LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	**LIGO-T1600150-v1**	September 23 2016

Final Report: Acoustic Emissions in Metals

Danielle Frostig, Gabriele Vajente

Distribution of this document:

LIGO Scientific Collaboration

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW17-161
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

1 Abstract

Crackling noise occurs when a system responds to slowly changing external conditions with discrete energy events at higher frequencies. This effect may be significant in the maraging steel blades used in the suspension system of Advanced LIGO. Vertical displacement noise caused by crackling events could couple with horizontal displacement noise along the path of the interferometer, influencing Advanced LIGO's sensitivity. The goal of this project is to continue a previous experiment aimed at directly detecting energy released from crackling events in the form of acoustic waves. This consists mainly of calibrating the output of ultrasonic acoustic emission microphones in terms of energy released into the blades in a crackling event. The microphones are calibrated via a simple ball drop experiment to compare the known energy of a system to the energy in the microphones. The microphones will be used to search for crackling events in loaded maraging steel blades modulated with a drive of variable frequencies and amplitudes. The ultimate goal of the project is to find an upper limit on crackling noise, even if no crackling events are directly detected.

2 Motivation

2.1 LIGO and Maraging Steel Blades

The Large Interferometer Gravitational-Wave Observatory (LIGO) is a ground-based system of large-scale detectors designed to observe gravitational waves. The experiment uses an enhanced Michelson interferometer to measure the distance between test masses in order to detect minute ripples in space-time caused by gravitational waves. The Advanced LIGO detectors must be extremely sensitive in order to successfully detect gravitational waves. For example, at the low frequency end of the audio band, between 10 and 20 Hz, the horizontal motion of the test masses needs to be on the order $10^{-19} m/\sqrt{Hz}$ [1]. In order to detect this displacement–which is four orders of magnitude smaller than a proton–within a 4 km long detector arm, complex noise isolation systems are employed.

Local seismic activity is a prominent source of noise in the ground-based detectors. In order to isolate the test mass system from this noise, a quadruple pendulum isolates the system horizontally and three levels of maraging steel cantilever spring pairs isolate it vertically (Figure 1). Noise originating in the cantilevers can propagate throughout the system, especially in the upper intermediate stage (UIM), and cause vertical displacement in the system. Due to the curvature of the Earth, the test masses, hanging along the pull of Earth's gravity, must undergo corrections to be held truly parallel to each other [2]. Consequently, the "vertical" displacements become coupled with horizontal displacement and becomes a relevant source of noise in the horizontal direction that could potentially obscure gravitational wave signals.

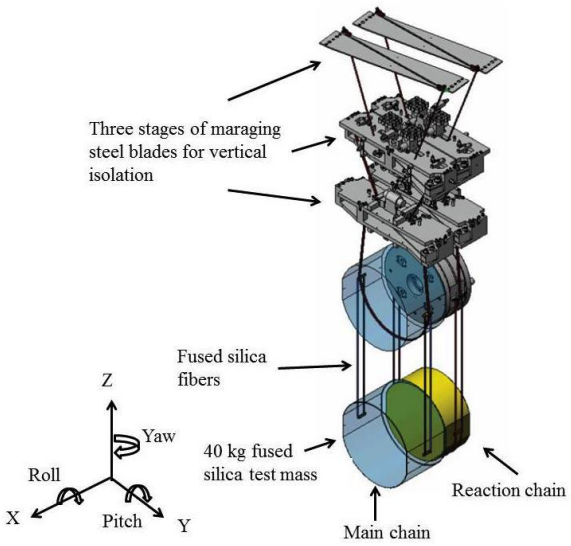

Figure 1: The Advanced LIGO suspension system, featuring a quadruple pendulum horizontal isolation system and three stages or maraging steel cantilever springs. The diagram is used courtesy of [7].

2.2 Crackling Noise

Crackling noise results when a system subject to slowly changing external conditions responds via discrete crackling events due to a nonlinear conversion of energy [3]. This noise can occur in a variety of systems, from earthquakes on fault lines to stock market fluctuations [4]. However, not all systems respond to external force through crackling noise. The nonlinear conversion of energy in crackling events falls between the limits of a system responding to external forces in a single event–such as a piece of chalk snapping in half–and a system responding with small, similar sized events–such as popcorn popping. A simple example of crackling noise occurs when a sheet of paper is crumpled. Slowly changing external force (e.g. a hand crumpling the paper) causes the system (paper) to respond with a nonlinear conversion of energy (the crumpling sound heard) [4].

Ideally, systems could be assumed to behave elastically, meaning the resultant strain exhibited is proportional to the stress applied. However, non-linear deviations occur in systems loaded past yield stress. Consequently, metal deformations could occur through discrete releases of strain, and therefore cause crackling noise. There is a possibility of non-linear up-conversion of low frequency excitations (below 1 Hz) to high frequency (audio) noise in elastic metals such as maraging steel. Crackling noise has been detected in the plastic regime and so far there has been no direct measurement of mechanically up-converted noise in the elastic regime [1].

The goal of the investigation is to experimentally detect crackling noise in the maraging steel blades used in the Advanced LIGO experiment. The cantilevers used are in the elastic regime at about 50% of the yield stress. It is expected that low frequency motion in the suspension could result in a non-linear up-conversion of displacement noise. These events are further expected to be time-correlated with the stress or stress rate of the low frequency motion. Consequently, the project could detect not only crackling noise events, but could also find an increased rate of events

with increased driving force or rate of force [1].

2.3 Previous Work

The goal of the project is to directly detect crackling events in maraging steel. Additionally, even if the experiment does not ultimately detect crackling events, it can help set upper limits for the amplitude and rate of such events. Two experiment types have been designed previously in order to detect crackling noise in maraging steel: one based on interferometry and one on ultrasonic Acoustic Emission (AE). Both experiments are being run simultaneously and this project focuses on the AE approach.

In the interferometric experiment, a pair of blade springs supporting optical instruments in a Michelson interferometer are driven by a low frequency force. If a discrete crackling event occurs, a resultant displacement should occur between the detected signals in each blade [1]. This experimental design is limited because the blade acts as a low-pass filter, which can obscure some of the noise. Additionally, in order to keep the optical system close the operating point, the maximum blade motion is limited to few tens of microns. A potential alternative is to use ultrasonic AE microphones on blades loaded with a range of weights. Although the ultrasonic microphones are not as sensitive as the interferometer, they can be employed at higher frequencies to possibly directly detect single crackling events. Previously, this experiment was run with a variety of materials stressed with a range of weights and no crackling events were detected [5].

In the first set of tests, a maraging steel blade was loaded with an 11 kg mass, 50% of the maximum weight. An attached voice-coil actuator was driven with a DAC through an amplifier. Additionally, passive low-pass filters were used to prevent mistaking high-frequency events from the power amplifier for crackling events [5].

