

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

Technical Note	LIGO-T1500228-v1	2015/07/02
<b>Progress report 2: Sensing and control of suspended optic breadboard in Crackle2 experiment</b>		
Sai Kanth Dacha   Mentors: Gabriele Vajente, Rana Adhikari		

*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](mailto: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](mailto: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](mailto: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](mailto:info@ligo.caltech.edu)

LIGO-T1500228-v1

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

## Abstract

Crackling noise arises when a system responds to external conditions through discrete, impulsive releases of energy. Various locations in the Advanced LIGO setup have been identified over the years as suspected to produce crackling noise. Of specific interest is the crackling noise arising in maraging steel blades that are used to suspend end mirrors of the Michelson Interferometer in a quadruple pendulum stage, where vertical displacement noise due to crackle events couples to horizontal displacement noise along the arm of the interferometer.

In order to study this noise experimentally, a setup was designed a couple of years ago: a Michelson Interferometer configuration to measure vertical displacement noise at the tip of the maraging steel blade. One of the major difficulties in measuring crackle noise using this setup has been the high sensitivity required [1], and having a good seismic isolation system with a good control over the motion of the interferometer is essential.

What I describe in this paper focuses on my work towards establishing a feedback damping control loop around the suspended optics breadboard which houses the interferometer. I describe the whole process of my control system design, starting from sensing to actuation and control. I also discuss certain miscellaneous things I have worked on to improve speed and efficiency for further rounds of implementing the control system, upon any possible modifications to the setup. I conclude by talking about the improvement in seismic isolation this control system has brought about.

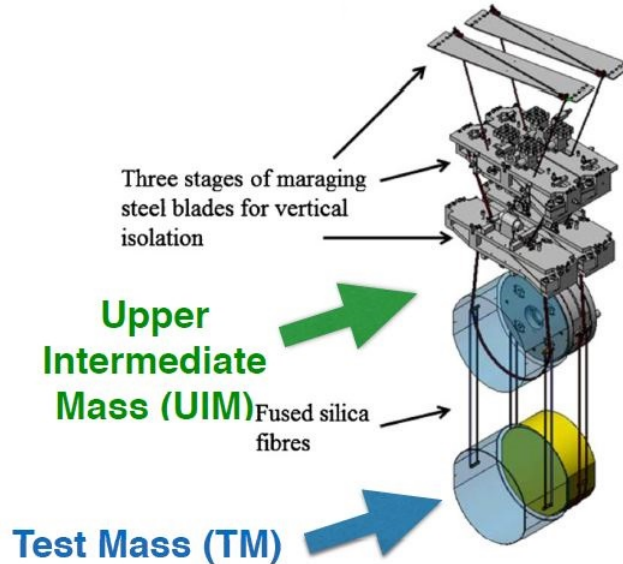
## 1 Introduction

In metals, dislocations are 'pinned' by obstacles like grain boundaries or other surfaces. (Dislocations are dis-junctions in the periodic lattice structure.) Under small oscillatory stress, these dislocation lines bow in and out, but the response of the complex network on the whole is known to act nonlinearly through long-range interactions between dislocations. This nonlinear behavior, among a broad class of other nonlinear phenomena, is known to be the cause of "crackling". "Crackling" here refers to impulsive releases of energy, acoustic emissions, or changes in the geometry of attachments between suspension elements. It has been suspected that this "crackling noise" in various components and suspensions might produce excess noise in aLIGO. [1]

Many possible locations of crackle have been identified, some of which are:

- The maraging steel blades used for vertical isolation
- The silica fibers which suspend the test masses from the penultimate masses
- The welds which attach the fibers to the ears
- The clamps which hold the suspension wires to the steel blades

Figure 1 shows the test mass suspension scheme in Advanced LIGO.



**Figure 1: Suspension scheme in Advanced LIGO**

The maraging steel blades have been under investigation in the first crackle experiments, as they present a mechanical system which can be driven and stressed easily. In addition, mechanical crackling noise is sufficiently generic that an experiment capable of measuring this noise in maraging steel blade springs will be well suited for investigating crackling in other components used in aLIGO as well.

As crackling noise is inherently nonlinear, there is the potential for noise to be upconverted. Specifically, motion of the suspension at sub-Hz microseismic frequencies may induce blade motion, causing the blades internal stresses to fluctuate, resulting in an avalanche of crackle events with high-frequency content. As we do not currently possess a reliable analytical model to predict the magnitude, or frequency dependence, of these events, we hope to do so experimentally directly at the frequencies of interest. [1]

## 2 Where I started

**This section describes the work that was already done before my arrival.** I picked up one part of the Crackle2 experiment - suspension damping of the breadboard - and I started off in direction of establishing a feedback damping control loop for it.

### 2.1 Measurement strategy

With reference to figure 1, below the upper intermediate mass (UIM) in the quadruple suspension system, there is no more spring blade isolation, thus any crackling noise in the UIM maraging steel

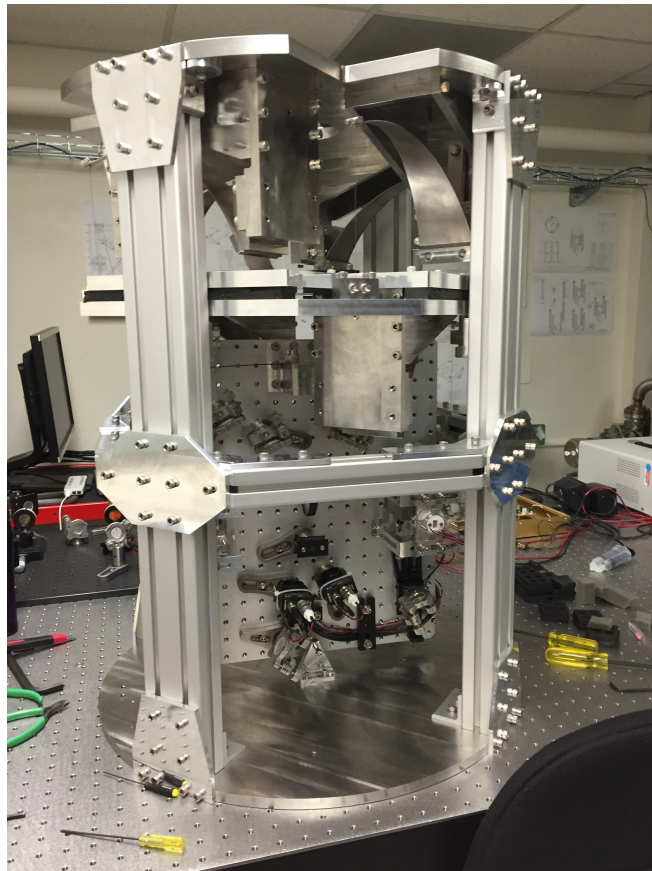
blades will propagate directly to the test mass. Therefore, one would want to ensure low enough crackling noise at the UIM blade tip itself.

A direct measurement of crackling noise is very difficult. However, one can make measurements of the blade displacements directly using a Michelson interferometer with end mirrors mounted to loaded blade springs which are driven with a low frequency, common-mode force. Since crackling noise occurs incoherently in each blade, it will show up in the Michelson's displacement signal. In order to ensure repeatability and applicability of results, the setup has been made to be similar to the existing aLIGO configuration.

## 2.2 Experimental design

The setup consists of a Michelson interferometer using blade-suspended masses as end mirrors. A crackle event will change the differential displacement of the mirrors, and hence be reflected in the interferometer output. The events are excited by a low frequency, common-mode, drive on the two blades. [1] The apparatus is going to be housed in a vacuum chamber to mitigate acoustic noise.

