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

Technical Note **LIGO-T10XXXX-00-R** Nov. 1, 2010

Finite Element Analysis of Seismic Isolation Stacks at the 40m Interferometer

Gopal Nataraj
Mentors: Koji Arai and Rana Adhikari

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/

Abstract

The sensitivity required for gravitational wave detection demands effective isolation of astrophysical disturbances from surrounding seismic and mechanical disturbances. In an effort to reduce the effects of these unwanted forces through attenuation of ground motions, the LIGO facilities use passive isolation systems consisting of alternating layers of metal and viscoelastic dampers. Mechanical analysis of these seismic isolation stacks amounts to understanding the eigenmodes of oscillation as well the transfer functions relating stack-base drives to stack-top responses. We use Finite Element Analysis to perform such characterizations on the mode cleaner stacks at the 40-meter Prototype Interferometer at Caltech. Eventually, these efforts will be useful towards developing an adaptive active noise cancellation control system for Advanced LIGO.

Contents

List of Figures

List of Tables

1 Introduction

The sensitivity required for gravitational wave detection demands effective isolation of astrophysical disturbances from surrounding seismic and mechanical disturbances. While horizontal motions are of primary concern, attenuation of vertical, rotational, and tilting motions are also of importance due to possible cross-coupling of the eigenmodes [1].

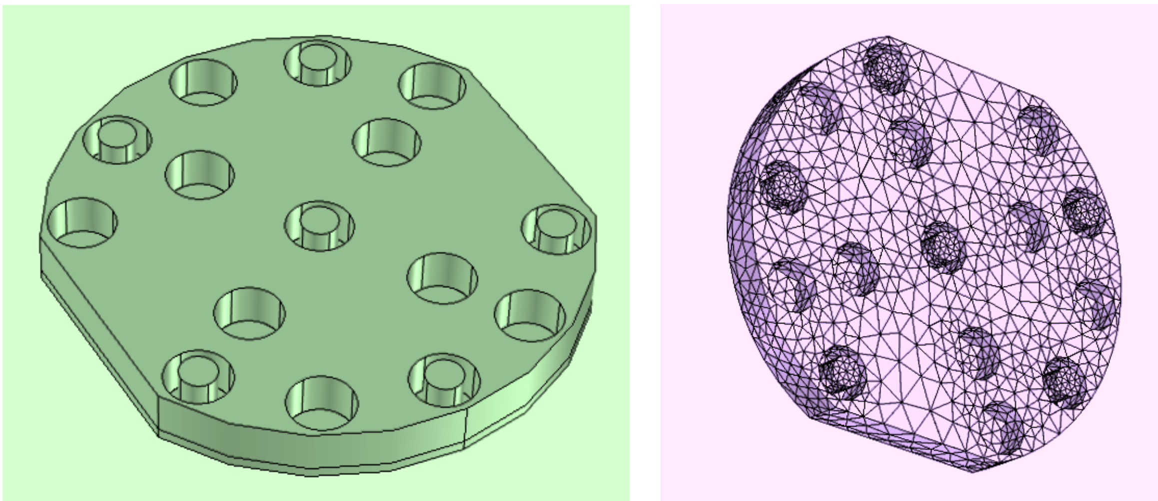
In an effort to reduce the effects of these unwanted forces, the LIGO facilities have placed all components on top of passive isolation systems which consist of alternating layers of metal and viscoelastic material. Each "stack" is designed in such a way as to distribute its own weight approximately evenly across each rubber spring. This is achieved by placing many springs below the base layer, fewer beneath the next layer, and so on until just a couple springs are placed below the top stack layer. With similar individual loads, the springs collectively resonate at certain frequencies in a constructive manner.

By choosing the dimensions and material properties of each stack carefully, we can exploit this constructive behavior and utilize the stack as a highly effective low-pass filter. Indeed, as long as horizontal resonance occurs well below 10 Hz and vertical resonance well below 30 Hz, seismic displacements at the base of the stack will be heavily attenuated before interacting with any objects secured to the top layer.

Mechanical analysis of the isolation system amounts to understanding the eigenmodes of oscillation as well as the transfer functions relating base drives to stack-top responses. Eigenfrequency analysis can show us which types of motion are most prone to excitation, given a seismic perturbation of a certain frequency. This can assist in prioritizing the actions of auxiliary active isolation systems that may work in conjunction with the stacks. Transfer function analysis will provide the frequency dependence of each motional degree of freedom available to the top layer when subjected to different types of oscillatory motion at the base. Such relations will describe the degree of filtering capability that each stack can provide.

Unfortunately, simple analytical calculation is complicated by the asymmetries in elastic damper placement in between the metallic layers: depending on initial conditions, multiple modes of motion may or may not couple in the previously mentioned fashion. For this reason, we are in the process of developing software simulations, using the FEA package COMSOL 4.0, to characterize the passive isolation systems currently installed within all of the chambers at the 40-meter Interferometer.

Figure 1: Top Stack Layer



(a) Before Meshing (b) After Meshing

Table 1: Material Properties of Viton Springs

Property	Value	Unit
Young's Modulus	6.6	MPa
Density	1800	kg/m^3
Spring Constant	3900	N/m

2 Methods

2.1 Model Design

2.1.1 Single-Layer Model

We first searched for documentation providing details about each of the mode cleaner chambers. Proper CAD drawings of MC1/MC3 chamber as well as its stack layers (Appendix A) were used to first model a single layer of the stack[1] (Figure 1).

In order to ensure that the materials were being properly modeled, we next used several sources to cross-check the material properties. The stainless steel plates were given a near-infinite stiffness matrix in order to simplify the computations by ignoring the negligible stresses and strains within the metal itself. Simple mechanical compression tests on the Viton (elastic) springs confirmed many of its material properties (Table 1).

[1]For more detailed documentation, please examine the IOC/OOC binders in the bookshelves of the LIGO 40m control room, from which many of these drawings were extracted

Figure 2: The Completed Stack

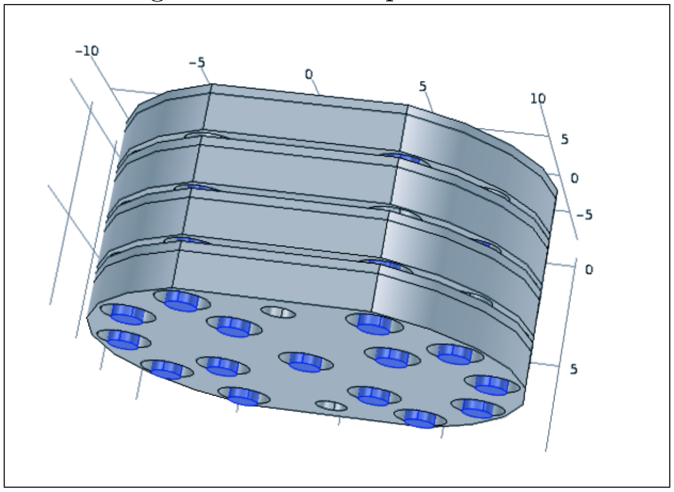


Table 2: First Four Eigenmodes of a Single Stack Layer

Harmonic #	Description	Eigenfrequency (Hz)
1^{st}	Y-Translational	7.493674
2^{nd}	X-Translational	7.546216
3^{rd}	Z-Rotational	8.633928
4^{th}	Z-Translational	18.255518

2.1.2 Multi-Layer Model

Once the top layer was completed, building the subsequent lower layers of the isolation stack was a straightforward task. After the completion of each layer, we updated the material properties and added extra dampers to the system, as laid out in the drawings (Figure 2).

2.2 Eigenfrequency Analysis

2.2.1 Single-Layer Model

We temporarily reverted to the single-layer stack for the initial eigenfrequency tests in order to easily perform order-of-magnitude checks on COMSOL's calculations. The software produced a first harmonic of about 7.5 Hz. Conveniently, the dampers were arranged in the past to produce an approximately constant 43 kg/spring load. A quick paper-and-pencil calculation indicated that the resonant frequency should be around 9.5 Hz, a reasonable first approximation. As seen in Appendix B.1, the single-layer case also allowed easier qualitative analysis of the first few eigenmodes. By recording the maximum displacement from equilibrium (Table 2), it was easy to decipher that the first, second, and fourth modes were translational, while the third was rotational.