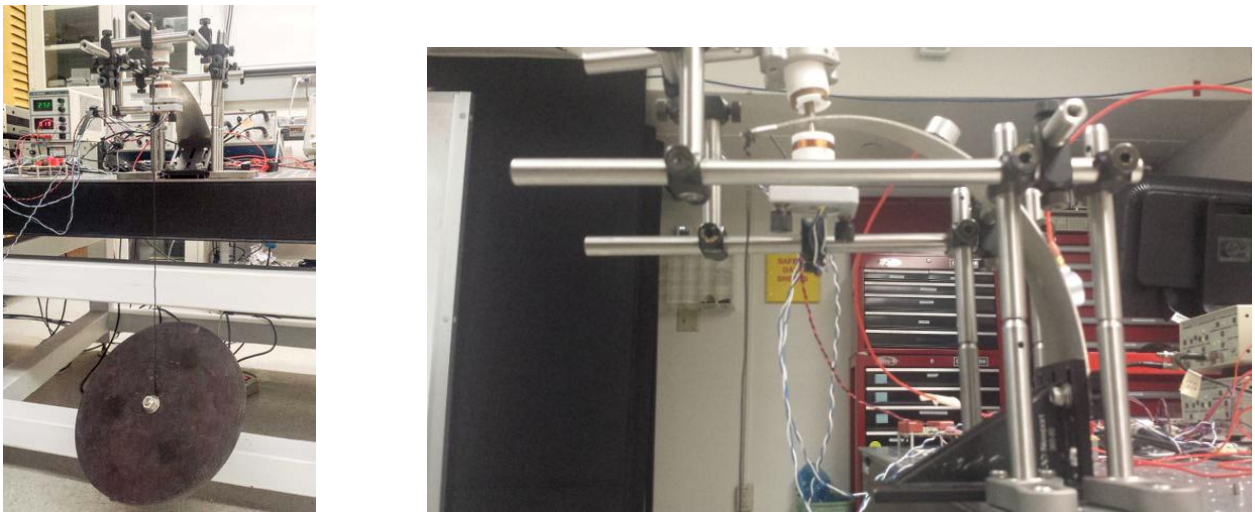

Figure 2: Maraging steel blade loaded with an 11 kg mass (right) and 16 kg mass (left).

The system was excited with a sinusoidal displacement of about 500 um at 0.4 Hz every other hour, with the drive off for the alternate hours. No significant change was measured between the on and

off periods, implying no crackling noise was excited by the low frequency drive [3].

Lack of evidence of crackling noise was similarly found in a blade loaded with 16 kg, 75% of the nominal yield stress, and clamped vertically. Despite days of data, no difference was detected between on and off periods. Similarly, the experiment yielded the same results when repeated with brass or high carbon steel blades. The results imply any crackling noise that could have occurred did so below the noise floor. An upper limit on noise of $10^{-15}\frac{m}{\sqrt{Hz}}$ from 30 kHz to a few hundred kHz was set in maraging and high carbon steel blades [3].

3 Microphone Calibration

Score Dunegan microphones [6] are used to collect the ultrasonic AE data for this experiment. These piezoelectric contact ultrasonic microphones can detect in-plane and out-of-plane waves in the range of hundreds of kilohertz [6]. They are secured directly onto the blades with an incompressible medium, typically wax. One goal of the project is to calibrate the microphones' output to the energy released in the blade, a more useful output for the sake of the experiment. The most straightforward way to calibrate the microphones is to compare their output with the known energy release of a simple system – in this case, dropping a small ball onto the blade. By comparing the energy just before and after one bounce, the energy released into the blade can be approximated and used to calibrate the microphone.

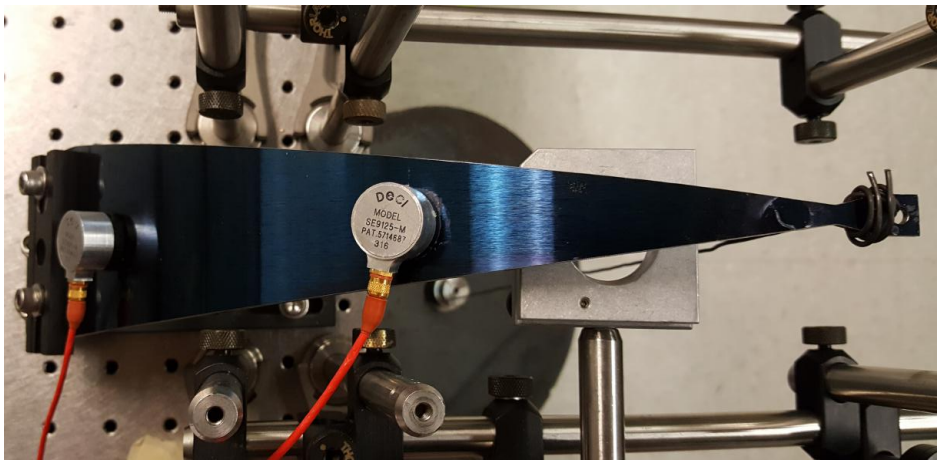

Figure 3: Score Dunegan microphones secured onto a stressed maraging steel blade with wax.

3.1 Motion-Tracking

With the resources available, the most accurate and efficient way to measure the height of the ball after one bounce is to video record the experimental setup and employ motion-tracking software. A simple ball drop experiment was set up in an office consisting of a General Relativity textbook, a binder clip, and a pen spring, which was later replaced by small steel balls ranging from 3/64" to

Figure 4: Early experimental setups for testing motion-tracking. A first iteration of the experiment (right) demonstrates the binder clip and coffee stirrer release mechanisms. A second iteration of the experiment (left) has a monochromatic background and demonstrates motion-tracking. The colors have been inverted for ease of viewing.

1/4" in diameter (Figure 4a). Videos of the experiment were taken with a Samsung Galaxy Note 5 and analyzed using *Tracker*, a free motion-tracking software.[1] The program could track a ball with relative success, but could not export the data in a useful format.

Consequently, MATLAB was employed for the motion-tracking and analysis. The script is based on Kalman filtering for predicting and tracking the motion of the object (See appendix). The code was written based on the kalmanFilterForTracking[2] function in MATLAB and built from there.

The motion-tracking software is based on tracking the color of an object, which initially posed minor difficulty as the first experimental setup contained multiple objects that are silver in color. Consequently, a second ball drop experiment was set up with a completely black backdrop composed of a router and another spare computer part (Figure 4b). The MATLAB script ran with increased success with the new background.

3.1.1 Motion-Tracking Data Analysis

A Samsung Galaxy Note 5 was used for recording the motion of the ball. The slow motion video capture was recorded at 120 frames per second (fps) and a duplicate still frame was added in between each frame, for a total of 240 fps. Consequently, only every other video frame was used as to not repeat data points.

