t18 a18];
H18 = -0.0336;

drop19 = load('drop19.csv');
t19 = drop19(4500:10000,1);
a19 = drop19(4500:10000,3)-mean(drop19(4500:10000,3));
data19 = [t19 a19];
H19 = -0.04;

%FFT FOR FINDING NOISE FREQ%%%%%%%%%%%%%%%%%%%

% Nf14 =
% Nf15 =
% Nf16 =
% Nf17 =
% Nf18 =
% Nf19 =
% Nf21 =
% Nf22 =
% Nf23 =
% Nf24 =
% Nf25 =
% Nf26 =
```

```
%FREQUENCY OF OSCILLATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Of14 = 34.56;
Of15 = 37.7;
Of16 = 37.7;
Of17 = 31.42;
Of18 = 31.42;
Of19 = 31.42;
Of21 = 43.98;
Of22 = 43.98;
Of23 = 47.1;
Of24 = 40.84;
Of25 = 40.84;
Of26 = 40.84;

%CONVERT VOLTAGE TO ACCELERATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A14  = a14*53;
A15  = a15*53;
A16  = a16*53;
A17  = a17*53;
A18  = a18*53;
A19  = a19*53;
A21  = a21*53;
A22  = a22*53;
A23  = a23*53;
A24  = a24*53;
A25  = a25*53;
A26  = a26*53;

%FILTERING NOISE OUT OF DATA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

AY14 = filter(b,a,A14);
AY15 = filter(b,a,A15);
AY16 = filter(b,a,A16);
AY17 = filter(b,a,A17);
AY18 = filter(b,a,A18);
AY19 = filter(b,a,A19);
AY21 = filter(b,a,A21);
AY22 = filter(b,a,A22);
AY23 = filter(b,a,A23);
AY24 = filter(b,a,A24);
AY25 = filter(b,a,A25);
AY26 = filter(b,a,A26);
```

```
%NUMERICALLY INTEGRATE DATA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%14-16
Vn14 = cumtrapz(t14,AY14);
VY14  = Vn14-VCF(Vn14)';

Pn14 = cumtrapz(t14,VY14);
PY14  = Pn14-PCF(Pn14,H14)';

Vn15 = cumtrapz(t15,AY15);
VY15  = Vn15-VCF(Vn15)';

Pn15 = cumtrapz(t15,VY15);
PY15  = Pn15-PCF(Pn15,H15)';

Vn16 = cumtrapz(t16,AY16);
VY16  = Vn16-VCF(Vn16)';

Pn16 = cumtrapz(t16,VY16);
PY16  = Pn16-PCF(Pn16,H16)';

%17-19

Vn17 = cumtrapz(t17,AY17);
VY17  = Vn17-VCF(Vn17)';

Pn17 = cumtrapz(t17,VY17);
PY17  = Pn17-PCF(Pn17,H17)';

Vn18 = cumtrapz(t18,AY18);
VY18  = Vn18-VCF(Vn18)';

Pn18 = cumtrapz(t18,VY18);
PY18  = Pn18-PCF(Pn18,H18)';

Vn19 = cumtrapz(t19,AY19);
VY19  = Vn19-VCF(Vn19)';

Pn19 = cumtrapz(t19,VY19);
PY19  = Pn19-PCF(Pn19,H19)';

%21-23

Vn21 = cumtrapz(t21,AY21);
VY21  = Vn21-VCF(Vn21)';
```

```
Pn21 = cumtrapz(t21,VY21);
PY21  = Pn21-PCF(Pn21,H21)';

Vn22 = cumtrapz(t22,AY22);
VY22  = Vn22-VCF(Vn22)';

Pn22 = cumtrapz(t22,VY22);
PY22  = Pn22-PCF(Pn22,H22)';

Vn23 = cumtrapz(t23,AY23);
VY23  = Vn23-VCF(Vn23)';

Pn23 = cumtrapz(t23,VY23);
PY23  = Pn23-PCF(Pn23,H23)';

%24-26

Vn24 = cumtrapz(t24,AY24);
VY24  = Vn24-VCF(Vn24)';