2.2.2 Multi-Layer Model

Extension to the MC1/MC3 stack introduced some interesting principles (Appendix B.2). For instance, the first few harmonics dropped from around 7-8 Hz, down to 3-4 Hz, in agreement with theoretical stipulations claiming that the low-pass filtering capabilities of seismic stacks improve as more layers are added [2]. Furthermore, the multilayered nature introduced tilting eigenmodes with resonant frequencies of around 8 Hz.

Only after these tilting modes did the z-translational eigenmode reappear, this time as the sixth harmonic. As predicted, the horizontal-filtering capabilities of our stack were shown to be superior to the vertical, rotational, and tilting modes. However, since all of these eigenmodes are well within the 10-Hz threshold (Table 3), our model predicts that attenuation of seismic noise from all directions and rotations will be reasonable.

Table 3: First Six Eigenmodes of the MC1/MC3 Stack

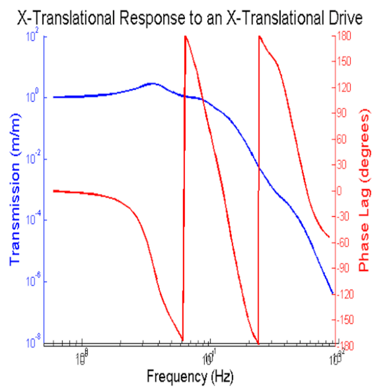
Harmonic #	Description	Eigenfrequency (Hz)
1^{st}	Y-Translational	3.343873
2^{nd}	X-Translational	3.391904
3^{rd}	Z-Rotational	3.387656
4^{th}	YX-Tilt	8.021973
5^{th}	XY-Tilt	8.201671
6^{th}	Z-Translational	8.551589

2.3 Transmissibility Analysis

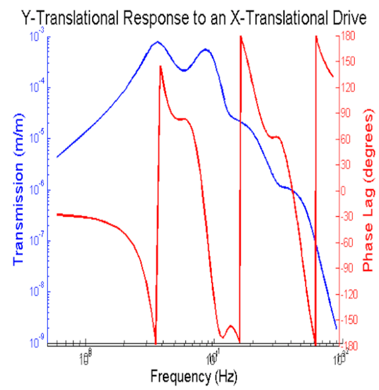
In order to complete the mechanical characterization of the MC1/MC3 seismic isolation stack, transfer functions relating the response of the center of the top layer to different driving forces at the base were measured using COMSOL's Frequency-Domain Analysis toolkit. An oscillatory drive with fixed amplitude was initialized in different directions on the base of the stack. As the frequency of oscillation varied from 0.3 Hz to 30 Hz, the displacement and phase lag (with respect to the base) of the top-center of the stack was recorded. The resulting nine transfer functions form a "transmissibility tensor" of sorts, which can be used to predict the translational response of objects placed on top, subject to any translational drive at the base of the stack (Figure 3).

Some further explanation of Figure 3 may provide a bit more intuition towards interpreting these results. A double y-axis scheme allows both transmission (m/m) and phase lag (degrees) to be plotted simultaneously against frequency. Phase angles are constrained between 0 and 2π. The diagonal plots take on the familiar form of a simple harmonic oscillator: a dominant peak in transmissibility locates the first eigenfrequency in the same direction as the drive. Our plots predict the first harmonic to be around 3 Hz for the horizontal directions and around 8 Hz for the vertical directions, in good agreement with the eigenfrequency measurements. Furthermore, a Q-factor of about 3 can be deciphered for all three diagonal

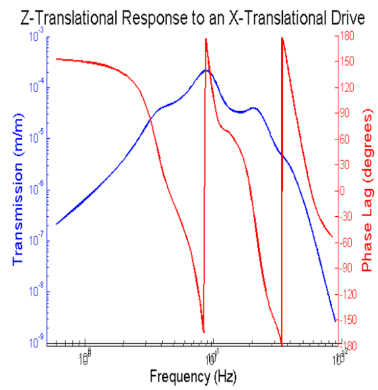
Figure 3: Translational Transfer Function of the MC1/MC3 Stack



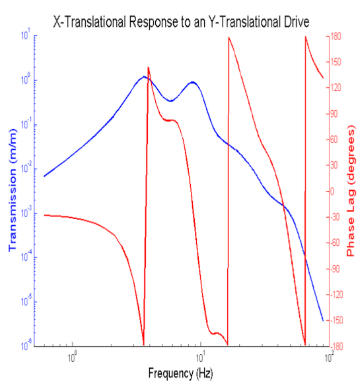
(a) X-Translational Response from X-Displacement Drive



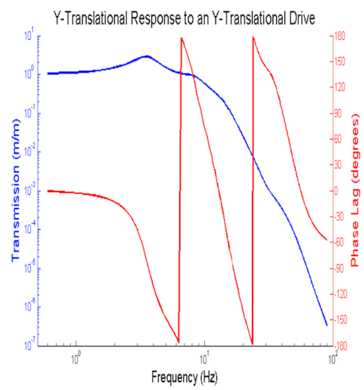
(b) Y-Translational Response from X-Displacement Drive



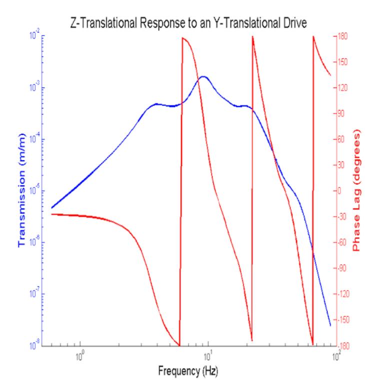
(c) Z-Translational Response from X-Displacement Drive



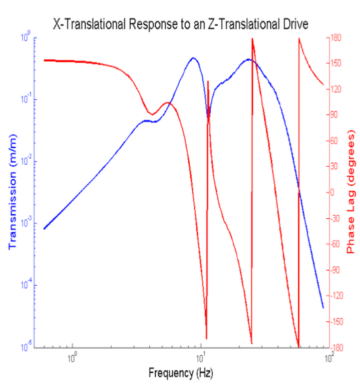
(d) X-Translational Response from Y-Displacement Drive



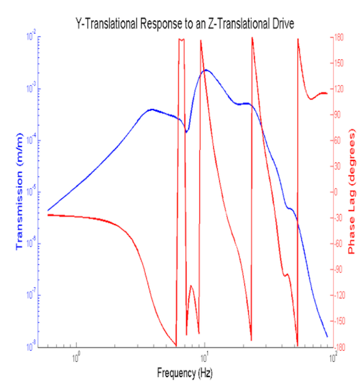
(e) Y-Translational Response from Y-Displacement Drive



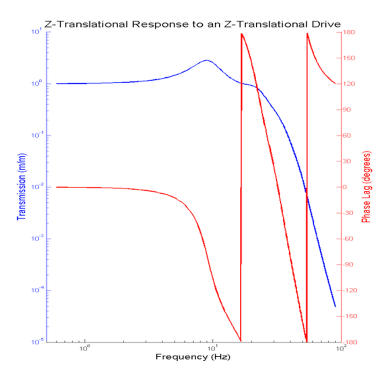
(f) Z-Translational Response from Y-Displacement Drive



(g) X-Translational Response from Z-Displacement Drive



(h) Y-Translational Response from Z-Displacement Drive



(i) Z-Translational Response from Z-Displacement Drive graphs, from the heights of the dominant transmission peaks. Higher-frequency peaks are less exaggerated but nevertheless identify the higher modes of oscillation which result from utilizing a multi-stage attenuator.