In addition to collecting data for the position of the ball, velocity and acceleration were also an-

[1] *Tracker* can be found at http://physlets.org/tracker/

[2] The function and documentation can be found at http://www.mathworks.com/help/vision/examples/using-kalman-filter-for-object-tracking.html

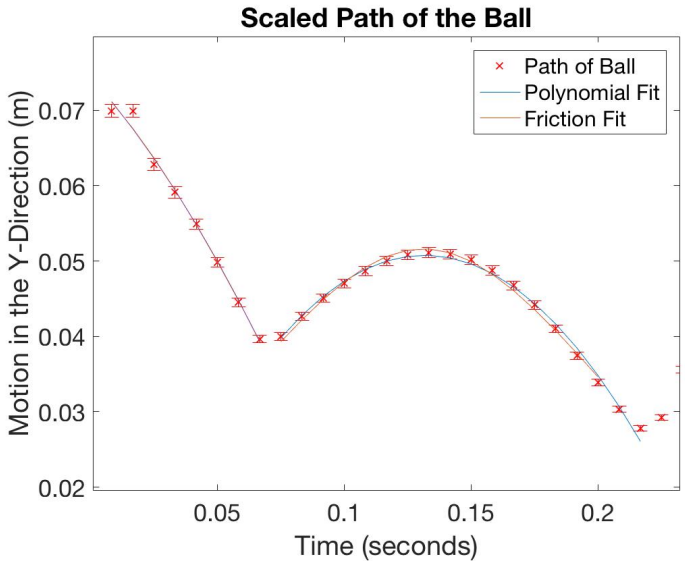

Figure 5: A sample output graph of the motion-tracking script for a ball drop on 7/20/16. The graph plots the path of the ball, a least-squares parabolic fit, and a least squares curve fit assuming friction in the air. The parabolic and friction fits are difficult to distinguish in this graph. There are also error bars on each data point from errors in measurement and video tracking. See section 3.1.2 for further explanation of the uncertainty analysis.

alyzed (Figure 5, Figure 8, Figure 7). In order to analyze the data points, simple models were fit to the data. In the first, a constant acceleration in the y-direction, g, was assumed (Equation 1). Consequently, a linear and parabolic equations could be fit to the velocity and position graphs respectively (Equation 2, Equation 3).

$$\frac{dv}{dt} = g \tag{1}$$

$$v(t) = k_1 x + k_2 \tag{2}$$

$$s(t) = k_1 x^2 + k_2 x + k_3 \tag{3}$$

Additionally, another set of fits was analyzed with the data assuming acceleration is not constant but rather slows due to friction in the air (Equation 4) [8]. The velocity and position solutions to this differential equation were then also fit to, and plotted with, the data (Equation 5, Equation 6) [3].

$$\frac{dv}{dt} = g - cv^2 \tag{4}$$

$$v(t) = \frac{\sqrt{g} \cdot tanh(\sqrt{g}\sqrt{c}(k+t))}{\sqrt{c}} \tag{5}$$

[3]Equations 5 and 6 were solved using *Mathematica*

$$s(t) = \frac{log(cos(\sqrt{g}\sqrt{c}(k_1 + t))}{c} + k_2 \tag{6}$$

The first bounce of each path was defined as when the velocity changed from negative to positive the first two times. The residuals from both the parabolic and friction fits were plotted for each trial. The residuals, and RMS values, for each fit were on the same order of magnitude and both exhibited clear patterns, indicating they might not be ideal fits (for example, Figure 6). However, for the scope of this experiment, both fits were considered sufficient and the friction fit was used for data analysis.

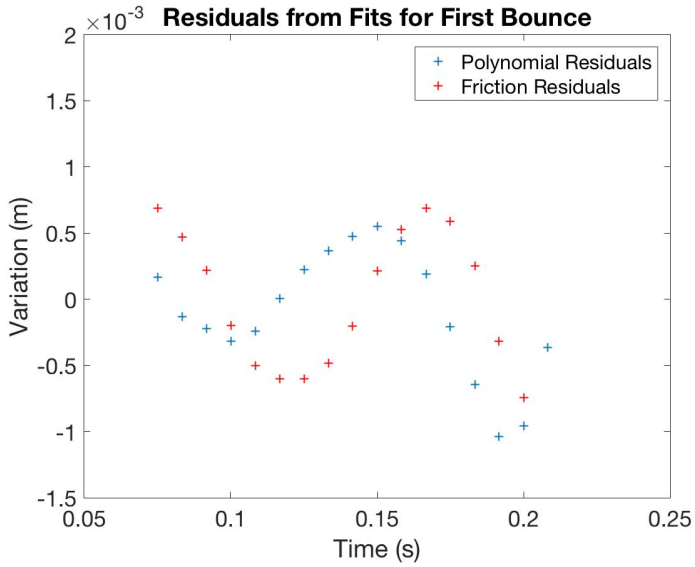

Figure 6: The residuals of the parabolic and friction-based fits for the first bounce of a sample trial taken on 7/20/16. Both fits are similar and exhibit a clear pattern, indicating that they are not ideal fits. The friction fit has a slightly lower RMS of 2.33×10^{-4}m versus a RMS of 5.39×10^{-4}m for the parabolic fit.

Energy released into the blade due to a ball's bounce was estimated with the kinetic energy lost by the ball during a bounce (Equation 8). The velocity right before and right after a bounce was estimated from the velocity fits with friction at the moment of the bounce, approximated as when velocity changes from negative to positive (Figure 7).

$$v = \frac{x_1 - x_2}{\Delta t} \tag{7}$$

$$\Delta KE = \frac{1}{2} m(v_\circ^2 - v_f^2) \tag{8}$$

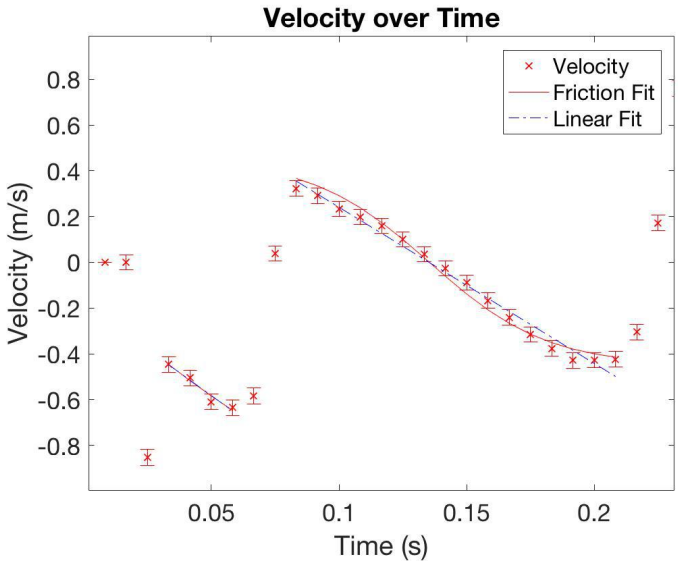

Figure 7: Velocity over time for a sample trial taken on 7/20/16. The initial drop and each subsequent bounce are plotted with both a linear fit and a fit which tries to account for friction. The change in velocity during each bounce allows for the calculation of kinetic energy lost in each bounce.