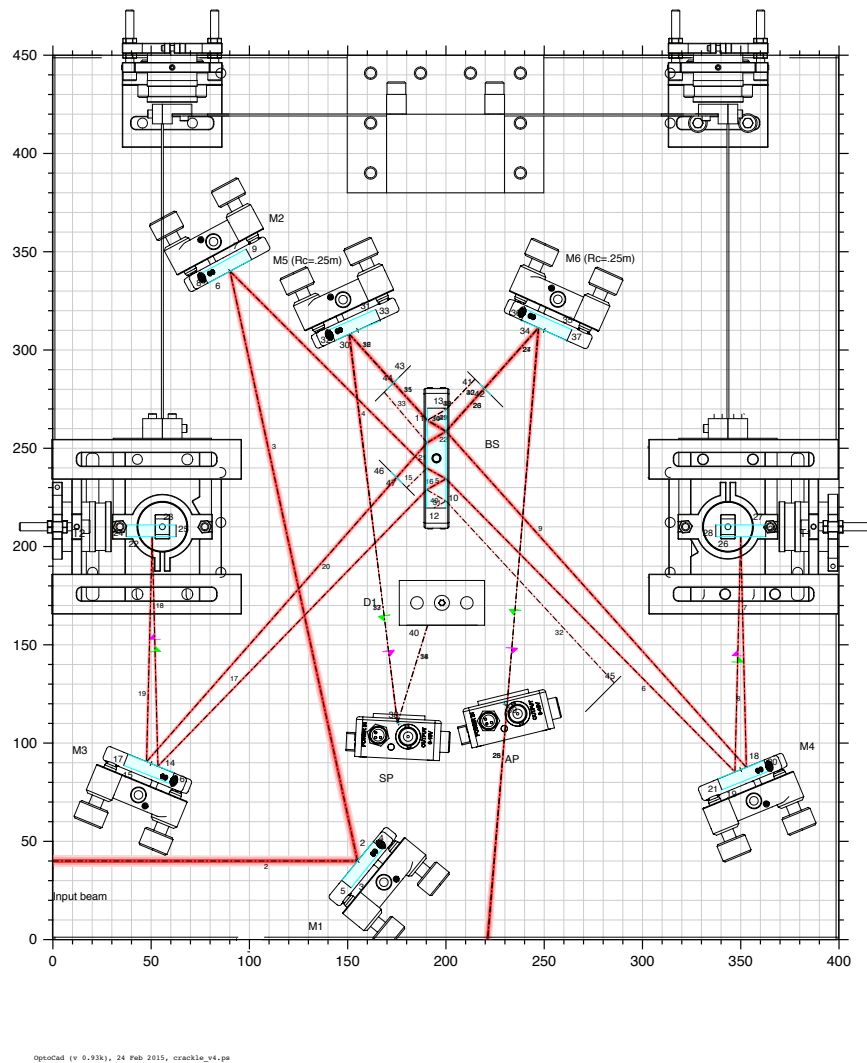
Figure 2 is a picture of the full setup.



**Figure 2: Complete setup**

## 2.2.1 Optical layout

The optical layout schematic in figure 3 below.



**Figure 3: Optical layout**

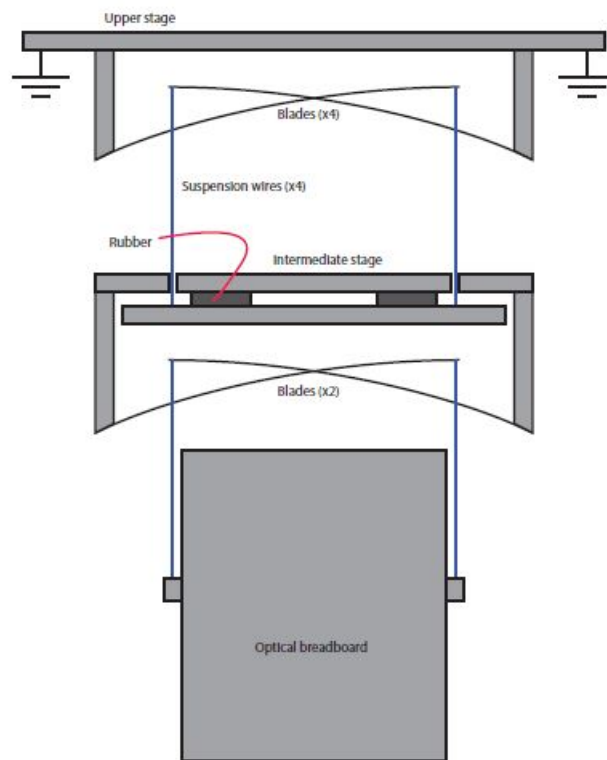
Light from the laser comes in from the bottom left corner through a viewport. Two folding mirrors then redirect the beam to the beam splitter. The two arms of the Michelson interferometer are folded in such a way that the beams impinging on the end mirrors are almost vertical, but tilted enough so that the beams propagating in opposite directions (before and after reflection from the mirror) are separate. The end mirrors of the Michelson arms are horizontal.

## 2.2.2 Seismic isolation system

Earlier versions of the setup used a stack of two steel plates resting on blocks made of rubber to provide isolation of the Michelson from the ground motion. Ideally, seismic motion of the optical setup wouldn't couple to the Michelson signal because the motion would be common to both mirrors. However, any differential motion of the blades would result in a spurious signal.

Lately, a suspension system has been employed for the breadboard with the Michelson interferometer. A basic control of the breadboard has already been established, and a more detailed modeling of the system and damping in all six d.o.f. is to be implemented.

Figure 4 below shows a simplified scheme of the suspension system. Vertical isolation has been achieved using maraging steel spring blades. A two stage system has been designed. The upper stage is composed of four such blades, each one supporting a wire which is attached to an intermediate stage.



**Figure 4: Scheme of the suspension system**

## 3 Progress

The problem statement for my project, to put it in very simple words, is to implement a damping feedback control loop for the suspended optical breadboard. The following subsections are written

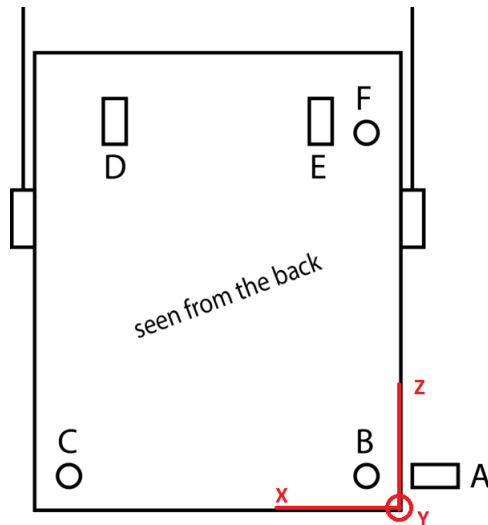
in an order so as to build up to the said aim.

### 3.1 Sensing motion of breadboard using OSEM

This subsection talks about using the suspension OSEM to track the motion of the breadboard in physical degrees of freedom.

#### 3.1.1 Analytical model

Figure 5 shows the mounting scheme of the six OSEMs.



**Figure 5: Scheme of OSEMs**

It is possible to use the outputs of these shadow sensors to sense motion of the suspended breadboard. Here, I build an analytical model for the same.

I start with the basic relation that gives the velocity  $\mathbf{v}_p$  at a point P whose position vector is  $\mathbf{r}_p$  on a rigid body whose center of mass (CM) moves with a velocity  $\mathbf{v}_{cm}$ . The angular velocity is  $\boldsymbol{\omega}$ . The relation can be written as:

$$\mathbf{v}_p = \mathbf{v}_{cm} + \boldsymbol{\omega} \times \mathbf{r}_p \quad (1)$$

In an infinitesimal time interval, the equation can be rewritten as:

$$\mathbf{s}_p = \mathbf{s}_{cm} + d\boldsymbol{\theta} \times \mathbf{r}_p \quad (2)$$



where now,  $\mathbf{s}_p$  is the infinitesimal displacement vector of the point P,  $\mathbf{s}_{cm}$  is the infinitesimal displacement vector of the center of mass and  $d\boldsymbol{\theta}$  represents infinitesimal rotation of the position vector of point P,  $\mathbf{r}_p$ , about the center of mass.