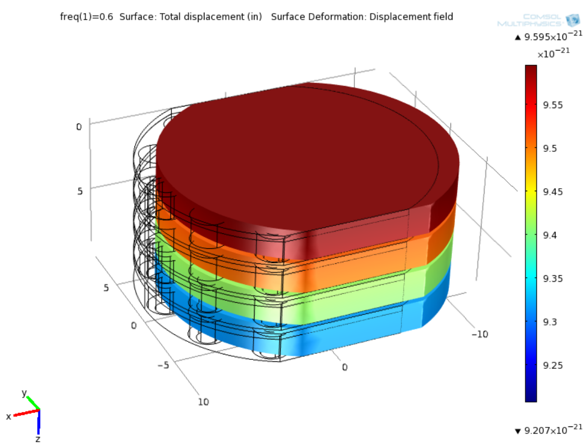
While the magnitudes of transmission of the off-diagonal plots are orders of magnitude smaller (Q $\sim 10^{-3}$) than their diagonal counterparts, they produce much more interesting behavior. Numerous peaks in transmissibility span the full range of frequencies tested. Often, several dominant frequencies complicate the expected linear log-log drop-off for high frequencies. As previously mentioned, these motions arise from the inevitable cross-coupling apparent in any type of asymmetric multiple-stage damper.

As a more visual demonstration of the transfer function tensor, Figure 4 depicts the response of sinusoidal drives with varying frequency. At the lowest frequencies, motion is translated nearly in phase with the drive, and the unit essentially moves as one rigid object. As the frequency is increased, phase lags become apparent, increasing in magnitude for higher and higher stack layers. For the highest frequencies, the response is physically unable to "keep up" with the drive, and so the stack acts as a mechanical low pass filter, heavily attenuating the ground motions with better efficiency as frequency increases.

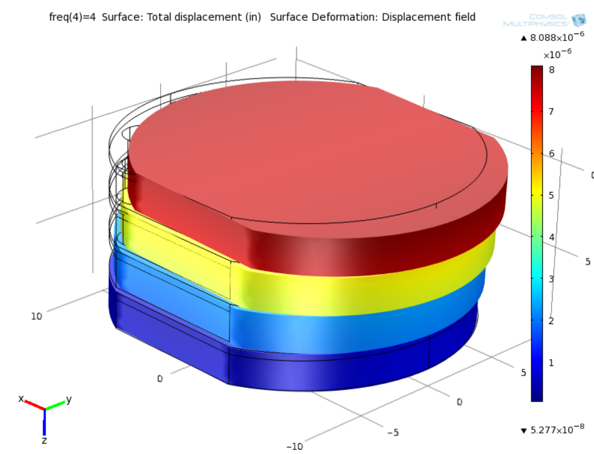
It should be noted that, while original models were able to incorporate frequency-dependent amplitude modulation and phase lag through the Young's Modulus, Q-factor control required utilization of COMSOL's material damping option. For a first-order approximation, the metal's damping factor was maintained at zero, while the rubber's was varied between 0.4 and 1.0, until a suitable Q-factor was achieved. The transmission plots given utilized a damping ratio of 0.5, though variation of the resultant Q-factor amounts to a simple modification.

While Section 3.2.3 will provide much greater detail regarding the issue, it is important to immediately recognize that this model does not incorporate the effects of gravity. Nevertheless, some physical reasoning will allow us to predict how gravity will perturb our current model. To first order, our isolation stack is essentially an inverted pendulum with a fixed base; upon such an object, gravity acts as an anti-restoration force, pulling the top away from equilibrium as it oscillates. This will increase the period and thus decrease the frequency of oscillation for any given drive. If we assume that this frequency-shift will leave the amplitudes and phases themselves unaffected, inclusion of gravity will shift our Bode plots rightward. In other words, a slightly higher input frequency will be required to observe the same response (at the original, lower frequency). Comparing the transmission data's first resonant peak (2.8 Hz) with the first eigenmode predicted through Eigenfrequency Analysis (3.344 Hz), such a right-shift correction would in fact result in the two models agreeing more closely. In the hopes of consistency between our models, we therefore predict that including gravity will produce roughly a 0.5 Hz rightward shift in all measured Bode Diagrams.

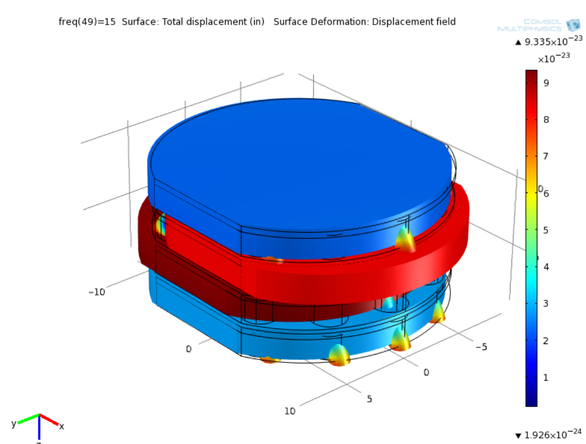
Figure 4: Displacement Responses of the MC1/MC3 Stack as Frequency is Varied



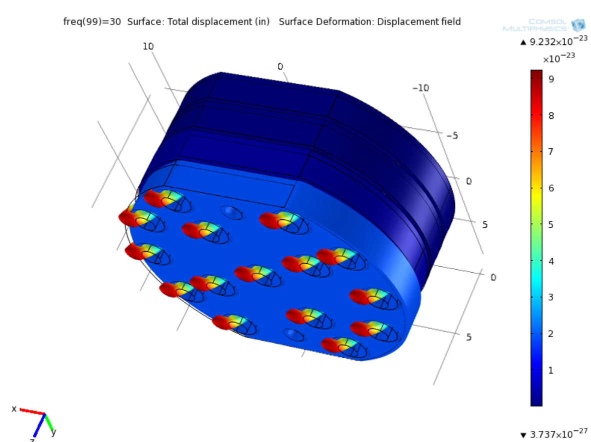
(a) 0.6 Hz X-Translational Drive



(b) 4 Hz X-Translational Drive



(c) 15 Hz X-Translational Drive



(d) 30 Hz X-Translational Drive

3 Discussion

3.1 Results and Accomplishments

Major advancements in understanding the dynamics of multi-stage stacks were made through the COMSOL Multi-Physics finite element analysis approach. Numerical eigenvalues have been documented after being coupled with visual descriptions of the various eigenmodes of oscillation. Our eigenvalues agree, to reasonable accuracy, with experimental measurements as well as first-order analytical calculations [2].

We then used a frequency domain approach to measure both the on-axis and shear motions of the model stack, given different driving forces. This allowed us to construct a matrix of transfer functions, which now provide a "basis" that can describe responses from translational drives in any coordinate system. The first resonant frequencies of the diagonal transmittance transfer functions also agreed to within a couple Hz with the results of the eigenfrequency analysis, which provided further confidence in the model. Combined, these two tests have provided the data necessary to comprehensively characterize the translational mechanics of the MC1-MC3 mode-cleaner chamber's multiple-layer seismic isolation stack.

Future modification of the stack model parameters may allow one to better fit the measured data. For instance, while the off-diagonal Q-factors were reasonable, the diagonal Q-factors slightly overestimated the expected values. For this reason, a slight reduction in the diagonal terms of the absorption coefficient matrix of our rubber dampers might improve results.

To facilitate the minor modification trial-and-error process, MATLAB scripts have been written to read in various COMSOL test parameters from an external source and output transfer function matrix elements similar to those shown in Figure 3 (Appendix D). By incorporating the intermediary mathematics directly into the code, our scripts will make repeating our measurements a quick and easy task.

3.2 Future Work

While many shortcomings of our prototypical model (e.g. run time reduction, mesh alterations, phase measurements, etc.) were resolved for the data displayed in this report, there are several extensions of particular interest which would allow for a more thorough mechanical analysis. The first subsection details a previously-encountered difficulty and the according solution, while the latter subsections provide motivation for current problems, any solutions proposed and their successes/failures, and future advancements.

3.2.1 Computational Power

As the complexity of the stacks increased and our tests became more extensive, run times for even the simplest simulations began extending from multiple minutes to multiple hours. While upgrading to the latest computers did assist appreciably, the issue required other techniques for reducing run times to reasonable levels.

Customization of the meshing was the most effective solution. Specifically, the Maximum Element Size and Maximum Element Growth Rate were greatly increased, while keeping other parameters normal. This not only allowed for sparse meshing of the large metallic plates, whose deformations were negligible and not of interest, but also maintained fine meshing of the action-heavy dampers. Because much of the stack volume comprised of the metal, this modification greatly reduced the number of mesh points, thereby vastly decreasing simulation run times without sacrificing a significant amount of precision.