3.1.2 Uncertainty for Motion-Tracking Data Analysis

There were two main sources of uncertainty influencing the position data. The first was uncertainty in the scale of the video and therefore in the distances tracked. The ruler used for calibration had measurements up to millimeters, consequently the error associated was chosen to be $\pm$ 1 mm out of the total size of the video in both the x and y directions. Additionally, there was some uncertainty in the accuracy of the motion-tracking. In order to estimate this uncertainty, the script motion-tracked videos of balls fixed in one position. The average RMS of the position tracked versus the position of the ball for five trials was $\pm 2.59 \times 10^{-5} m$. Figure 5 shows one sample trial with error bars calculated in this way.

The goal of motion-tracking was to obtain estimates for kinetic energy, which is calculated using mass and velocity (Equation 8). Mass uncertainty was estimated from error in the scale, which measured the mass of each ball to be $8 \pm 0.33 \times 10^{-4} g$ (see section 3.2). Velocity was calculated with equation 7 where the change in time is 1/120 seconds and uncertainty was calculated from errors in the position data. There is also an uncertainty associated with the fit to the velocity data. The RMS of a linear fit to the velocity was used to calculate uncertainty from the fit.

Acceleration was not used for any calculations in the experiment, but was helpful as a sanity check. For example, for the trial shown in the plots, the average acceleration between bounces was $9.769 \pm 2.213 \frac{m}{s^2}$. This is measurement is compatible with the accepted acceleration due to gravity in Los Angeles - $9.796 \frac{m}{s^2}$ (Figure 8) .

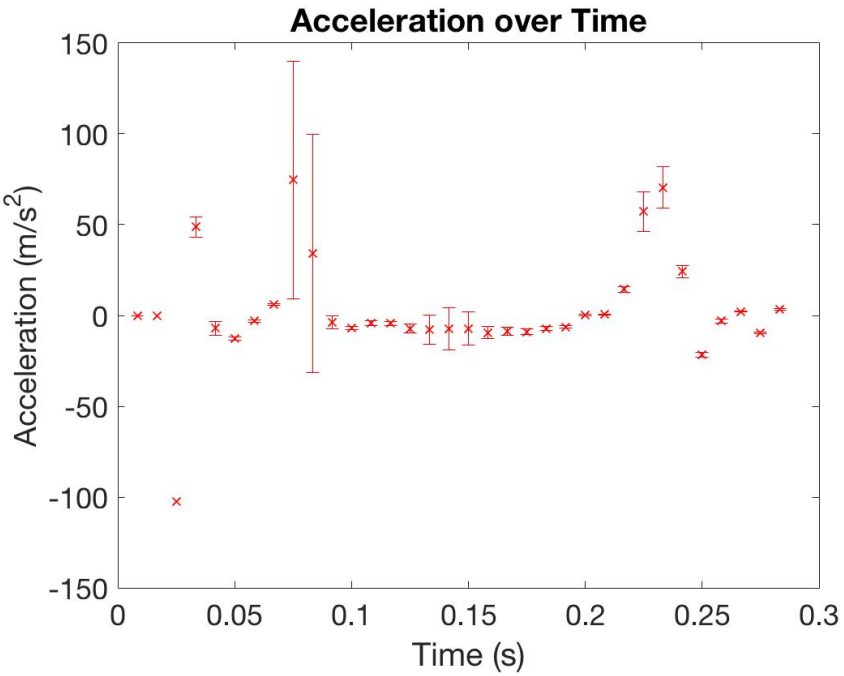

Figure 8: Acceleration over time for the sample video taken on 7/20/16. The graph exhibits a somewhat constant set of values between bounces. For this trial, the average acceleration between 0.1 and 0.2 seconds is $-9.769 \pm 2.213 \frac{m}{s^2}$ which is compatible with the acceleration due to gravity in Los Angeles: -9.796 $\frac{m}{s^2}$.

3.2 Calibration Experimental Design

Even though the initial experimental setup was effective for testing motion-tracking software, an improved setup was needed for the microphone calibration. An adjustable aluminum frame was built for dropping the ball from various heights and distances. A simple electromagnet - consisting of a battery, a switch, and a wire coiled around an iron screw - was placed at the end as a mechanism for dropping the magnetic steel balls (Figure 9). Additionally, a camera mount, a ruler for video calibration, and a black back drop for motion-tracking were added.

The experiment was originally run with steel balls with a diameter of $\frac{3}{64}$ inches and a mass of $7 \pm 1 \times 10^{-3} g$. The majority of initial trials run with these balls saturated the sensitive microphones. Consequently, smaller steel balls and balls of other materials were tested. Aluminum balls were ultimately chosen for the experiment over the other available materials due to their low density and high elasticity (Table 1). The aluminum balls had a diameter of $\frac{1}{32}$ inches and a mass of $8 \pm 0.33 \times 10^{-4} g$ (Figure 11).

Because the aluminum balls were not magnetic, the electromagnetic release mechanism was replaced. The final release mechanism consisted of half a coffee stirrer stuck through construction paper (Figure 10). When pulled through the construction paper, a ball in the straw would drop onto the blade.

Due to the small size of the balls, a black backdrop was set up for contrast and the videos were

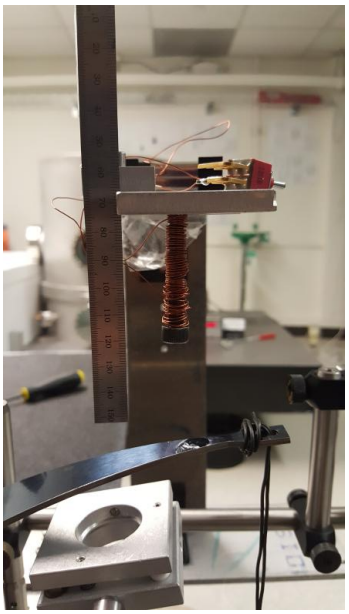

Figure 9: The electromagnet release mechanism, featuring a battery, switch, wire coiled around an iron screw, and a ball at the end, ready for release. A small ruler provides a scale in the plane of the ball to help with distance calibrations when analyzing videos.