Equation (2) can be written in the matrix form as:

$$\begin{pmatrix} s_{p,x} \\ s_{p,y} \\ s_{p,z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & -r_{p,z} & r_{p,y} \\ 0 & 1 & 0 & r_{p,z} & 0 & -r_{p,x} \\ 0 & 0 & 1 & -r_{p,y} & r_{p,x} & 0 \end{pmatrix} \begin{pmatrix} s_{cm,x} \\ s_{cm,y} \\ s_{cm,z} \\ d\theta_x \\ d\theta_y \\ d\theta_z \end{pmatrix} \quad (3)$$

Here,  $s_{p,x}$  corresponds to displacement of point P in the x direction.

Now, in the case of interest here, each of the 6 sensors has one such matrix equation. However, note that by construction, **each OSEM can measure displacement for only motion along its axis**. Coupling this with the fact that each of the six sensors are mounted such that their axis is along one of the principal axes (as can be seen in figure 5), it is clear that **only one of the three linear equations represented in equation (3) becomes relevant**.

Let us see this for an example: sensor A. The equation corresponding to  $s_{a,x}$  alone becomes relevant because the axis of sensor A is along the absolute (w.r.t ground) x-axis. Therefore, the exact linear equation governing  $s_{a,x}$  can be written down by finding out the values of  $r_{a,z}$  and  $r_{a,y}$ . The same can be repeated for all the 6 sensors to obtain a set of 6 linear equations. Notice that the terms on the left hand side of these equations correspond to displacements along sensor axes, and those on the right hand side to motion of the center of mass and rotation about the absolute (w.r.t. ground) axes: roll, pitch and yaw. The desired matrix transformation equation can, then, be established from these six equations, and is the following:

$$\begin{pmatrix} s_a \\ s_b \\ s_c \\ s_d \\ s_e \\ s_f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & -z_{cm,A} & y_{cm,A} \\ 0 & -1 & 0 & z_{cm,B} & 0 & -x_{cm,B} \\ 0 & -1 & 0 & z_{cm,C} & 0 & -x_{cm,C} \\ 0 & 0 & 1 & -y_{cm,D} & x_{cm,D} & 0 \\ 0 & 0 & 1 & -y_{cm,E} & x_{cm,E} & 0 \\ 0 & -1 & 0 & z_{cm,F} & 0 & -x_{cm,F} \end{pmatrix} \begin{pmatrix} s_{cm,x} \\ s_{cm,y} \\ s_{cm,z} \\ d\theta_x \\ d\theta_y \\ d\theta_z \end{pmatrix} \quad (4)$$

Here, as usual,  $x_{cm,A}$  refers to the x-coordinate of A in the CM frame. Also recall that these values can be negative too, since the center of mass is close to the geometric center of the breadboard and OSEMs are placed on either side.

### 3.1.2 Constructing the transformation: Sensing matrix

As mentioned above, the sensing transformation matrix can be arrived at using the set of 6 linear equations. The positions of the sensors **w.r.t the CM** fix the constants of the matrix.

#### 1. Finding coordinates of the center of mass (CM)

First, we find out the coordinates of the CM. The origin is as shown in figures 3 and 5. The suspension is as shown in figure 4. The axis convention is shown in figure 5.

The y-coordinate can be simply found to be -27 (in mm) by noticing the position of the suspension wires from the center plane of the breadboard.

For the x and z coordinates, I have estimated the mass and position of various components of the setup. Table 1 summarizes the same.

Table 1: Calculation of CM coordinates of the setup

Item(s)	Mass (kg)	Absolute coordinates (in xz plane, origin as shown in figure 5) (mm)	Comments
Optical posts and components	12 x 0.4 = 4.8	(217.83, 177.75)	0.2kg is the weight of each post, and 0.2kg accounts for the mass of the component placed on it. Figure 2 shows the positions of these posts/components. The coordinates posted here correspond to the CM of the system of posts and components
Suspended block 1	2.2	(356, 210)	
Suspended block 2	2.2	(44, 210)	
Fixed block	1.45	(200, 400)	This is the block to which steel blades are clamped
Breadboard	5.9	(200, 225)	
Reinforcement 1	1.97	(380, 225)	Mass calculated from material and dimensions
Reinforcement 2	1.97	(20, 225)	Mass calculated from material and dimensions

Upon calculation, the center of mass of the system comes out to be: 20.49kg at **(204, -27, 223)** (in mm).

#### 2. Calculation of positions of sensors w.r.t CM

The next thing I did was to find out the positions of each of the sensors in the absolute (w.r.t. ground) reference frame, and then find

their coordinates in the center of mass frame. Refer to table 2. 'X' means that we don't care about that particular coordinate since it is along the axis of the sensor.

Table 2: Positions of sensors

Sensor	Absolute coordinates (in xz plane, origin as shown in figure 5) (mm)	Coordinates in CM frame (mm)
A	(X, 0, 62.5)	(X, 27, -160.6)
B	(24, X, 62.5)	(-180.16, X, -160.6)
C	(376, X, 62.5)	(171.82, X, -160.6)
D	(316, 25.7, X)	(111.82, 52.7, X)
E	(84, 25.7, X)	(-120.18, 52.7, X)
F	(24, X, 397)	(-180.18, X, 173.91)

The matrix transformation equation can, after converting the length units to meter (so that angles are in micro-rad), then be written as:

$$\begin{pmatrix} s_a \\ s_b \\ s_c \\ s_d \\ s_e \\ s_f \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0.1606 & 0.027 \\ 0 & -1 & 0 & 0.1606 & 0 & -0.18018 \\ 0 & -1 & 0 & 0.1606 & 0 & 0.17182 \\ 0 & 0 & 1 & -0.0527 & 0.11182 & 0 \\ 0 & 0 & 1 & -0.0527 & -0.12018 & 0 \\ 0 & -1 & 0 & -0.17391 & 0 & -0.18018 \end{pmatrix} \begin{pmatrix} s_{cm,x} \\ s_{cm,y} \\ s_{cm,z} \\ d\theta_x \\ d\theta_y \\ d\theta_z \end{pmatrix} \quad (5)$$

On multiplying the above equation on both sides by the inverse of the matrix, we get:

$$\begin{pmatrix} s_{cm,x} \\ s_{cm,y} \\ s_{cm,z} \\ d\theta_x \\ d\theta_y \\ d\theta_z \end{pmatrix} = \begin{pmatrix} 1 & 0.0774 & -0.0767 & -0.6922 & 0.6922 & -6.88 \times 10^{-4} \\ 0 & -0.0034 & -0.5119 & 0 & 0 & -0.4847 \\ 0 & 0.1575 & 0 & 0.5180 & 0.4820 & -0.1575 \\ 0 & 2.989 & 0 & 0 & 0 & -2.989 \\ 0 & 0 & 0 & 4.31 & -4.31 & 0 \\ 0 & -2.87 & 2.84 & 0 & 0 & 2.5478 \times 10^{-2} \end{pmatrix} \begin{pmatrix} s_a \\ s_b \\ s_c \\ s_d \\ s_e \\ s_f \end{pmatrix} \quad (6)$$