3.2.2 Rotation and Tilt

Up until now, our transfer function tensor has only considered translational drives and responses. However, consideration of rotation about the z-axis as well as tilt about the x- and y-axes for both the input drive and output response is necessary in order to completely characterize the MC1/MC3 stack transfer function.

Input Drives Driving torques were successfully simulated by applying multiple forces of equal amplitude but appropriate relative phase along the edges of the base stack. To create tilt, opposing bottom edges were given vertical loads with a 180° phase shift relative to one another. To create rotation, opposing bottom edges were given horizontal loads with a similar phase difference. This technique has been shown to produce torsional drives as expected on simple objects; extension to the MC1/MC3 stack will soon follow.

Output Responses COMSOL provides users an option to record the normal vector of any point on a surface of the object, both before and after a Solid Mechanics test. In theory, this should provide sufficient information to transform to the rotation and tilt responses desired. However, when we attempted to utilize this feature on a point placed at the center of the top stack layer, the outputted transfer functions were nonsensical.

This problem indicates that the normal vector measurements are not being taken at the point of maximum displacement. Future efforts will be given towards either discovering methods of changing the phase of measurement, or implementing a new method to measure the rotational and tilting output responses altogether.

3.2.3 Gravity

Because of the surprising level of difficulty it poses, gravity has not yet been incorporated into our transfer function analyses. However, because of the possibly significant modifications that it could place on the eigenfrequencies and transfer functions, we are very interested in including its effects in our models.

Understanding why gravity poses such problems requires some background on how COMSOL conducts Frequency Domain Analysis. The user first defines all point, edge, and body loads as DC forces. Upon initializing FDA, COMSOL automatically tacks on an oscillatory term onto all loads and only then solves the equations of motion using FEA. It is therefore impossible to define constant loads while using FDA.

One solution involving direct manipulation of the differential equations has been proposed, but has yet to be tested. For small-angle deviations, the effect of gravity on a stack bounded at the base is similar to that of an inverted pendulum: there will be an anti-restoring force proportional to the x- and y-displacement from equilibrium. By replacing the gravitational load with such anti-restoring foreign terms, it may be possible to "trick" the software into not realizing that these factors are actually forces. Consequently, COMSOL may "forget" to tack on the time dependence onto the gravitational force that we manually define and subsequently solve the correct equations of motion.

A more promising solution involves switching from working in the frequency domain into the time domain. For this method, an impulsive drive must be analytically defined as time-varying force applied on the base of the stack. Since all forces are now defined as functions of time rather than frequency, a constant gravitational field will pose no problems. The time-domain response is then Fast Fourier-Transformed and divided by the Transform of the prescribed Gaussian approximation to produce a different transfer function.

3.2.4 Analytical Modeling

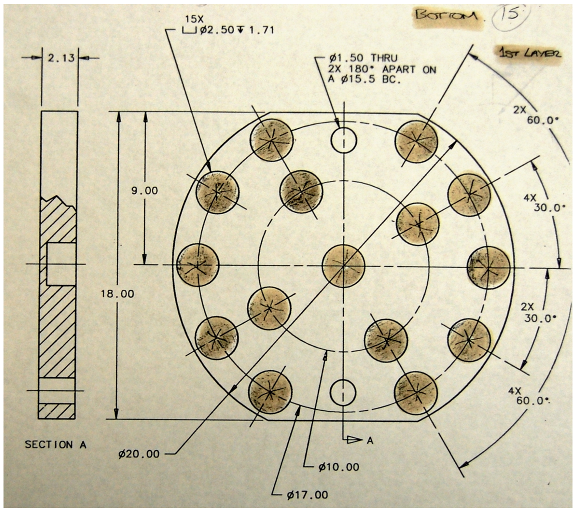
Once all of the prior concerns have been resolved, it will behoove us to fit analytical models which relate transmittance to driving frequency for each of the elements of the 6x6 transfer function tensor. From each elements' equation fittings, we can then ultimately construct a matrix equation which relates the 6-dimensional driving vector to the 6-dimensional response vector. This equation will be used in conjunction with seismometers and feedback loops to ultimately incorporate an active isolation system into Advanced LIGO.

4 Acknowledgements

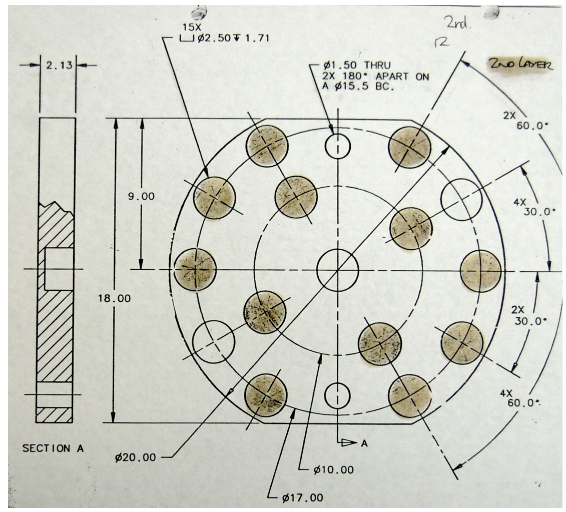
There are many people without whom this project would not have been possible, but for the sake of the reader, I will keep my list brief. First and foremost, I would like thank Dr. Koji Arai for his enthusiastic assistance, extensive discussions, and patient teaching. I also must thank my other mentor, Dr. Rana Adhikari for our insightful discussions and his honest criticism. I would also like to give thanks to the LIGO 40m team and the other LIGO SURF students for surrounding me with their brilliant creativity, thereby motivating me to stretch my potential. My appreciation to Caltech for providing me this once-in-a-lifetime opportunity should go without mention, but I will nevertheless reinstate my utmost gratitude. Finally, special thanks is due to Dr. Jan Harms, who took time out of his schedule on numerous occasions to stop into the 40m and sit down with me and answer my COMSOL questions. Without his experienced words of wisdom, I would probably still be debugging COMSOL boundary values conditions and MATLAB code instead of finishing this paper.