Material	Min Diameter (in)	Volume (m^3)	Density ($\frac{kg}{m^3}$)	Mass (g)	Elasticity ($\frac{lb}{in^2} \times 10^6$)
Steel	0.0469	8.84E-10	8050	0.0071	29
Tungsten Carbide	0.0313	2.62E-10	19600	0.0051	79.77
Aluminum	0.0313	2.62E-10	2700	0.0007	10.2
Titanium	0.0313	2.62E-10	4500	0.0012	16.5
Lead	0.0800	4.39E-09	11340	0.0498	2
Copper	0.0625	2.09E-09	8930	0.0187	18
Brass	0.0625	2.09E-09	8500	0.0178	14.5
Rubber	0.1875	5.66E-08	1230	0.0696	3×10^{-4}
Ceramic (Si_3N_4)	0.0625	2.09E-09	3200	0.0067	30
Plastic (Resin)	0.0625	2.09E-09	1410	0.0030	0.3
Plastic (Nylon)	0.0625	2.09E-09	1150	0.0024	1.5

Table 1: A comparison of the size, mass and elasticity of possible materials to be used in the ball drop experiment. Aluminum was ultimately chosen for its low mass and relatively high elasticity.

recorded with the room lights off and the camera flash on.

Figure 10: Two views of the final release mechanism for the ball drop experiment. Left: a top view of the mechanism consisting of half of a small straw poked through construction paper. A ball resting in the straw drops onto the blade when the straw is pulled through the paper. Right: a front view of the release mechanism, as seen by the camera recording experimental trials. One of the aluminum balls used is resting in the straw and a ruler is in the plane of the ball for calibration.

Figure 11: A size comparison for the aluminum balls used in the final experiment. One ball, with a diameter of $\frac{1}{32}$ inches, sits on the shoulder of Abraham Lincoln.

3.3 Electronic Setup and Data Acquisition

The maximum sampling rate of the CYMAC computer - a real time digital system with ADC and DAC 16 bit channels - used for data acquisition in the microphone calibration experiment is 65 kHz. This limits the analysis of the microphone output to the 32.5 kHz Nyquist frequency. In order to counteract this limitation, a few techniques are used to down-convert the high-frequency signals in the microphone output.

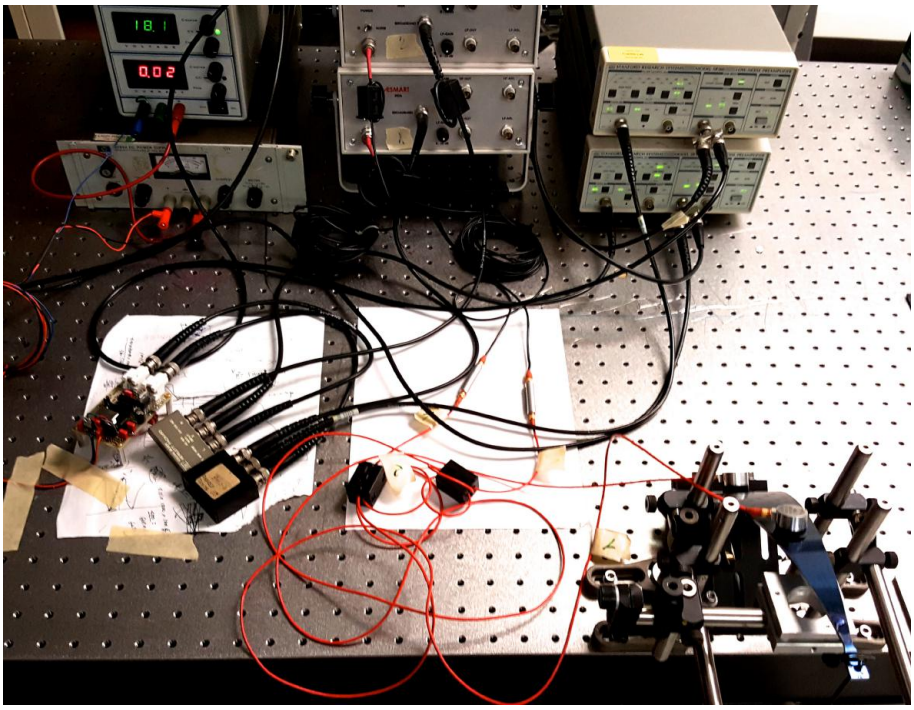

Figure 12: The electronic setup for the microphone calibration design. From left to right, in the upper row there are two DC power supplies, two conditioners, and two amplifiers. In the middle row there is the quartz Local Oscillator, two mixers, two ferrite beads, and two preamplifiers. In the bottom right corner is the loaded blade with two microphones mounted with wax.

A 5 MHz quartz Local Oscillator (L.O.) was used to output a square wave. The system was powered by about 18 V from a DC power supply and two counters (five count and two count) were used to obtain a 100 kHz wave from the 5 MHz L.O. Each microphone signal was passed through a ferrite bead to prevent the addition of extraneous signals and amplified 10 times by a preamplifier. The signal was then derived from two AESmart302A conditioners, which produced a broadband output.

Two HP1053A mixers were then used to multiply the broadband output from the microphones with the output of the L.O. This is done in order to down-convert higher frequency bands to the requisite baseband: 0-32.5 kHz. Because the 100kHz wave is a square wave with odd harmonics at different amplitudes, the mixer can down-convert multiple frequency bands to the baseband, which can then be sampled by the computer. The multiplied signal is sent to a Stanford Research Systems low-noise amplifier to be low-pass filtered (0-100 kHz) and amplified 100 times.

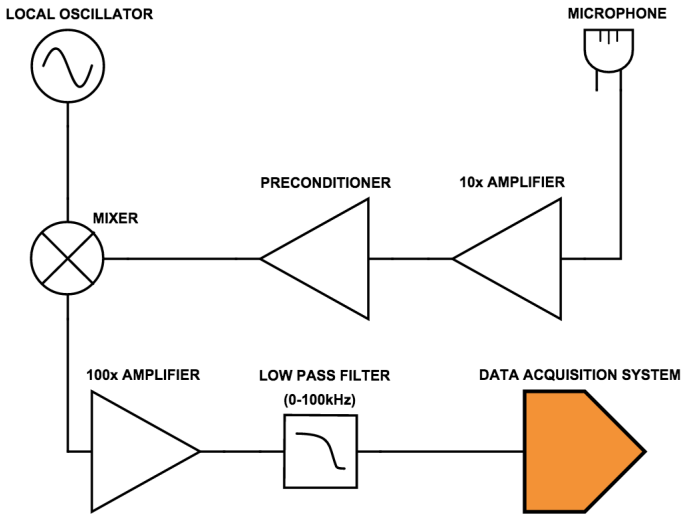

Figure 13: A diagram of the electronic experimental setup. The broadband output of the microphones are amplified, mixed with the signal from a local oscillator, further amplified, low-pass filtered, and finally sent to the data acquisition system.