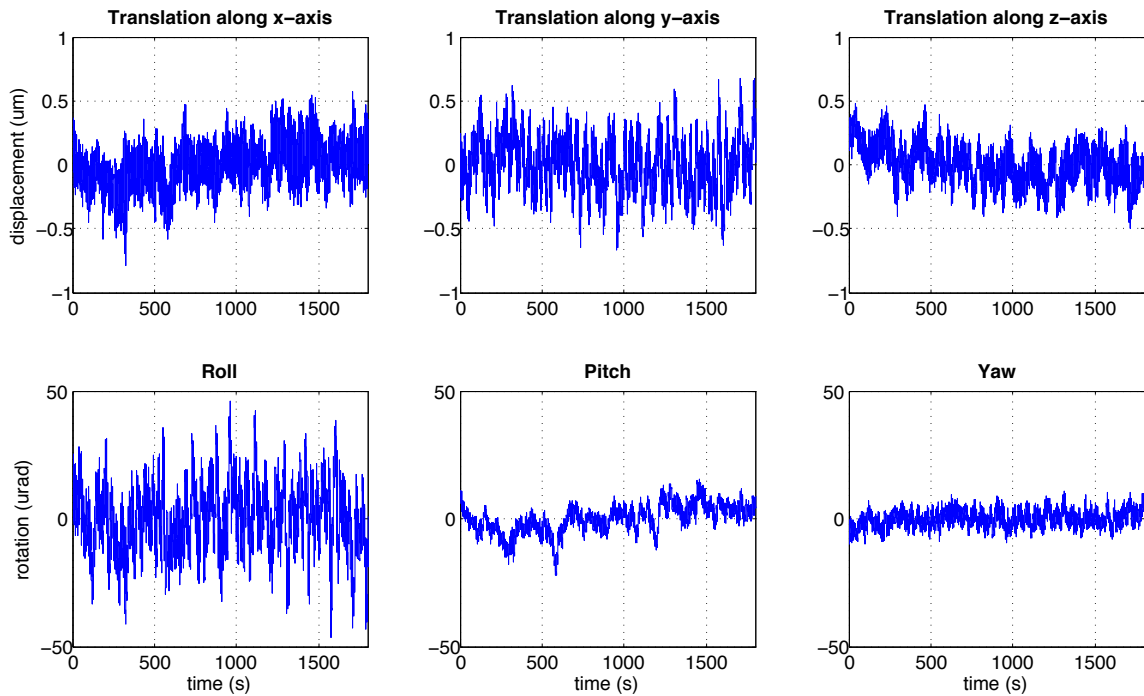
Once again, recall that the angles are in micro-rad and displacements in micron.

### 3.2 Reconstruction of breadboard motion in 6D

Having constructed the sensing matrix, the next thing I did was to try it out on some real data. The sensor data was exported to struct files using the ligoDV interface. (The MATLAB ligoDV interface can be used to acquire sensor data, from a specified period of time, from the relevant server. In this report, I will not delve into any technical details of the interface or the settings.)

I wrote a MATLAB script that extracted sensor data arrays in double format from the struct files. Since we are interested in infinitesimal displacements around the equilibrium point, I subtracted from each sensor data array the mean of the data. Then, I inverted the transformation matrix from equation 5 to reconstruct motion in the ground frame.

Figure 6 shows plots of motion in 6 d.o.f.



**Figure 6: Plot of motion in 6 d.o.f**

Notice that the order of magnitude of motion is only a fraction of a micron. These plots cannot really reveal any interesting information. What can be more interesting is the spectra of these motions: one can see the modes and resonance frequencies.

### 3.3 Spectral analysis

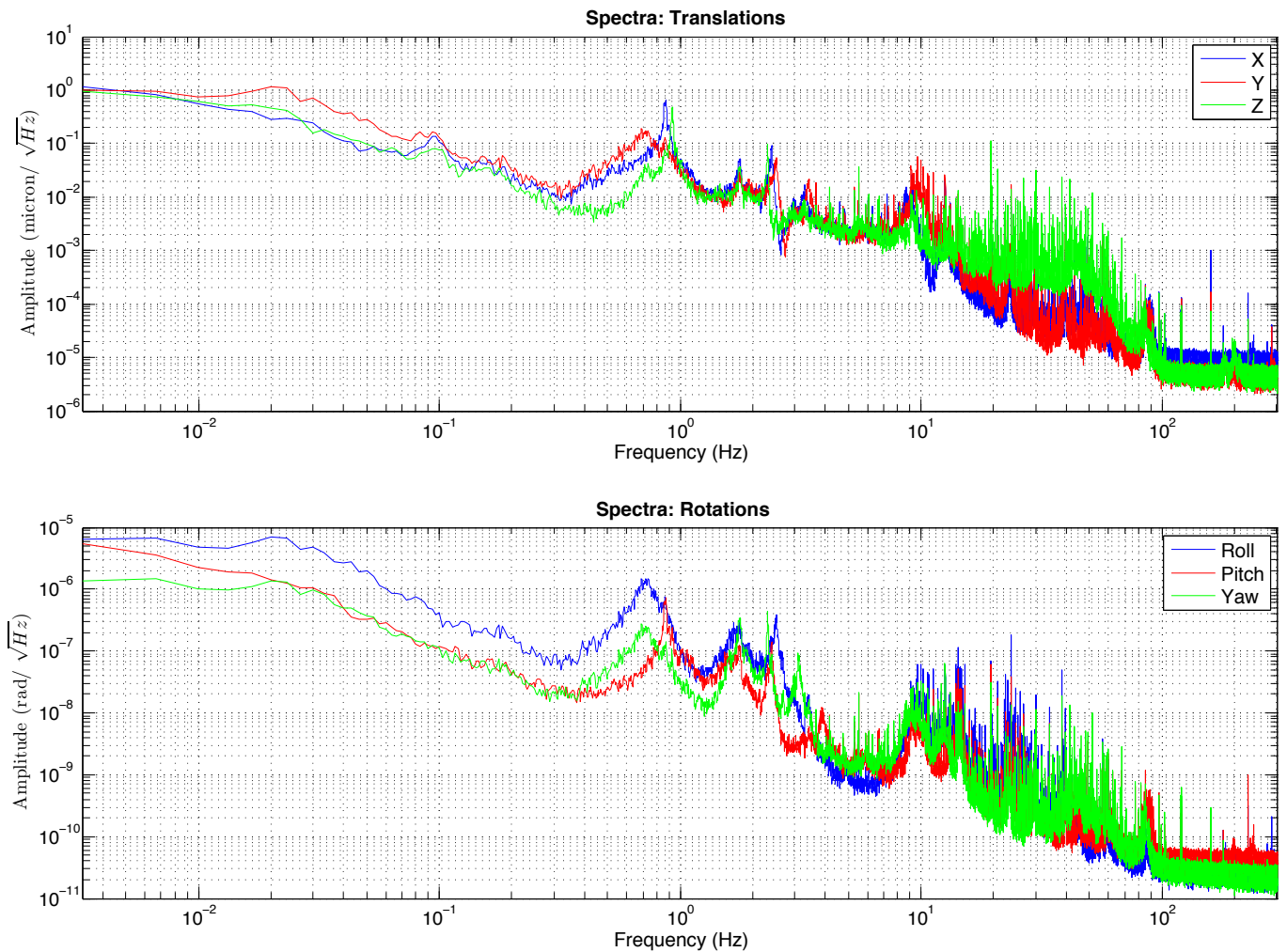
I have used the pwelch function in MATLAB to find the power spectral density (PSD) of the collected motion data. The following points summarize the process.

- $[P_{xx}, f] = \text{pwelch}(\text{data}, \text{hanning}(\text{nfft}), \text{nfft}/2, \text{nfft}, \text{fs})$

on MATLAB gives the PSD  $P_{xx}$  and the corresponding frequency array  $f$

- nfft was chosen to be 1/6th of the total number of points, so as to guarantee about 10 averages through hanning windowing.
- fs represents the sampling frequency, it is equal to 2048Hz.
- The square root of the PSD, amplitude, was used in the plots.

Figure 7 shows the spectra corresponding to motion shown in figure 6.



**Figure 7: Spectra of motion in 6 d.o.f**

From the plots in figure 7, one can infer that:

1. The breadboard is suspended, and the OSEMs are fixed relative to the ground. Most of what is seen in the spectrum is due to motion of the ground (seismic activity); this seems to be concentrated in the  $<10\text{Hz}$  region. **This is what forms the region of interest for us:**

**the aim would be to damp motion at resonant frequencies of the suspension. Seismic isolation has been achieved by using a suspension, but the modes of the suspension are in the 0.1-10Hz region. At these frequencies, the breadboard moves more than the ground, and so motion must be damped.**

2. All the plots peaks at a particular (common) frequency just below 1Hz – one of the mode frequencies of the system. A few more peaks can be seen for a few frequencies in between 1Hz and 10Hz.