Appendices

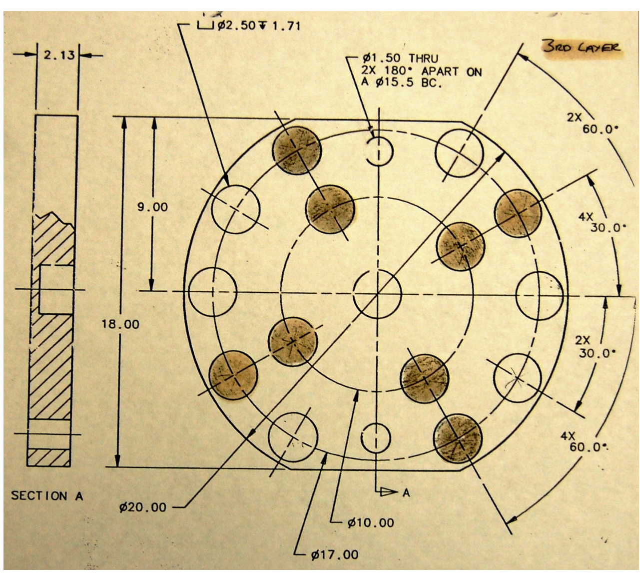
A Drawings of the MC1/MC3 Stacks



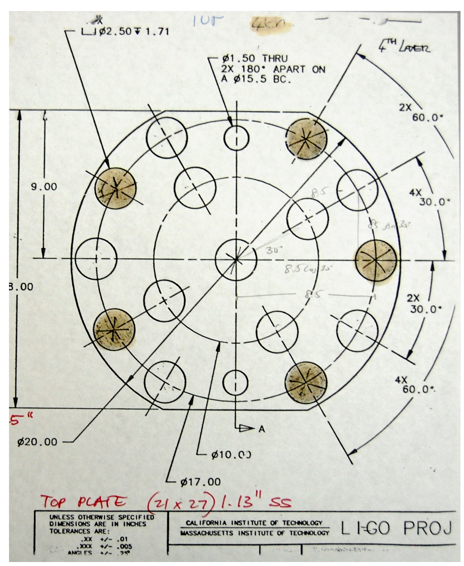
(e) Base Layer



(f) Second Layer



(g) Third Layer

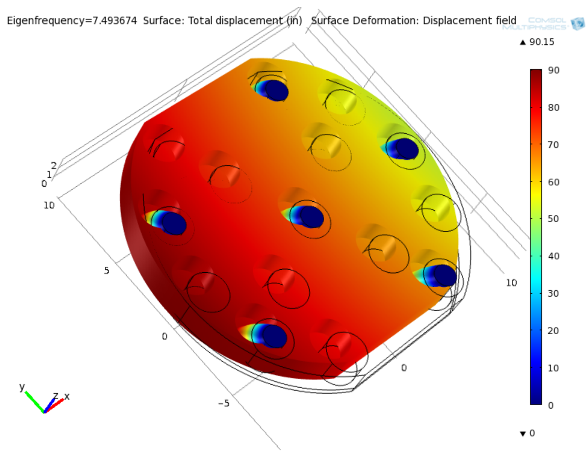


(h) Top Layer

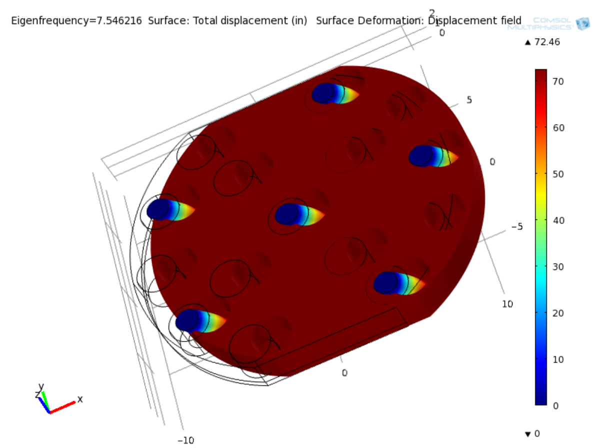
B Eigenfrequency Plots

B.1 Eigenfrequency Plots of a Single Stack Layer

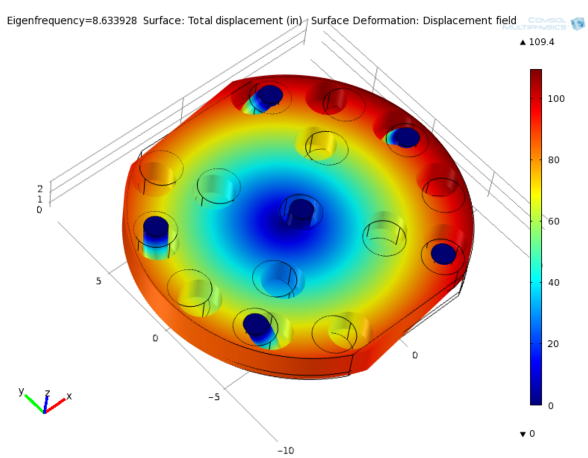
Figure 5: Eigenfrequency Analysis of a Single Stack Layer



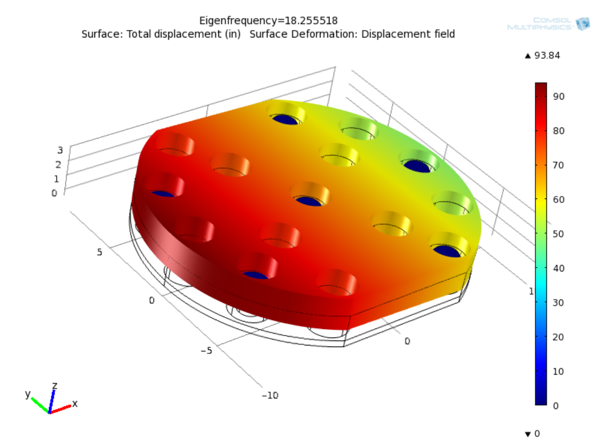
(a) First Eigenmode: Y-Translational



(b) Second Eigenmode: X-Translational



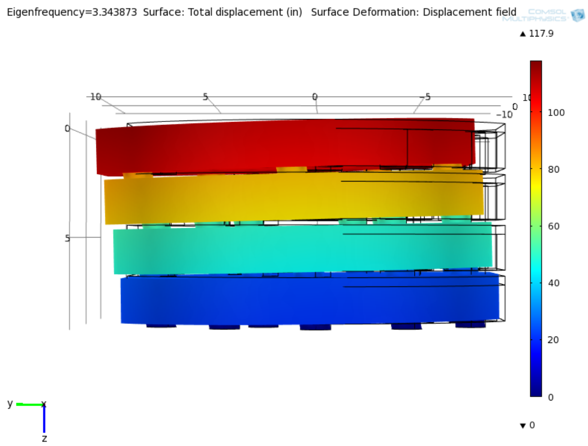
(c) Third Eigenmode: Z-Rotational



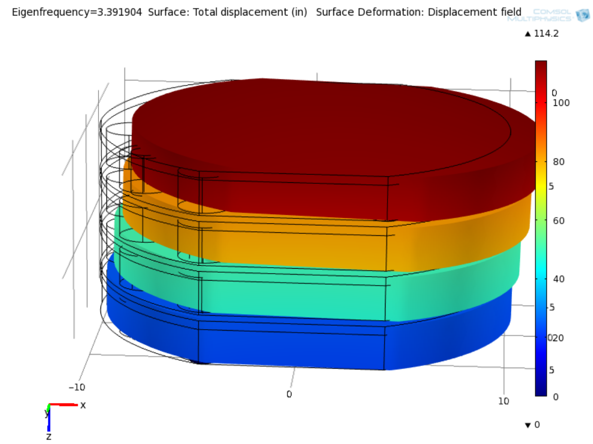
(d) Fourth Eigenmode: Z-Translational

B.2 Eigenfrequency Plots of the MC1/MC3 Stack

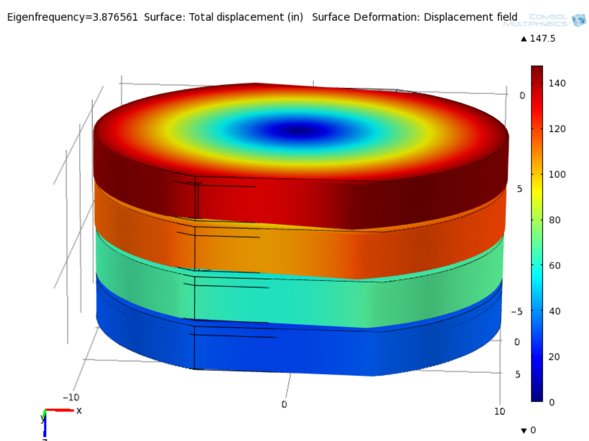
Figure 6: Eigenfrequency Analysis of a MC1/MC3 Stack