The amplified signal was then sent from the amplifier to the CYMAC computer and an oscilloscope, in order to analyze and visually check the output data respectively.

The digital signal can be viewed with dataviewer. We are interested in the two demodulated outputs of the microphones, sampled at 65 kHz, and the RMS of each signal, saved at 4 kHz and low pass filtered. The signal is recorded in units of volts and the RMS in units of volts squared. The RMS is useful for analyzing electrical power and will ultimately be used to analyze the microphones' relationship with the energy input in the blade.

The data was plotted using ligoDV in MATLAB. Initial tests found that a ball dropping on the blade over-saturated the microphones. Consequently, for the sake of the calibration experiment, the amplifier was set to a lower amplification and a correction factor will be added to the calculations. Tests found a good balance between maximizing sensitivity and avoiding saturation occurred at 10 times amplification, versus the original 100 times amplification.

3.4 Microphone Data Analysis

The output of the microphones was analyzed in the LIGO DataViewer (ligoDV) application in MATLAB. For each microphone, the voltage over time and the mean squared of the voltage over time were collected and plotted (Figure 14, Figure 15). The area under the curve of each peak of the RMS data was extracted for correlation analysis.

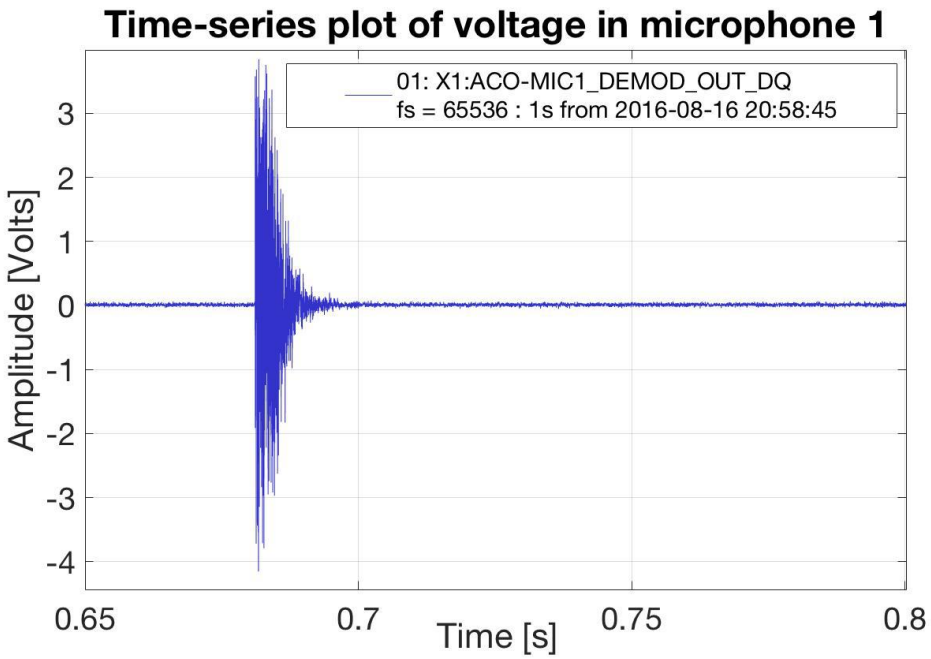

Figure 14: A plot of voltage measured in microphone one over time for a sample ball drop in which the ball bounced off the blade once.

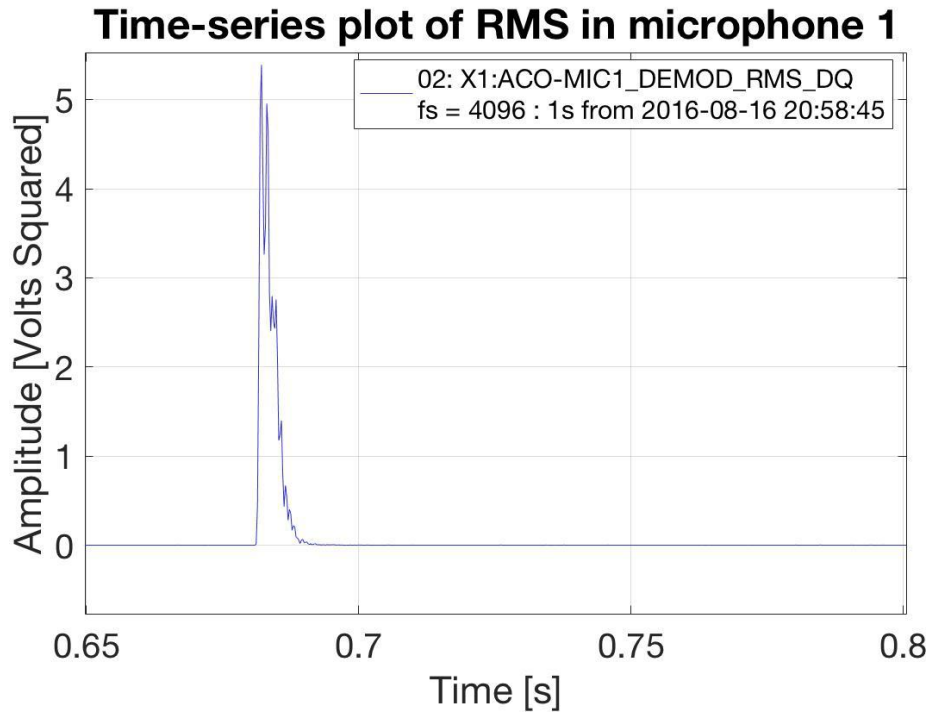

Figure 15: A plot of the mean squared of voltage measured in microphone one over time for a sample ball drop in which the ball bounced off the blade once.

4 Results and Conclusions

Forty-two trials of usable data were obtained over the course of two days. In twenty-one trials, balls were dropped from different heights onto the same location on the blade. In the other twenty-one trials, balls were dropped from the same height onto different locations onto the blade. I unfortunately did not save the data properly, and it was deleted. We expected a plot of energy output from the microphones versus energy input from the ball to be a direct linear relationship that levels off at a point of saturation. The calibration would have been useful for analyzing the results of future acoustic crackle experiments.

5 Future Work

The next steps are to actually get the microphones calibrated and to continue to run the acoustic emissions crackling experiment using the calibration data for the microphones. This will consist of rerunning the previous experiment to repeat the results and then improving the experiment. For example, the preexisting experiment only compares output for when the drive is on or off. This does not test for the possible relationship between the amplitude of the force or the derivative of force and the rate and size of crackling events. To analyze this, one could vary external force and analyze the resultant rate of variations in the data or changes in the noise floor. There are two possible ways to approach this. The first is to modify an algorithm used in the interferometer experiment [1] in order to find noise modulated coherently with the already running low frequency drive. The second is to look for small events in the microphone data and correlate them with external force. This approach could be done using LIGO software designed to detect transient gravitational waves. Ideally, both approaches will be tested.