**Note:** As we later realized, the magnets **might have been**, in the past, touching the OSEMs. The spectra or data, hence, is not completely reliable.

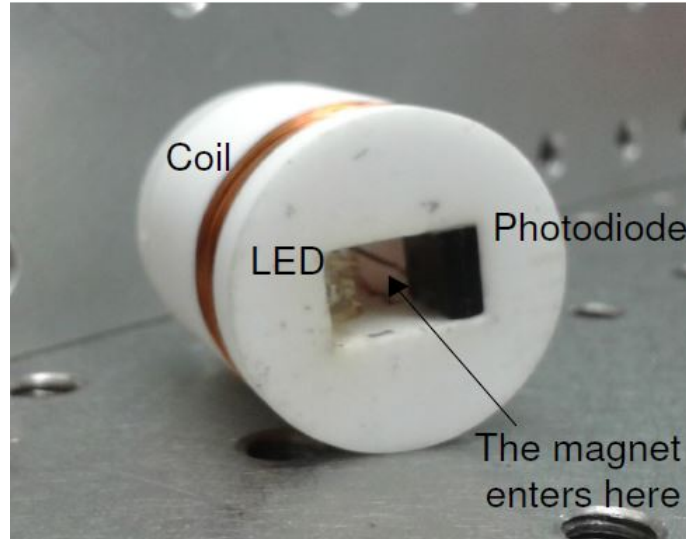
The idea is to damp motion of the breadboard at resonant frequencies. This would require us to have an idea of the mechanical response of the system for damping using OSEM coils at positions A through F. This forms the next part of my work.

### **3.4 Characterization of mechanical response of the suspension**

Six OSEMs, A through F, are mounted at different positions in the cage in which the breadboard is suspended. The scheme has been depicted in figure 5. It is important to understand how OSEMs function in order to get a feel for these measurements.

#### **3.4.1 Detour: A brief about sensing and actuation using OSEM**

**O**ptical **S**hadow Sensor and **E**lectro**M**agnetic Actuator (OSEM) consists of a shadow sensor part and an electromagnetic actuator part. The former consists of an LED and a photodetector mounted opposite to each other, as shown in figure 8. Insertion of an object through the hole would block the light from the LED reaching the photodetector, and this change can be used to quantify the extent to which the object has come inside the hole. The actuator part consists of a coil which can generate magnetic field upon driving current through it.



**Figure 8: Optical Shadow Sensor and Electromagnetic Actuator**

Upon thought, one can easily see that both sensing and actuation can be done by the same unit by simply using a magnetic material for the object whose displacement is to be sensed; and this is precisely what is done: magnets are attached to the points on the breadboard indicated by points A through F. The OSEM modules are mounted outside the suspension, and they are at rest w.r.t the ground.

Then, motion of the magnet can be sensed as usual, and it can even be driven by varying the current through the coil. This is what is used as an actuation for the breadboard.

### 3.4.2 Measurement of transfer functions

Characterization of the mechanical response of the system consists of exciting one of the coils and recording response in all other sensors or in motion in physical d.o.f. Both are equivalent and are related by the same transformation as the displacements are. I have adopted the first method: that is, record sensor signals and reconstruct the physical d.o.f. motion offline. The following will explain the same in detail and discuss some results.

As I discussed above, by using magnets, it becomes possible for one to modify as well as sense motion of a point on the suspended optical breadboard. Transfer function measurement, then, becomes a matter of exciting coils and measuring sensor displacements; the LIGO DTT provides a pretty convenient way of doing this. Before I talk about the results, it is important to note the following about the DTT configuration used for the measurements.

- The DTT is operated in a **swept-sine mode** in order to perform the desired measurements. The choice is obvious: we wish to study the behavior of sensor displacement for various frequencies of excitation.

- I have, in the measurements that I describe below, swept the excitation frequency from **0.5 Hz to 15 Hz**.
- It is important to realize that the number of counts output to the coil under excitation depends on the frequency of excitation. This is because of the presence of whitening and de-whitening filters. Given the filter shapes, this means that at lower frequencies, due to lower gain from whitening and de-whitening filters, the output counts are lower as compared to that at higher frequencies **for the same input amplitude**. It is also known that a higher amplitude of excitation gives higher coherence. Combining both of the above facts, it is interesting to see that one can improve coherence at lower frequencies by exciting the coil at higher amplitudes than those at higher frequencies for a similar number of counts output to the coil. As long as the number of counts output to the coil is below a saturation limit (which is about 16000), this is perfectly safe.

This was exploited, and the excitation amplitude profile was modified as follows:

Table 3: Excitation amplitude profile

<b>Frequency</b>	<b>Excitation amplitude (uN)</b>	<b>No. of counts output to coil</b>
15	400	10000
10	600	10000
5	1200	10000
2	2800	10000
1	4700	10000
0.5	6300	10000

It is possible to command DTT to build an envelope around the above points and then use that profile to define amplitudes for other frequencies in the range of measurement; this is what I have done for the measurements whose results are shown below.

As mentioned earlier, I have first measured transfer functions of coil excitation to sensor displacement and then transformed them to coil excitation to physical d.o.f. motion transfer functions. The transformation matrix remains the same as the sensing transformation matrix in (6). Here are the results.