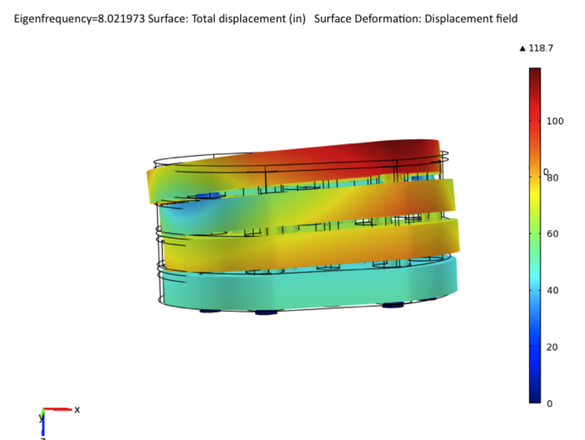
(a) First Eigenmode: Y-Translational



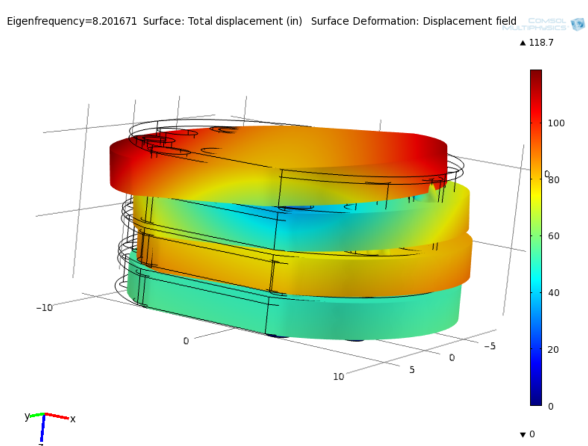
(b) Second Eigenmode: X-Translational



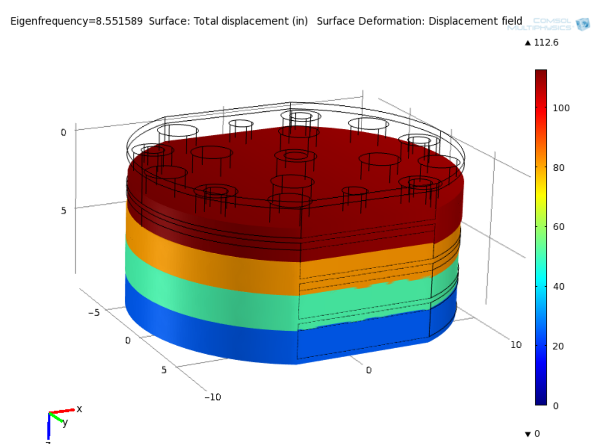
(c) Third Eigenmode: Z-Rotational



(d) Fourth Eigenmode: YX-Tilt



(e) Fifth Eigenmode: XY-Tilt

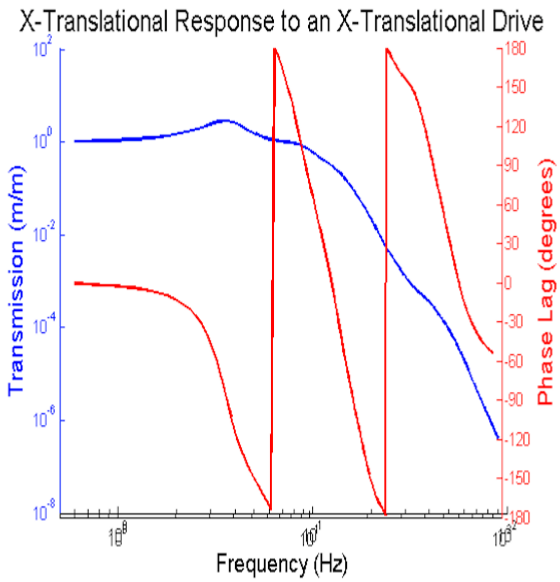


(f) Sixth Eigenmode: Z-Translational

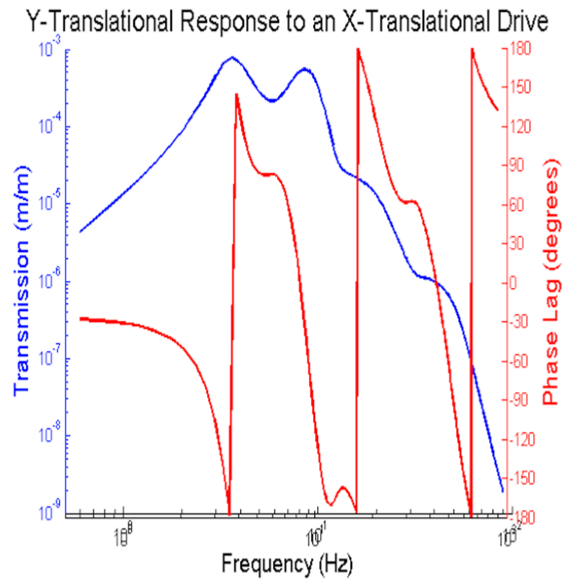
C Enlarged Versions of the Stack Transfer Functions

C.1 X-Displacements

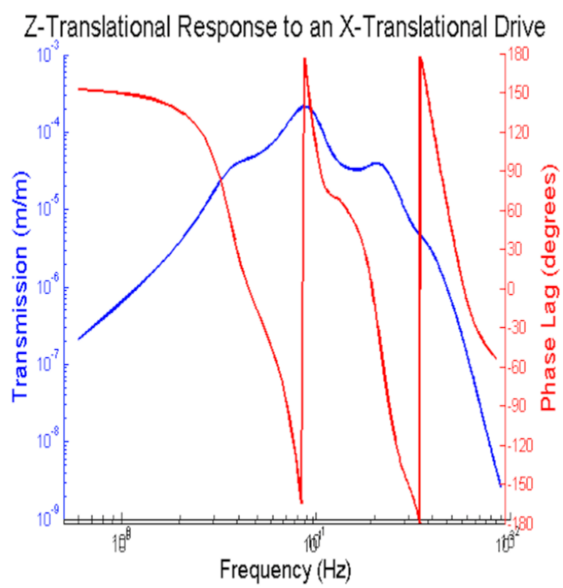
Figure 7: X-Drive Transfer Function of the MC1/MC3 Stack



(a) X-Translational Response



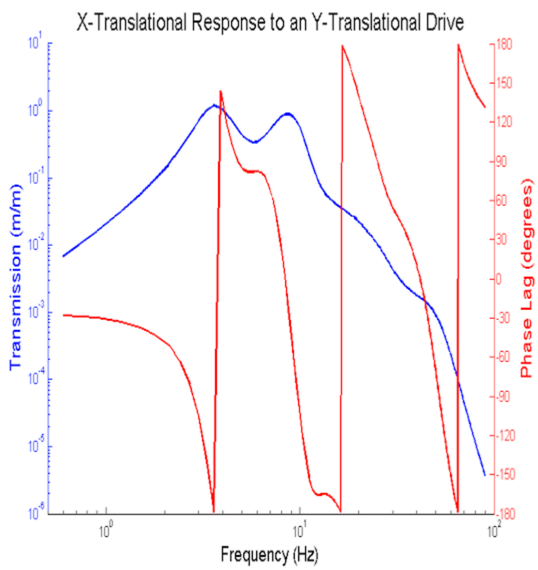
(b) Y-Translational Response



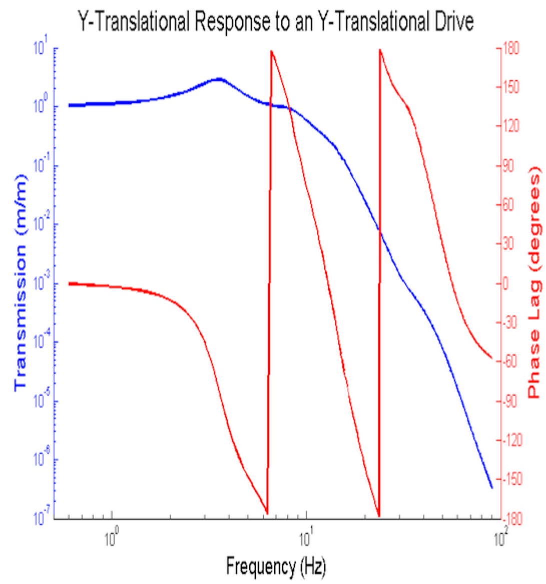
(c) Z-Translational Response

C.2 Y-Displacements

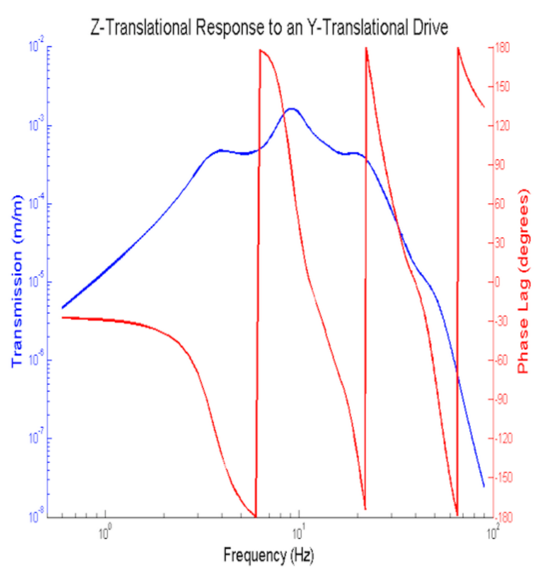
Figure 8: Y-Drive Transfer Function of the MC1/MC3 Stack