Pn24 = cumtrapz(t24,VY24);
PY24  = Pn24-PCF(Pn24,H24)';

Vn25 = cumtrapz(t25,AY25);
VY25  = Vn25-VCF(Vn25)';

Pn25 = cumtrapz(t25,VY25);
PY25  = Pn25-PCF(Pn25,H25)';

Vn26 = cumtrapz(t26,AY26);
VY26  = Vn26-VCF(Vn26)';

Pn26 = cumtrapz(t26,VY26);
PY26  = Pn26-PCF(Pn26,H26)';

%PLOTTTTTTTINGINGINGINGING%%%%%%%%%%%%%%%%%%%%%

%14-16
figure(1)

subplot(3,3,1)
plot(t14,A14)
hold on
plot(t14,AY14,'r')
```

```
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 14 Acceleration')

subplot(3,3,2)
plot(t14,VY14,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 14 Velocity')

subplot(3,3,3)
plot(t14,PY14,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 14 Position')

subplot(3,3,4)
plot(t15,A15)
hold on
plot(t15,AY15,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 15 Acceleration')

subplot(3,3,5)
plot(t15,VY15,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 15 Velocity')

subplot(3,3,6)
plot(t15,PY15,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 15 Position')

subplot(3,3,7)
plot(t16,A16)
hold on
plot(t16,AY16,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 16 Acceleration')

subplot(3,3,8)
```

```
plot(t16,VY16,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 16 Velocity')

subplot(3,3,9)
plot(t16,PY16,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 16 Position')

%17-19
figure(2)

subplot(3,3,1)
plot(t17,A17)
hold on
plot(t17,AY17,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 17 Acceleration')

subplot(3,3,2)
plot(t17,VY17,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 17 Velocity')

subplot(3,3,3)
plot(t17,PY17,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 17 Position')

subplot(3,3,4)
plot(t18,A18)
hold on
plot(t18,AY18,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 18 Acceleration')

subplot(3,3,5)
plot(t18,VY18,'r')
xlabel('Time (s)')
```

```
ylabel('Velocity (m/s)')
title('Drop 18 Velocity')

subplot(3,3,6)
plot(t18,PY18,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 18 Position')

subplot(3,3,7)
plot(t19,A19)
hold on
plot(t19,AY19,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 19 Acceleration')

subplot(3,3,8)
plot(t19,VY19,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 19 Velocity')

subplot(3,3,9)
plot(t19,PY19,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 19 Position')

%21-23
figure(3)

subplot(3,3,1)
plot(t21,A21)
hold on
plot(t21,AY21,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 21 Acceleration')

subplot(3,3,2)
plot(t21,VY21,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s2)')
title('Drop 21 Velocity')
```

```
subplot(3,3,3)
plot(t21,PY21,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 21 Position')

subplot(3,3,4)
plot(t22,A22)
hold on
plot(t22,AY22,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 22 Acceleration')

subplot(3,3,5)
plot(t22,VY22,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 22 Velocity')

subplot(3,3,6)
plot(t22,PY22,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 22 Position')

subplot(3,3,7)
plot(t23,A23)
hold on
plot(t23,AY23,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 23 Acceleration')

subplot(3,3,8)
plot(t23,VY23,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 23 Velocity')

subplot(3,3,9)
plot(t23,PY23,'r')
xlabel('Time (s)')
ylabel('Position (m)')
```

```
title('Drop 23 Position')

%24-26
figure(4)

subplot(3,3,1)
plot(t24,A24)
hold on
plot(t24,AY24,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 24 Acceleration')

subplot(3,3,2)
plot(t24,VY24,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 24 Velocity')

subplot(3,3,3)
plot(t24,PY24,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 24 Position')

subplot(3,3,4)
plot(t25,A25)
hold on
plot(t25,AY25,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 25 Acceleration')

subplot(3,3,5)
plot(t25,VY25,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 25 Velocity')

subplot(3,3,6)
plot(t25,PY25,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 25 Position')
```