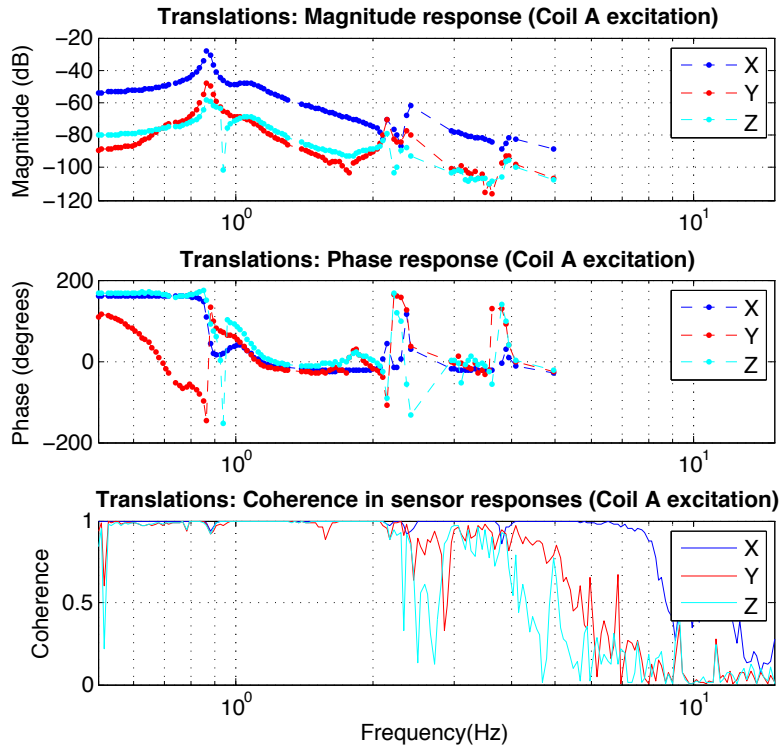


Figure 9: Bode plots for translational d.o.f. for coil A excitation

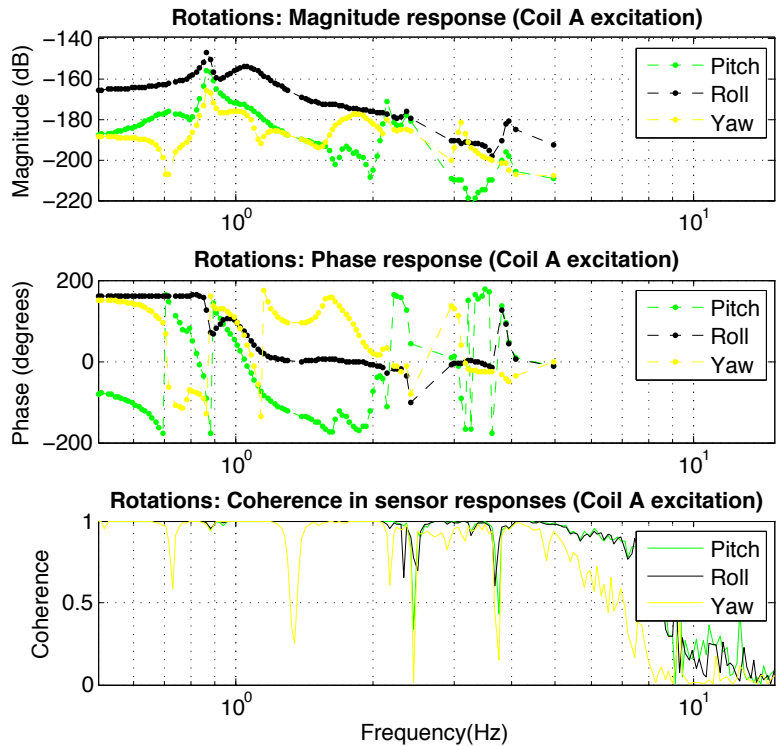


Figure 10: Bode plots for rotational d.o.f. for coil A excitation

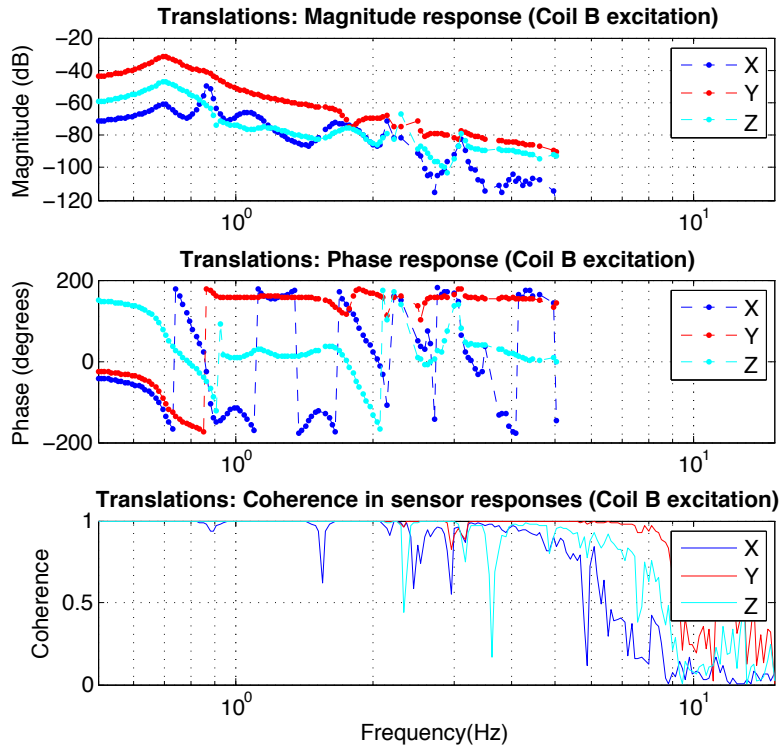


Figure 11: Bode plots for translational d.o.f. for coil B excitation

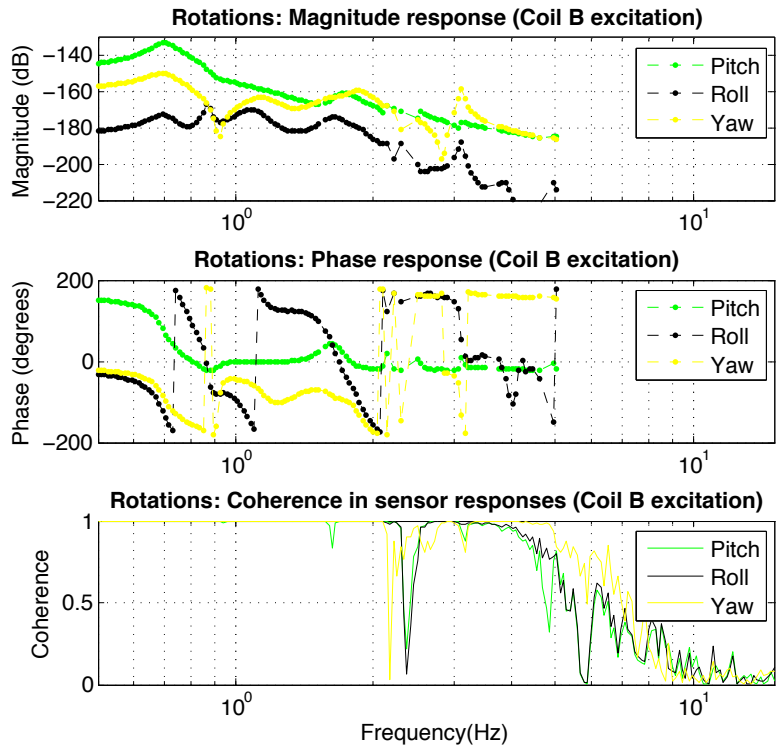


Figure 12: Bode plots for rotational d.o.f. for coil B excitation

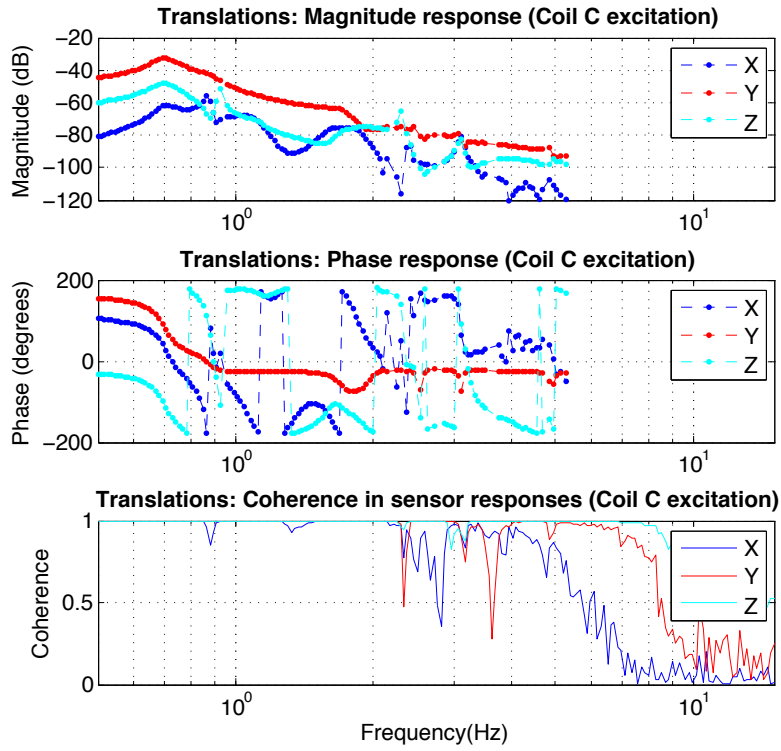


Figure 13: Bode plots for translational d.o.f. for coil C excitation

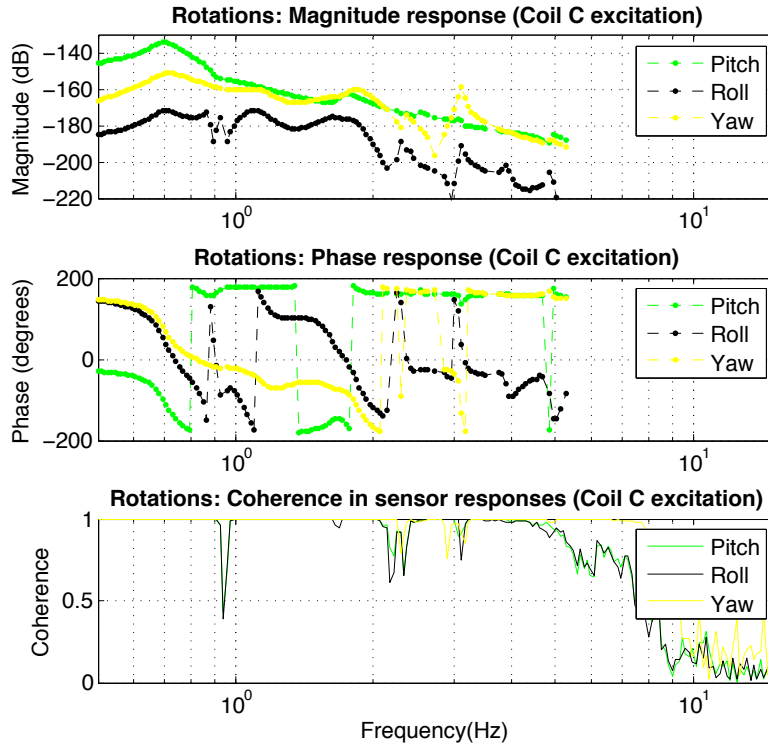


Figure 14: Bode plots for rotational d.o.f. for coil C excitation

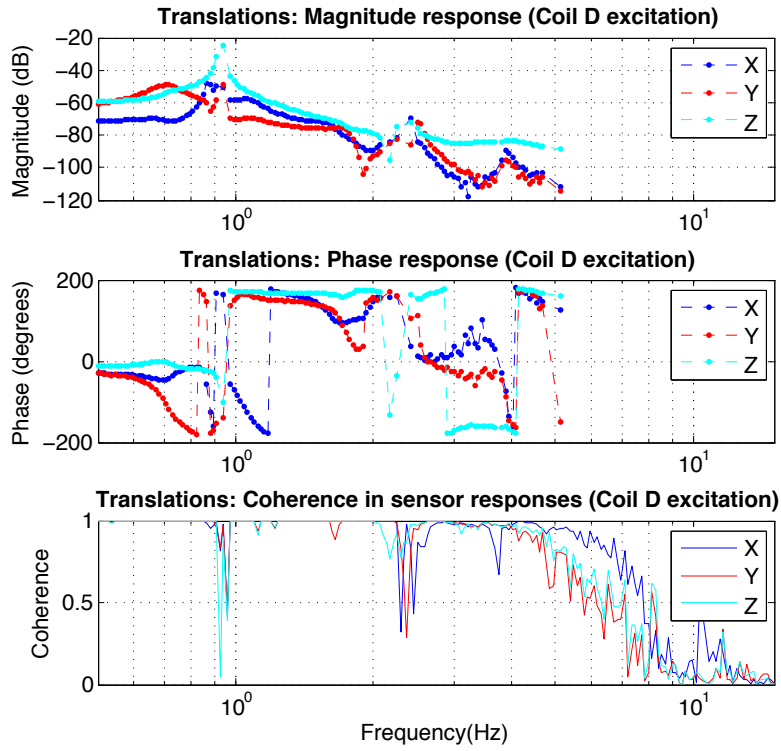


Figure 15: Bode plots for translational d.o.f. for coil D excitation

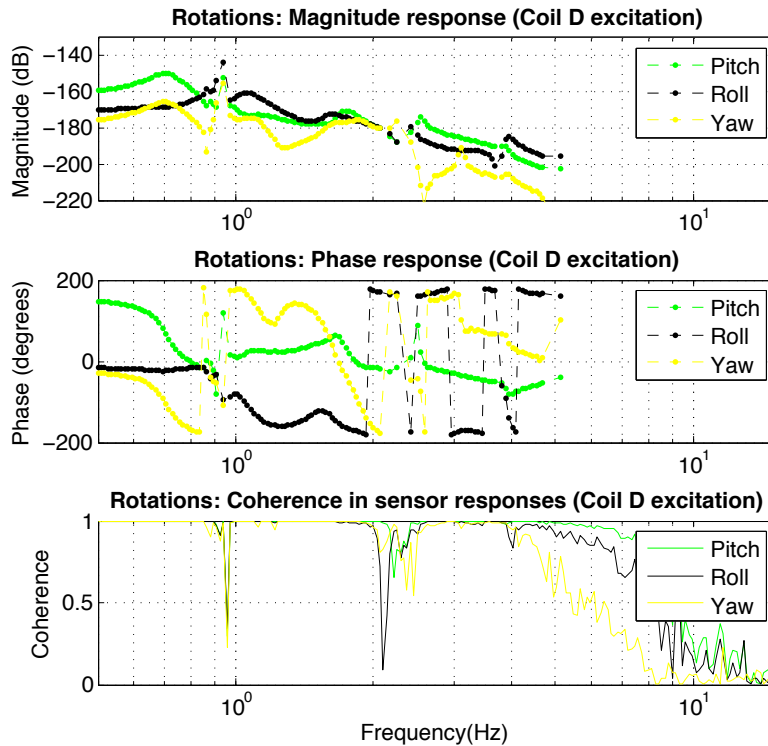


Figure 16: Bode plots for rotational d.o.f. for coil D excitation

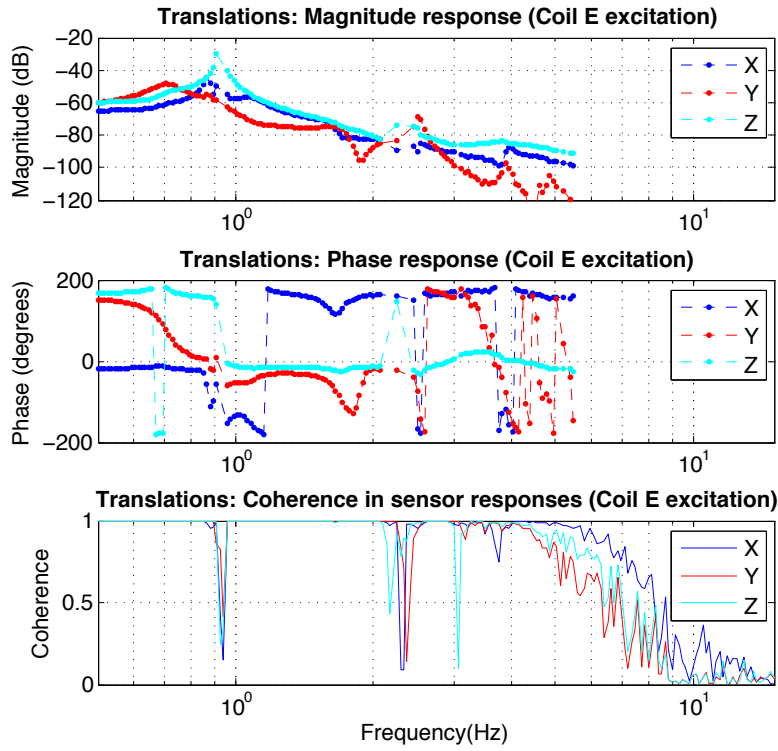


Figure 17: Bode plots for translational d.o.f. for coil E excitation

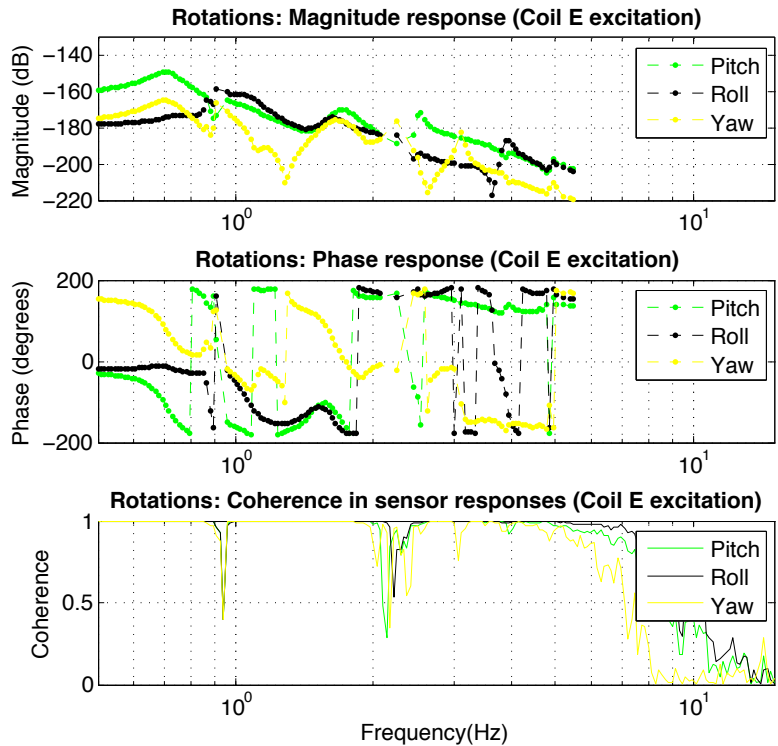