(a) X-Translational Response



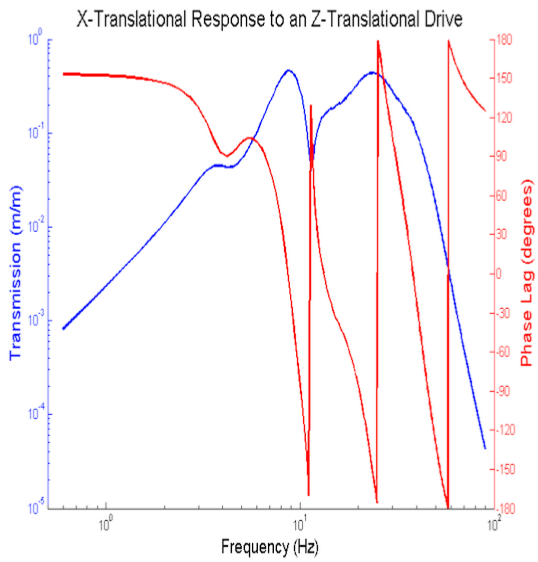
(b) Y-Translational Response



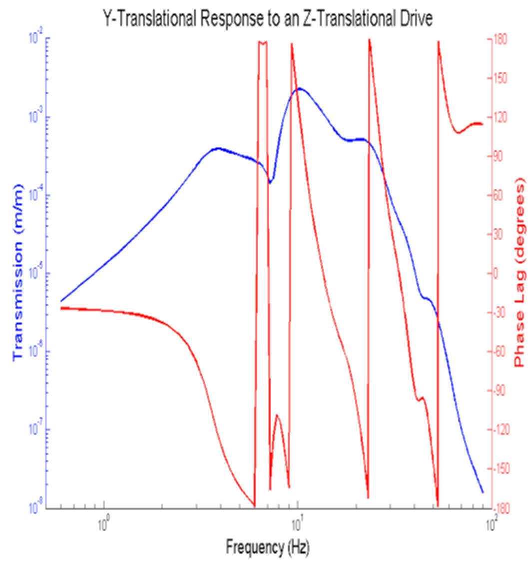
(c) Z-Translational Response

C.3 Z-Displacements

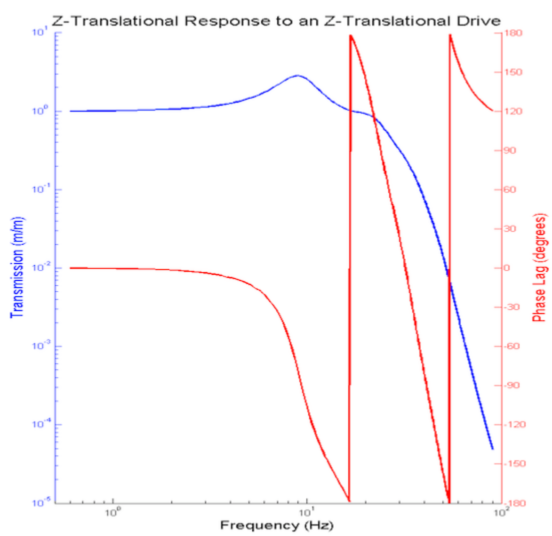
Figure 9: Z-Drive Transfer Function of the MC1/MC3 Stack



(a) X-Translational Response



(b) Y-Translational Response



(c) Z-Translational Response

D MATLAB Source Code

D.1 X-Displacements

```
function [] = data2xfer_x(xlsxfile)
% Given an Excel spreadsheet with extension, this script will give the
% transfer functions which characterize the response of the top of a
% stack, given a prespecified driving function.
%
% The format of the spreadsheet must contain this title: 'X Trans'. Each
% sheet must contain a total of 13 columns. The first column must contain
% a list of the frequencies used. The next 12 columns must contain
% the maximum displacement from equilibrium of the following:
%     1) The center of the bottom of the stack (Point O)
%     2) The center of the top of the stack (Point A)
%     3) A point displaced only in the y direction from A (Point B)
%     4) A point displaced only in the x direction from A (Point C)
% Each of these values must be given for displacements in the x, y,
% and z directions, producing a total of 12 vectors of data points.

% Read the file into number matrix A and character cell array B
[A,B] = xlsread(xlsxfile,'X Trans','A2:M300');

% Transform the cell array into a numeric matrix
X_Trans = str2double(B);

% Merge the matrices together
for(step = 1:size(X_Trans,2))
    if (isnan(X_Trans(1,step)))
        X_Trans(:,step) = A(:,step);
    end
end

% Extract the vectors
freq = X_Trans(:,1);
x_O  = X_Trans(:,2);
x_A  = X_Trans(:,3);
x_B  = X_Trans(:,4);
x_C  = X_Trans(:,5);
y_O  = X_Trans(:,6);
y_A  = X_Trans(:,7);
y_B  = X_Trans(:,8);
y_C  = X_Trans(:,9);
z_O  = X_Trans(:,10);
z_A  = X_Trans(:,11);
```

```
z_B  = X_Trans(:,12);
z_C  = X_Trans(:,13);

% X-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
xDisp  = abs(x_A) ./ abs(x_O);
xPhase = phase(x_A) - phase(x_O);
for(i = 1:length(xPhase))
    while((xPhase(i)<=(-pi))||(xPhase(i)>=(pi)))
        if(xPhase(i)<(-pi))
            xPhase(i)= xPhase(i) + 2*pi;
        else
            xPhase(i)= xPhase(i) - 2*pi;
        end
    end
end
xPhase = xPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure1 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure1,'YScale','log','YMinorTick','on',...
    'YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,xDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('X-Translational Response to an X-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure1,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,xPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Y-Translational Transfer Function
```

```
% (1) Calculate, keeping phase between +/- 180 degrees
yDisp  = abs(y_A) ./ abs(x_O);
yPhase = phase(y_A) - phase(x_O);
for(i = 1:length(yPhase))
    while((yPhase(i)<=(-pi))||(yPhase(i)>=(pi)))
        if(yPhase(i)<(-pi))
            yPhase(i)= yPhase(i) + 2*pi;
        else
            yPhase(i)= yPhase(i) - 2*pi;
        end
    end
end
yPhase = yPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure2 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure2,'YScale','log','YMinorTick','on',...
    'YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,yDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Y-Translational Response to an X-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure2,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,yPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Z-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
zDisp  = abs(z_A) ./ abs(x_O);
zPhase = phase(z_A) - phase(x_O);
for(i = 1:length(zPhase))
    while((zPhase(i)<=(-pi))||(zPhase(i)>=(pi)))
```

```
        if(zPhase(i)<(-pi))
            zPhase(i)= zPhase(i) + 2*pi;
        else
            zPhase(i)= zPhase(i) - 2*pi;
        end
    end
end
zPhase = zPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure3 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure3,'YScale','log','YMinorTick',...
    'on','YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,zDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Z-Translational Response to an X-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure3,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale',...
    'log','XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,zPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;
end
```