These crackle experiments, along with others happening at LIGO, will serve to directly detect crackle events in maraging steel blades or at least set an upper limit on crackling noise. Through these experiments, we can hope to learn more about acoustic emissions in metals and potential sources of noise in the sensitive Advanced LIGO suspensions.

6 Acknowledgements

I had an amazing time doing research this summer at Caltech and I have a lot of people to thank for that.

I would like to thank my mentor, Dr. Gabriele Vajente, for all of his guidance, support, and insight. Additional thanks to the crackle and suspension lab groups for inviting me into their lab meetings and for all their help, and especially to Xiaoyue Ni for being a patient and helpful lab-mate.

The SURF pen was truly the most unique office I have ever worked in and I am grateful for their

friendship, support, and troubleshooting help this summer.

I would also like to thank LIGO and Caltech for hosting me this summer and NSF and the Victor Blanco Fellowship for funding me and making all of this possible.

7 Appendix

Below is the MATLAB code used for motion-tracking.

```
% Clear previous data and figures.
clc;
close all;
clear all;

% Open video player
videoReader = vision.VideoFileReader('ball28.mp4');

% Initialize MATLAB functions to video
videoPlayer = vision.VideoPlayer('Position',[100,100,500,400]);
foregroundDetector = vision.ForegroundDetector(...
    'NumTrainingFrames',10,'InitialVariance',0.05);
blobAnalyzer = vision.BlobAnalysis(...
    'AreaOutputPort',false,'MinimumBlobArea',70);

% Set empty matrices for later
location_y = [];
location_x = [];
detect = [];
acc = [];
vel = [];

% Set the pixel to m conversion
scale = 7.957/4800;


for i=1:200
kalmanFilter = []; isTrackInitialized = false;
   while ¬isDone(videoReader)
     % Set first frame as background
     colorImage  = step(videoReader);
     % Define foreground
     foregroundMask = step(foregroundDetector, rgb2gray(colorImage));
     % Define an object as variant from the first frame
     detectedLocation = step(blobAnalyzer,foregroundMask);
     isObjectDetected = size(detectedLocation, 1) > 0;

     if ¬isTrackInitialized
       % Configure a Kalman filter if an object is detected for first time
       % and initialize tracking.
       if isObjectDetected
```

```
            kalmanFilter = configureKalmanFilter( ...
                'ConstantAcceleration',detectedLocation(1,:), ...
                [1 1 1]*1e5, [25, 10, 10], 25);
          isTrackInitialized = true;
        end
        label = ''; circle = zeros(0,3);
      else
        % If an object is detected again, track location with Kalman
        % filtered predictions.
        if isObjectDetected
          predict(kalmanFilter);
          trackedLocation = correct(kalmanFilter, detectedLocation(1,:));
          % Place this location in a matrix
          location_x(i) = scale*(480 + (-1*trackedLocation(1)));
          location_y(i) = scale*(480 + (-1*trackedLocation(2)));
          i = i+1;
          label = 'Corrected';
        else
          % If there is no object detected after an object has been seen
          % Predict the location and store in the location matrix.
          trackedLocation = predict(kalmanFilter);
          location_x(i) = scale*(480 + (-1*trackedLocation(1)));
          location_y(i) = scale*(480 + (-1*trackedLocation(2)));
          i = i+1;
          label = 'Predicted';
        end
        circle = [trackedLocation, 5];
      end

      colorImage = insertObjectAnnotation(colorImage,'circle',...
          circle,label,'Color','red');
      step(videoPlayer,colorImage);

    end
end

% Take every other frame and set time
half_x=location_x(2:2:end);
half_y=location_y(2:2:end);
half = sqrt(half_x.^2 + half_y.^2);
%half = sqrt(location_x.^2 + location_y.^2);
len = length(half);
lenb= [1:len];
time = lenb/120;



% Calculate velocity (simple)
for i=2:len
    vel(i) = (half(i)-half(i-1))/(1/120);
end

% Calculate acceleration
for i=2:len
    acc(i) = (vel(i)-vel(i-1))/(1/120);
end
```

```
98
99  % Isolate first bounce
100
101 % Assume whole clip is first bounce
102 start = 1;
103 term = len;
104
105
106 % Set start and end frames for first bounce
107 for j=1:len-1
108     % If velocity changes signs, set start frame.
109    if vel(j) < 0 & vel(j+1)>0;
110       start = j+1;
111       for i=j+2:len-1
112           % If it changes signs again, set end frame.
113           if vel(i) < 0 & vel(i+1)>0;
114               term = i;
115               break
116           end
117       end
118       break
119    end
120 end
121
122
123 % Fit to velocity
124 fun = @(c,time)(sqrt(9.796)*tan(sqrt(c(1))*sqrt(9.796)*c(2) ...
125     + sqrt(c(1))*sqrt(9.796)*time))/(-sqrt(c(1)));
126 c = [0.00625,1];
127 vel_fit = lsqcurvefit(fun,c,time(start+1:term-1),vel(start+1:term-1));
128
129 % Fit to velocity initial drop
130 funf = @(f,time)(sqrt(9.796)*tan(sqrt(f(1))*sqrt(9.796)*f(2) ...
131     + sqrt(f(1))*sqrt(9.796)*time))/(-sqrt(f(1)));
132 f = [0.00625,1];
133 velf_fit = lsqcurvefit(funf,f,time(4:start-2),vel(4:start-2));
134
135 % Fit to velocity second bounce
136 funh = @(f,time)(sqrt(9.796)*tan(sqrt(f(1))*sqrt(9.796)*f(2) ...
137     + sqrt(f(1))*sqrt(9.796)*time))/(-sqrt(f(1)));
138 h = [0.00625,1];
139 velh_fit = lsqcurvefit(funh,h,time(term+2:end),vel(term+2:end));
140
141 % Calculate velocity with air resistance
142 for i=start:term
143     vel2(i) = (sqrt(9.796)*tanh(sqrt(vel_fit(1))*sqrt(9.796)*vel_fit(2)...
144         + sqrt(vel_fit(1))*sqrt(9.796)*time(i)))/(-sqrt(vel_fit(1)));
145 end
146
147
148 % Recalulate position
149 for i=start+1:term-2
150     pos(i) = trapz(time(start:i),vel2(start:i))+half(start);
151 end
152
153 % Fit to position for first bounce
```

```
d = [0.00625,-0.1179,1];
fund = @(d,time)(log(cos(sqrt(d(1))*sqrt(9.796)*(d(2)+time)))/d(1)+d(3));
pos_fit = lsqcurvefit(fund,d,time(start:term-2),half(start:term-2))

% Fit to position for initial fall
e = [-1,-0.1179,0];
fune = @(e,time)(log(cos(sqrt(e(1))*sqrt(9.796)*(e(2)+time)))/e(1)+e(3));
pose_fit = lsqcurvefit(fune,e,time(1:start-1),half(1:start-1))

%fun(vel_fit,time(start))
%funf(velf_fit,time(start))

% Plot velocity over time and fits
err2(1:len) = 0.1697;
figure;

% Initial drop
Ac0 = polyfit(time(4:start-2),vel(4:start-2),1);
yfit0 = Ac0(1)*time(4:start-2)+Ac0(2);
yfit0_all = Ac0(1)*time+Ac0(2);
% First bounce
Ac = polyfit(time(start+1:term-1),vel(start+1:term-1),1);
yfit = Ac(1)*time(start+1:term-1)+Ac(2);
yfit_all = Ac(1)*time+Ac(2);
res5 = vel(start+1:term-1) - yfit;
rms5 = sqrt(sum(res5.^2)/length(res5));
%Second Bounce
Ac2 = polyfit(time(term+2:end),vel(term+2:end),1);
yfit2 = Ac2(1)*time(term+2:end)+Ac2(2);
yfit2_all = Ac2(1)*time+Ac2(2);

% Calculate uncertainties

% Set initial uncertainty
scale_unc = 0.1/(scale*720);
% Uncertainty on position
half_unc = sqrt((half*scale_unc).^2 + (2.59*10.^-5).^2);
% Uncertainty on velocity
half_perc = half_unc./half;
for i=2:len
    vel_unc(i) = ...
        sqrt((vel(i)*half_perc(i)).^2+(vel(i-1)*half_perc(i-1)).^2 + ...
        rms5.^2);
end
% Uncertainty on acceleration
vel_perc = vel_unc./vel;
for i=2:len
    acc_unc(i) = sqrt((acc(i)*vel_perc(i)).^2+(acc(i-1)*vel_perc(i-1)).^2);
end


%Plot Velocity
errorbar(time, vel, vel_unc, 'rx')
hold on;
plot(time(start+1:term-1), fun(vel_fit,time(start+1:term-1)), ...
    'r',time(start+1:term-1),yfit,'b-.',time(4:start-2), ...
```

```
208     funf(velf_fit,time(4:start-2)),'r',time(term+2:end), ...
209     funh(velh_fit,time(term+2:end)),'r')
210 hold on;
211 plot( ...
212     time(term+2:end),yfit2,'b-.',time(4:start-2),yfit0,'b-.');
213 title('Velocity over Time');
214 legend('Velocity','Friction Fit','Linear Fit');
215 xlabel('Time (s)');
216 ylabel('Velocity (m/s)');
217 set(gca,'FontSize',18);
218
219 % Find velocity values right before and after first bounce
220 a = yfit0_all(start-1)
221 b = yfit_all(start-1)
222 % Find velocity values right before and after second bounce
223 n = yfit_all(term)
224 m = yfit2_all(term)
225
226 % Find a best fit parabola for the first bounce and find the maximum.
227 [p1, S, mu] = polyfit(time(start:term), half(start:term) , 2);
228 [y3,Δ] = polyval(p1, time(start:term),S,mu);
229 mean(Δ);
230 time_max = -.5*(p1(2)/p1(1));
231 y1_max = polyval(p1,time_max);
232
233 % Find a best fit parabola for the drop and find the maximum.
234 p4 = polyfit(time(1:start-1), half(1:start-1) , 2);
235 y4 = polyval(p4, time(1:start-1));
236 time_max1 = -.5*(p4(2)/p4(1));
237 y4_max = polyval(p1,time_max1);
238
239 % Plot motion
240 figure;
241 err(1:len) = 0.001;
242 errorbar(time,half, half_unc, 'rx')
243 hold on
244 plot(time(start:term),y3,time(start:term-2),fund(pos_fit, ...
245     time(start:term-2)),time(1:start-1),fune(pose_fit,time(1:start-1)), ...
246     time(1:start-1),y4);
247 title('Scaled Path of the Ball')
248 xlabel('Time (seconds)');
249 ylabel('Motion in the Y-Direction (m)');
250 set(gca,'FontSize',18);
251 legend('Path of Ball','Polynomial Fit', 'Friction Fit');
252
253 % Plot residuals from fit
254 res3 = half(start:term) - y3;
255 res4 = half(start:term-2) - fund(pos_fit,time(start:term-2));
256 figure, plot(time(start:term),res3,'+',time(start:term-2),res4,'r+')
257 title('Residuals from Fits for First Bounce');
258 xlabel('Time (s)');
259 ylabel('Variation (m)');
260 legend('Polynomial Residuals','Friction Residuals');
261 set(gca,'FontSize',18);
262
263 % Find RMS on residuals
```

```
rms3 = sqrt(sum(res3.^2)/length(res3));
rms4 = sqrt(sum(res4.^2)/length(res4));

% Percent error
perc3 = (res3.*100)./half(start:term);
perc4 = (res4.*100)./half(start:term-2);

% Plot acceleration over time
figure;
errorbar(time,acc, acc_unc, 'rx')
hold on
la(1:len) = -9.8;
plot(time,acc,'rx')
title('Acceleration over Time');
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');
set(gca,'FontSize',18);
mn  = mean(acc(start+2:term-4));
```

