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

Technical Note	**LIGO-T1600150-v1**	June 30 2016

Progress Report 1: Acoustic Emissions in Metals

Danielle Frostig, Gabriele Vajente

Distribution of this document:

LIGO Scientific Collaboration

Draft

1 Motivation

1.1 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.

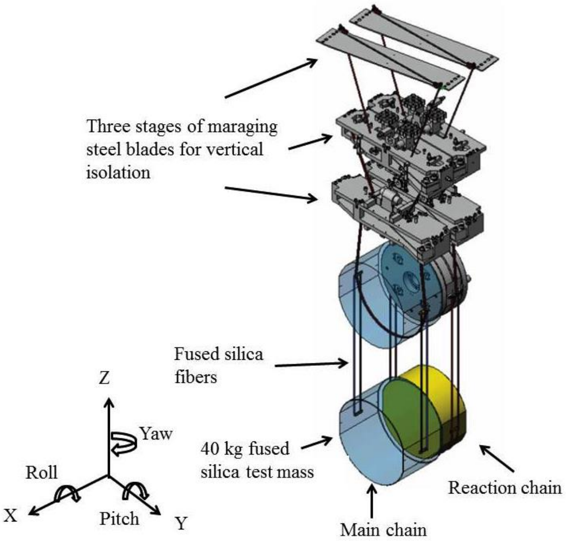

Figure 1: The Advanced LIGO suspension system, featuring a quadruple pendulum horizontal isolation system and three stages or maraging steel cantilever springs. The diagram is used courtesy of [7].

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.

1.2 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 and increased rate of events with increased driving force or rate of force [1].

2 Problem

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].

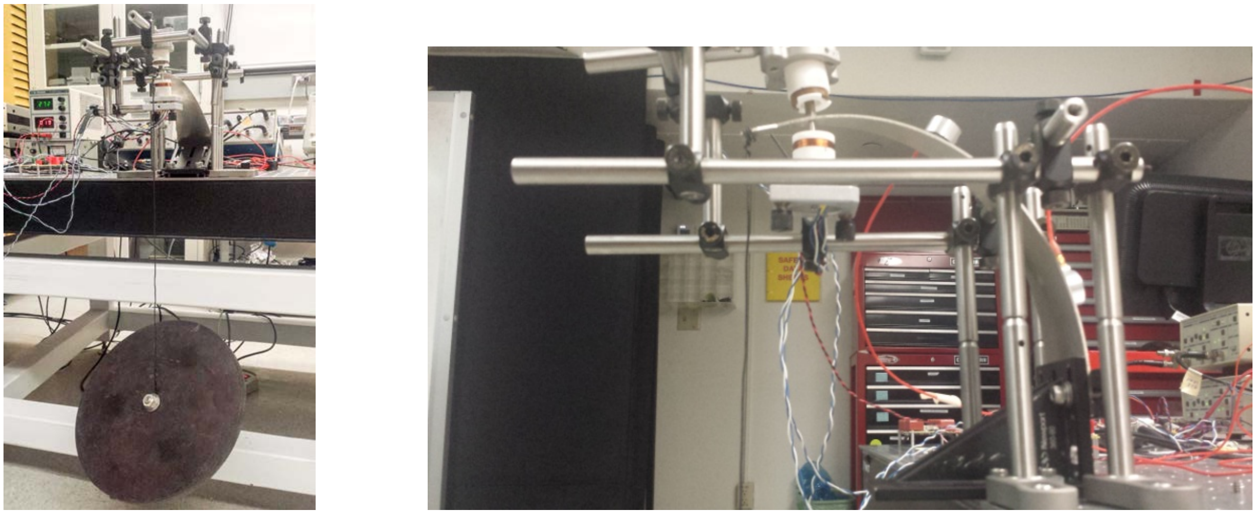

Figure 2: Maraging steel blade loaded with an 11 kg mass (right) and 16 kg mass (left).

The system was excited with a sinusoidal displacement of about 500 um at 0.4 Hertz 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 30kHz to a few hundred kHz was set in maraging and high carbon steel blades [3].

The overall goal of this project is to further improve this experimental setup and continue the search for crackling noise. Ideally, this will include the calibration of the microphones to energy output, improvement of data analysis, testing variable amplitudes and rates on the drive, and possibly even finding evidence of crackling noise.

3 Progress

3.1 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 it to compare their output with the known energy release of a simple system – in this case, dropping a steel ball onto the blade. By comparing the height of the ball initially with its height after one bounce, the energy released into the blade can be approximated and used to calibrate the microphone.

3.1.1 Motion-Tracking

Figure 3: Early experimental setups for testing motion-tracking. The earlier experiment (right) demonstrates the binder clip and coffee stirrer release mechanisms. The later experiment (left) has a monochromatic background and demonstrates motion-tracking. The colors have been inverted for ease of viewing.