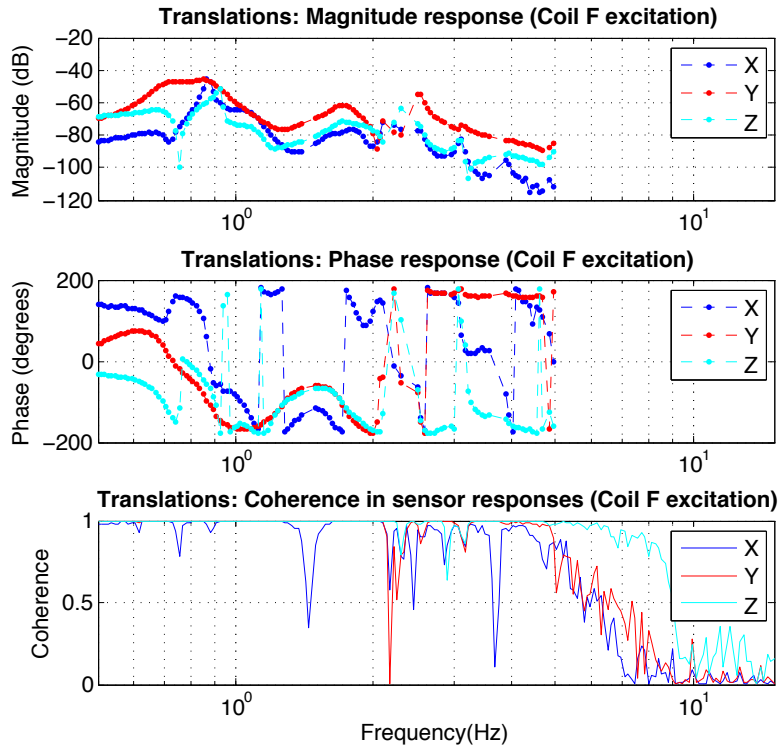
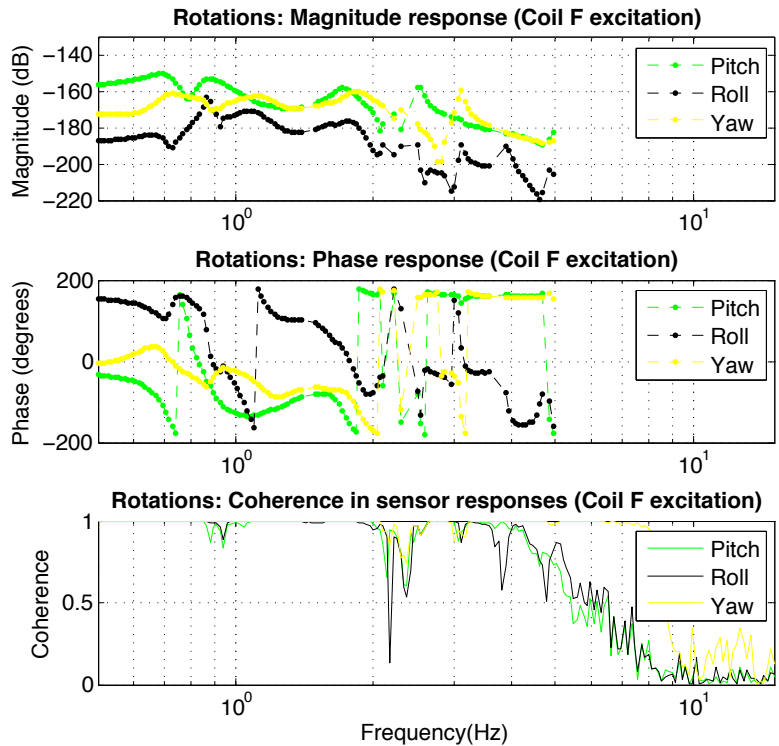


Figure 18: Bode plots for rotational d.o.f. for coil E excitation



**Figure 19: Bode plots for translational d.o.f. for coil F excitation**



**Figure 20: Bode plots for rotational d.o.f. for coil F excitation**

Figures 9 through 20 show magnitude and phase response of motion along physical d.o.f. (that is, after reconstruction) for various excitation of coils A through F. There was no kind of damping turned on while these measurements were performed. One can clearly see peaks around the 0.7Hz - 1.0Hz region. This represents a resonance frequency of the system.

Since the system is like a double pendulum, one would expect two resonance peaks. However, these plots show only one. The reason might be that there were certain problems with the mechanical setup, such as some part of the suspension setup touching the cage, or some magnets touching the OSEMs. This is also supported by the fact that the Q of the poles are not as high as we had expected.