References

[1] Vajente, G., et al. ”An Instrument to Measure Non Linear Mechanical Noise in Metals in the Elastic Regime.” (2016): LIGO Laboratory, California Institute of Technology, Pasadena. Submitted to Review of Scientific Instruments.

[2] Weinstein, Alan. ”Intro to LIGO.” June 2016.

[3] Paoletti, Federico, Vajente, Gabriele. ”Acoustic Ultrasonic Measurements to Investigate Crackling Noise in Maraging Steel Blade Springs.” LIGO-T1500510-v1 (2015)

[4] James P. Sethna, Karin A. Dahmen, Christopher R. Myers, ”Crackling Noise.” Nature, 410, 242-250, 8 March 2001.

[5] Paoletti, Federico, Vajente, Gabriele, Ni, Xiaoyue. ”Acoustic Emissions in Maraging Steel Blades in the Elastic Regime.” (2015)

[6] ”SE9125-M AE Sensors.” Score Atlanta Inc, n.d. Web. 12 May 2016. https://score-atlanta.com/products/AE%20Sensors/SE9125-M

[7] Ni, Xiaoyue, et al. ”Crackling Noise in Gravitational Wave Detectors.” (2016): LIGO Laboratory, California Institute of Technology, Pasadena. LIGO RD. https://www.ligo.caltech.edu/LA/page/research-development

[8] Taylor, John R. ”Classical Mechanics.” (2005): University of Colorado. 58-60.