D.2 Y-Displacements

```
function [] = data2xfer_y(xlsxfile)
% Given an Excel spreadsheet with extension, this script will give the
% transfer functions which characterize the response of the top of a
% stack, given a prespecified driving function.
%
% The format of the spreadsheet must contain this title: 'Y Trans'. Each
% sheet must contain a total of 13 columns. The first column must contain
% a list of the frequencies used. The next 12 columns must contain
% the maximum displacement from equilibrium of the following:
%     1) The center of the bottom of the stack (Point O)
%     2) The center of the top of the stack (Point A)
%     3) A point displaced only in the y direction from A (Point B)
%     4) A point displaced only in the x direction from A (Point C)
% Each of these values must be given for displacements in the x, y,
% and z directions, producing a total of 12 vectors of data points.

% Read the file into number matrix A and character cell array B
[A,B] = xlsread(xlsxfile,'Y Trans','A2:M300');

% Transform the cell array into a numeric matrix
Y_Trans = str2double(B);

% Merge the matrices together
for(step = 1:size(Y_Trans,2))
    if (isnan(Y_Trans(1,step)))
        Y_Trans(:,step) = A(:,step);
    end
end

% Extract the vectors
freq = Y_Trans(:,1);
x_O  = Y_Trans(:,2);
x_A  = Y_Trans(:,3);
x_B  = Y_Trans(:,4);
x_C  = Y_Trans(:,5);
y_O  = Y_Trans(:,6);
y_A  = Y_Trans(:,7);
y_B  = Y_Trans(:,8);
y_C  = Y_Trans(:,9);
z_O  = Y_Trans(:,10);
z_A  = Y_Trans(:,11);
z_B  = Y_Trans(:,12);
z_C  = Y_Trans(:,13);
```

```
% X-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
xDisp  = abs(x_A) ./ abs(y_O);
xPhase = phase(x_A) - phase(y_O);
for(i = 1:length(xPhase))
    while((xPhase(i)<=(-pi))||(xPhase(i)>=(pi)))
        if(xPhase(i)<(-pi))
            xPhase(i)= xPhase(i) + 2*pi;
        else
            xPhase(i)= xPhase(i) - 2*pi;
        end
    end
end
xPhase = xPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure1 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure1,'YScale','log','YMinorTick','on',...
    'YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,xDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('X-Translational Response to an Y-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure1,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,xPhase,'Parent',axes2,'LineWidth',...
    2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Y-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
```

```
yDisp  = abs(y_A) ./ abs(y_O);
yPhase = phase(y_A) - phase(y_O);
for(i = 1:length(yPhase))
    while((yPhase(i)<=(-pi))||(yPhase(i)>=(pi)))
        if(yPhase(i)<(-pi))
            yPhase(i)= yPhase(i) + 2*pi;
        else
            yPhase(i)= yPhase(i) - 2*pi;
        end
    end
end
yPhase = yPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure2 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure2,'YScale','log','YMinorTick',...
    'on','YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,yDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Y-Translational Response to an Y-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure2,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,yPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Z-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
zDisp  = abs(z_A) ./ abs(y_O);
zPhase = phase(z_A) - phase(y_O);
for(i = 1:length(zPhase))
    while((zPhase(i)<=(-pi))||(zPhase(i)>=(pi)))
        if(zPhase(i)<(-pi))
```

```
            zPhase(i)= zPhase(i) + 2*pi;
        else
            zPhase(i)= zPhase(i) - 2*pi;
        end
    end
end
zPhase = zPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure3 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure3,'YScale','log','YMinorTick',...
    'on','YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,zDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Z-Translational Response to an Y-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure3,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,zPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;
end
```

D.3 Z-Displacements

```
function [] = data2xfer_z(xlsxfile)
% Given an Excel spreadsheet with extension, this script will give the
% transfer functions which characterize the response of the top of a
% stack, given a prespecified driving function.
%
% The format of the spreadsheet must contain this title: 'Z Trans'. Each
% sheet must contain a total of 13 columns. The first column must contain
% a list of the frequencies used. The next 12 columns must contain
% the maximum displacement from equilibrium of the following:
%     1) The center of the bottom of the stack (Point O)
%     2) The center of the top of the stack (Point A)
%     3) A point displaced only in the y direction from A (Point B)
%     4) A point displaced only in the x direction from A (Point C)
% Each of these values must be given for displacements in the x, y,
% and z directions, producing a total of 12 vectors of data points.

% Read the file into number matrix A and character cell array B
[A,B] = xlsread(xlsxfile,'Z Trans','A2:M300');

% Transform the cell array into a numeric matrix
Z_Trans = str2double(B);

% Merge the matrices together
for(step = 1:size(Z_Trans,2))
    if (isnan(Z_Trans(1,step)))
        Z_Trans(:,step) = A(:,step);
    end
end

% Extract the vectors
freq = Z_Trans(:,1);
x_O  = Z_Trans(:,2);
x_A  = Z_Trans(:,3);
x_B  = Z_Trans(:,4);
x_C  = Z_Trans(:,5);
y_O  = Z_Trans(:,6);
y_A  = Z_Trans(:,7);
y_B  = Z_Trans(:,8);
y_C  = Z_Trans(:,9);
z_O  = Z_Trans(:,10);
z_A  = Z_Trans(:,11);
z_B  = Z_Trans(:,12);
z_C  = Z_Trans(:,13);
```

```
% X-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
xDisp  = abs(x_A) ./ abs(z_O);
xPhase = phase(x_A) - phase(z_O);
for(i = 1:length(xPhase))
    while((xPhase(i)<=(-pi))||(xPhase(i)>=(pi)))
        if(xPhase(i)<(-pi))
            xPhase(i)= xPhase(i) + 2*pi;
        else
            xPhase(i)= xPhase(i) - 2*pi;
        end
    end
end
xPhase = xPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure1 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure1,'YScale','log','YMinorTick',...
    'on','YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,xDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('X-Translational Response to an Z-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure1,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,xPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Y-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
yDisp  = abs(y_A) ./ abs(z_O);
```

```
yPhase = phase(y_A) - phase(z_O);
for(i = 1:length(yPhase))
    while((yPhase(i)<=(-pi))||(yPhase(i)>=(pi)))
        if(yPhase(i)<(-pi))
            yPhase(i)= yPhase(i) + 2*pi;
        else
            yPhase(i)= yPhase(i) - 2*pi;
        end
    end
end
yPhase = yPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure2 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure2,'YScale','log','YMinorTick',...
    'on','YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,yDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Y-Translational Response to an Z-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure2,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,yPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;

% Z-Translational Transfer Function

% (1) Calculate, keeping phase between +/- 180 degrees
zDisp  = abs(z_A) ./ abs(z_O);
zPhase = phase(z_A) - phase(z_O);
for(i = 1:length(zPhase))
    while((zPhase(i)<=(-pi))||(zPhase(i)>=(pi)))
        if(zPhase(i)<(-pi))
            zPhase(i)= zPhase(i) + 2*pi;
```

```
        else
            zPhase(i)= zPhase(i) - 2*pi;
        end
    end
end
zPhase = zPhase*180/pi;

% (2) Plot, keeping common x-axis (yy-plot)
figure3 = figure('Color',[1 1 1]);
axes1 = axes('Parent',figure3,'YScale','log','YMinorTick','on',...
    'YColor',[0 0 1],'XScale','log','XMinorTick','on');
xlim(axes1,[0.5 100]);
hold(axes1,'all');
loglog(freq,zDisp,'Parent',axes1,'LineWidth',2);
ylabel('Transmission (m/m)','FontSize',14,'Color',[0 0 1]);
title('Z-Translational Response to an Z-Translational Drive',...
    'FontSize',16);
xlabel('Frequency (Hz)','FontSize',14);
axes2 = axes('Parent',figure3,'YTick',...
    [-180 -150 -120 -90 -60 -30 0 30 60 90 120 150 180],...
    'YAxisLocation','right','YColor',[1 0 0],'XScale','log',...
    'XMinorTick','on','ColorOrder',...
    [0 0.5 0;1 0 0;0 0.75 0.75;0.75 0 0.75;0.75 0.75 0;...
    0.25 0.25 0.25;0 0 1],'Color','none');
xlim(axes2,[0.5 100]);
ylim(axes2,[-180 180]);
hold(axes2,'all');
semilogx(freq,zPhase,'Parent',axes2,'LineWidth',2,'Color',[1 0 0]);
ylabel('Phase Lag (degrees)','VerticalAlignment','cap',...
    'FontSize',14,'Color',[1 0 0]);
hold off;
end
```

References

[1] Cantley, C. A., Hough, J., Robertson, N. A., & Greenhalgh, R. J. S. (1992). *Vibration Isolation Stacks for Gravitational Wave Detectors – Finite Element Analysis* [Monograph]. Review of Scientific Instruments, 63(4, Serial No. LIGO-P920015-00-R).

[2] Giaime, J., Saha, P., Shoemaker, D., & Sievers, L. (1995). *A Passive Vibration Isolation Stack for LIGO: Design, Modeling, and Testing* [Monograph]. Review of Scientific Instruments, 67(1, Serial No. LIGO-P952005-00-R).