```
subplot(3,3,7)
plot(t26,A26)
hold on
plot(t26,AY26,'r')
xlabel('Time (s)')
ylabel('Acceleration (m/s2)')
title('Drop 26 Acceleration')

subplot(3,3,8)
plot(t26,VY26,'r')
xlabel('Time (s)')
ylabel('Velocity (m/s)')
title('Drop 26 Velocity')

subplot(3,3,9)
plot(t26,PY26,'r')
xlabel('Time (s)')
ylabel('Position (m)')
title('Drop 26 Position')

%SPRING CONSTANT CALCULATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m14 = 383; m15 = 383; m16 = 383;
m17 = 383; m18 = 383; m19 = 383;
m21 = 383; m22 = 383; m23 = 383;
m24 = 383; m25 = 383; m26= 383;

F  = [m14 m15 m16 m17 m18 m19 m21 m22 m23 m24 m25 m26]*9.81*(10^(-3));
xi = [50 50 50 50 50 50 19.15 19.15 19.15 19.15 19.15 19.15]*(10^(-3));
xf = [15 14 14.4 25.66 26.4 25 15.17 15.66 16.13 12.21 12.35
12.42]*(10^(-3));
dx = xi-xf;
k  = F./dx

dk = zeros(numel(F),1);
dk(:,1) = k.*sqrt((0.1/(F(:,1)./(0.001*9.81))).^2 + (0.00001/dx(:,1)).^2);

%DAMPING RATIO CALCULATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Find Peaks

[pks14,locs14] = findpeaks(PY14,t14);
[pks15,locs15] = findpeaks(PY15,t15);
[pks16,locs16] = findpeaks(PY16,t16);
[pks17,locs17] = findpeaks(PY17,t17);
[pks18,locs18] = findpeaks(PY18,t18);
```

```
[pks19,locs19] = findpeaks(PY19,t19);
[pks21,locs21] = findpeaks(PY21,t21);
[pks22,locs22] = findpeaks(PY22,t22);
[pks23,locs23] = findpeaks(PY23,t23);
[pks24,locs24] = findpeaks(PY24,t24);
[pks25,locs25] = findpeaks(PY25,t25);
[pks26,locs26] = findpeaks(PY26,t26);

pks17 = [AY17(1); pks17];
locs17 = [t17(1); locs17];

% indices = find(abs(pks16)<0.001);
% locs16(indices) = [];
% pks16(indices) = [];

pks14 = pks14-min(pks14);
pks15 = pks15-min(pks15);
pks16 = pks16-min(pks16);
pks17 = pks17-min(pks17);
pks18 = pks18-min(pks18);
pks19 = pks19-min(pks19);
pks21 = pks21-min(pks21);
pks22 = pks22-min(pks22);
pks23 = pks23-min(pks23);
pks24 = pks24-min(pks24);
pks25 = pks25-min(pks25);
pks26 = pks26-min(pks26);

%Fitting a Logarithmic Function

[f14,gof14] = fit(locs14,pks14,'exp1');
[f15,gof15] = fit(locs15,pks15,'exp1');
[f16,gof16] = fit(locs16,pks16,'exp1');
[f17,gof17] = fit(locs17,pks17,'exp1');
[f18,gof18] = fit(locs18,pks18,'exp1');
[f19,gof19] = fit(locs19,pks19,'exp1');
[f21,gof21] = fit(locs21,pks21,'exp1');
[f22,gof22] = fit(locs22,pks22,'exp1');
[f23,gof23] = fit(locs23,pks23,'exp1');
[f24,gof24] = fit(locs24,pks24,'exp1');
[f25,gof25] = fit(locs25,pks25,'exp1');
[f26,gof26] = fit(locs26,pks26,'exp1');