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 1/4" in diameter (Figure 3a). Videos of the experiment were taken with a Samsung Galaxy 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 MATLAB script

---

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

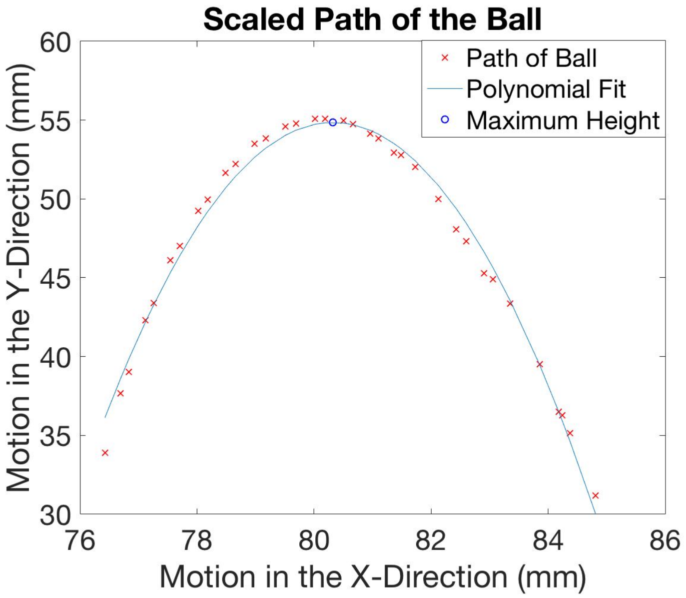

Figure 4: A sample output graph of the motion-tracking script. The graph plots the path of the ball, a least-squares regression parabola fit to the data, and the maximum height of the parabola. This graph is plotting the motion of the ball in figure 3b.

used is built on a prewritten motion-tracking program based on Kalman filtering.[2] So far, the script can track the path of the ball, plot the motion to the appropriate scale, apply a least squares fit parabola to the path, and find the maximum height of the bounce (See appendix).

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 3b). The MATLAB script ran with increased success with the new background.

3.1.2 Calibration Experimental Design

Even though, the test experimental setup was effective for testing motion-tracking software, an improved setup is needed for the microphone calibration. The ball drop mechanism and camera mount are currently in the process of being built. A simple electromagnet was used in favor of the binder clip for accuracy and ease of release.

---

[2] Script used courtesy of Vibhutesh Kumar Singh for Digital iVision Lab and can be found at http://www.divilabs.com/2013/11/motion-trackingdetection-in-MATLAB.html

4 Looking Forward

The most immediate next goal of the project is to finish setting up the microphone calibration experiment. Once the ball drop component is functioning, I can run the experiment many times from various heights, locations on the blade, and with balls of different weights. Then, I will figure out how to correlate the microphone output with the approximate energy released in the blade. Throughout this process it will be important to note if the data seems to make sense (e.g. does the distance between the ball and the microphone correlate to the output) and to obtain an estimate for uncertainty in the calibration.

Once the microphone is calibrated, I will rerun the previous experiment to repeat the results and then aim to improve 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, I 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. I will test both approaches and based on the success of each, decide to mainly pursue either approach, both, or another solution entirely. The ultimate goal of the project is to detect discrete crackling events but the project will still be successful even just by furthering the experiment via the above steps and by finding an upper limit on crackling noise.

There are seven weeks left of LIGO SURF. Below is a tentative schedule of the weeks that have transpired and the weeks to come.

Week	Task
6/13	Orientations, safety training, paperwork, and lectures.
6/20	Setting up motion tracking and initial ball drop experiment.
6/27	Finish setting up microphone calibration experiment.
7/4	Further analysis with LIGO software and algorithm while running improved experiments.
7/11	Running preexisting experiment and data collection.
7/18	Running experiment with variable drive forces and frequencies.
7/25	Data analysis with crackle experiment algorithm and continuous running of experiment.
8/1	Data analysis with LIGO software and continuous running of experiment.
8/8	Further analysis with LIGO software and algorithm while running improved experiments.
8/15	Solving any problems found; final experiments; collecting and synthesizing final data.

Table 1: A tentative schedule for the weeks to come.

5 Appendix

Below is the MATLAB code used for motion tracking.

```matlab
% I am using the base of this code courtesy of
% Vibhutesh Kumar Singh for digital iVision Lab
% I added comments to any of my own code added to
% the original.

clc;
close all;
clear all;

% Set empty matrices for storing motion data
xcent = [];
ycent = [];
xact = [];
yact = [];

% Set the pixel to mm conversion
scale = 1/4.8;

% Read video into MATLAB using VideoReader
video = VideoReader('ball91.mp4');

%'n' for calculating the number of frames in the video file
nframes = length(video);
nframes=video.NumberOfFrames;
for i=1:nframes
    mov(i).cdata=read(video,i)
end

% Calculate the background image by averaging the first 10 images
temp = zeros(size(mov(1).cdata));
[M,N] = size(temp(:,:,1));
for i = 1:10
    temp = double(mov(i).cdata) + temp;
end

% Initialization step for Kalman Filter
imbkg = temp/10;
centroidx = zeros(nframes,1);
centroidy = zeros(nframes,1);
predicted = zeros(nframes,4);
actual = zeros(nframes,4);

% % Initialize the Kalman filter parameters
% R - measurement noise,
% H - transform from measure to state,
% Q - system noise,
% P - the status covariance matrix,
% A - state transform matrix,
R=[[0.2845,0.0045]',[0.0045,0.0455]'];
```

```
H=[[1,0]',[0,1]',[0,0]',[0,0]'];
Q=0.01*eye(4);
P = 100*eye(4);
dt=1;
A=[[1,0,0,0]',[0,1,0,0]',[dt,0,1,0]',[0,dt,0,1]'];

% loop over all image frames in the video
kfinit = 0;
th = 38;
for i=1:nframes
    imshow(mov(i).cdata);
    hold on
    imcurrent = double(mov(i).cdata);

    % Calculate the difference image to extract pixels with
    % more than 40(threshold) change
    diffimg = zeros(M,N);
    diffimg = (abs(imcurrent(:,:,1)-imbkg(:,:,1))>th) ...
        | (abs(imcurrent(:,:,2)-imbkg(:,:,2))>th) ...
        | (abs(imcurrent(:,:,3)-imbkg(:,:,3))>th);
    % Label the image and mark
    labelimg = bwlabel(diffimg,4);
    markimg = regionprops(labelimg,['basic']);
    [MM,NN] = size(markimg);
    % Do bubble sort (large to small) on regions in case
    % there are more than 1.
    % The largest region is the object (1st one)
    for nn = 1:MM
        if markimg(nn).Area > markimg(1).Area
            tmp = markimg(1);
            markimg(1)= markimg(nn);
            markimg(nn)= tmp;
        end
    end

    % Get the upper-left corner, the measurement centroid and
    % bounding window size
    bb = markimg(1).BoundingBox;
    xcorner = bb(1);
    ycorner = bb(2);
    xwidth = bb(3);
    ywidth = bb(4);
    cc = markimg(1).Centroid;
    centroidx(i)= cc(1);
    centroidy(i)= cc(2);

    % Plot the rectangle of background subtraction algorithm -- blue
    hold on
    rectangle('Position',[xcorner ycorner xwidth ywidth],'EdgeColor','b');
    hold on
    plot(centroidx(i),centroidy(i), 'bx');

    % Kalman window size
    kalmanx = centroidx(i)- xcorner;
    kalmany = centroidy(i)- ycorner;

```

```
    if kfinit == 0
        % Initialize the predicted centroid and volocity
        predicted =[centroidx(i),centroidy(i),0,0]' ;
    else
        % Use the former state to predict the new centroid and velocity
        predicted = A*actual(i-1,:)';
    end

    % Use Kalman filtering equation
    kfinit = 1;
    Ppre = A*P*A' + Q;
    K = Ppre*H'/(H*Ppre*H'+R);
    actual(i,:) = (predicted + K*([centroidx(i),centroidy(i)]' - ...
       H*predicted))';
    P = (eye(4)-K*H)*Ppre;
    hold on
    rectangle('Position',[(actual(i,1)-kalmanx)...
        (actual(i,2)-kalmany) xwidth ...
            ywidth],'EdgeColor','r','LineWidth',1.5);

    % Plot the tracking rectangle after Kalman filtering -- red
    hold on
    plot(actual(i,1),actual(i,2), 'rx','LineWidth',1.5);

    % Fill matrices with object location
    xcent(i) = scale*actual(i,1);
    ycent(i) = scale*(480 + (-1*actual(i,2)));

    xact(i) = actual(i,1);
    yact(i) = actual(i,2);

    drawnow;
end

% Find a best fit parabola for the data and find the maximum.
p = polyfit(xcent, ycent , 2);
y2 = polyval(p, xcent);
x_max = -.5*(p(2)/p(1));
y_max = polyval(p,x_max)

% Plot final path of the ball
plot(xact,yact, 'bx');


% Plot motion and best fit parabola in a seperate window.
figure;
plot(xcent, ycent, 'rx',xcent, y2, x_max, y_max, 'bo')
title('Scaled Path of the Ball')
xlabel('Motion in the X-Direction (mm)')
ylabel('Motion in the Y-Direction (mm)')
legend('Path of Ball','Polynomial Regression', 'Maximum Height', ...
    'xcent','y2','ymax')
```

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