%errors
```

```
error14 = ones(numel(pks14),1)*gof14.rmse;
error15 = ones(numel(pks15),1)*gof15.rmse;
error16 = ones(numel(pks16),1)*gof16.rmse;
error17 = ones(numel(pks17),1)*gof17.rmse;
error18 = ones(numel(pks18),1)*gof18.rmse;
error19 = ones(numel(pks19),1)*gof19.rmse;
error21 = ones(numel(pks21),1)*gof21.rmse;
error22 = ones(numel(pks22),1)*gof22.rmse;
error23 = ones(numel(pks23),1)*gof23.rmse;
error24 = ones(numel(pks24),1)*gof24.rmse;
error25 = ones(numel(pks25),1)*gof25.rmse;
error26 = ones(numel(pks26),1)*gof26.rmse;

figure(5)

subplot(1,3,1)
errorbar(locs14,pks14,-error14,error14,'x')
hold on
plot(f14)
title('Drop 14')
xlabel('Time 14')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof14.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f14.b)])
subplot(1,3,2)
errorbar(locs15,pks15,-error15,error15,'x')
hold on
plot(f15)
title('Drop 15')
xlabel('Time 15')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof15.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f15.b)])
subplot(1,3,3)
errorbar(locs16,pks16,-error16,error16,'x')
hold on
plot(f16)
title('Drop 16')
xlabel('Time 16')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof16.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f16.b)])

figure(6)
```

```
subplot(1,3,1)
errorbar(locs17,pks17,-error17,error17,'x')
hold on
plot(f17)
title('Drop 17')
xlabel('Time 17')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof17.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f17.b)])
subplot(1,3,2)
errorbar(locs17,pks17,-error17,error17,'x')
hold on
plot(f18)
title('Drop 18')
xlabel('Time 18')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof18.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f18.b)])
subplot(1,3,3)
errorbar(locs19,pks19,-error19,error19,'x')
hold on
plot(f19)
title('Drop 19')
xlabel('Time 19')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof19.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f19.b)])

figure(7)

subplot(1,3,1)
errorbar(locs21,pks21,-error21,error21,'x')
hold on
plot(f21)
title('Drop 21')
xlabel('Time 21')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof21.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f21.b)])
subplot(1,3,2)
errorbar(locs22,pks22,-error22,error22,'x')
hold on
plot(f22)
title('Drop 22')
xlabel('Time 22')
```

```
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof22.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f22.b)])
subplot(1,3,3)
errorbar(locs23,pks23,-error23,error23,'x')
hold on
plot(f23)
title('Drop 23')
xlabel('Time 23')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof23.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f23.b)])

figure(8)

subplot(1,3,1)
errorbar(locs24,pks24,-error24,error24,'x')
hold on
plot(f24)
title('Drop 24')
xlabel('Time 24')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof24.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f24.b)])
subplot(1,3,2)
errorbar(locs25,pks25,-error25,error25,'x')
hold on
plot(f25)
title('Drop 25')
xlabel('Time 25')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof25.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f25.b)])
subplot(1,3,3)
errorbar(locs26,pks26,-error26,error26,'x')
hold on
plot(f26)
title('Drop 26')
xlabel('Time 26')
ylabel('Position (m)')
legend('Raw Data',['Fitted Curve' char(10) ' R2 = ' num2str(gof26.rsquare)
char(10) 'Time Decay Constant = ' num2str(1/f26.b)])

%Calculation Damping Coefficient from Fits
```

```
t14 = -1/f14.b;
t15 = -1/f15.b;
t16 = -1/f16.b;
t17 = -1/f17.b;
t18 = -1/f18.b;
t19 = -1/f19.b;
t21 = -1/f21.b;
t22 = -1/f22.b;
t23 = -1/f23.b;
t24 = -1/f24.b;
t25 = -1/f25.b;
t26 = -1/f26.b;

t = [t14 t15 t16 t17 t18 t19 t21 t22 t23 t24 t25 t26];
m = [m14 m15 m16 m17 m18 m19 m21 m22 m23 m24 m25 m26];

c14 = 2*m14/t14;
c15 = 2*m15/t15;
c16 = 2*m16/t16;
c17 = 2*m17/t17;
c18 = 2*m18/t18;
c19 = 2*m19/t19;
c21 = 2*m21/t21;
c22 = 2*m22/t22;
c23 = 2*m23/t23;
c24 = 2*m24/t24;
c25 = 2*m25/t25;
c26 = 2*m26/t26;

c = 2*m./t;
```