The legs of the optics table were changed from rigid, longer ones to shorter ones that have the capability of "floating" the table in air. (More about this in the next subsection.) As I realized, the problem was solved when we did another round of realignment after the legs of the breadboard were changed. Some of these results as well can be found in the next subsection.

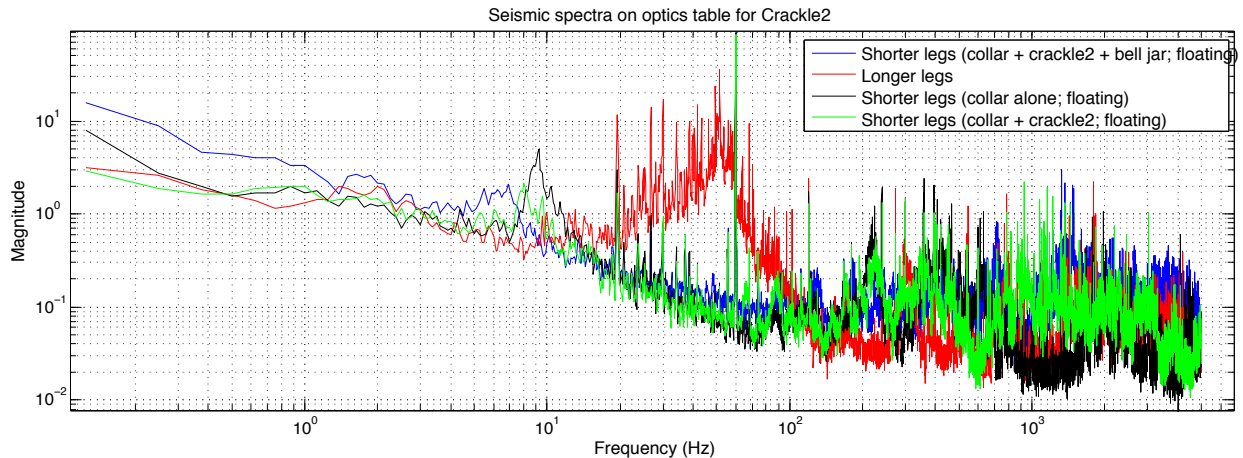
### 3.4.3 Changing legs of the optics table

As mentioned above, the legs of the (horizontal) optics table were changed from the longer, rigid legs to new shorter ones which can "float" the table on pumping air. The idea was to provide better isolation from seismic motion, especially for vertical displacement - which is of much interest. Acoustic noise can be reduced, too, significantly by installing a vacuum system. The vacuum system is yet to be fully commissioned, as of today. In this subsection, I describe the effects of the change in table legs alone.

#### Effect on motion spectra

##### Seismometer spectra

The first thing that we observed was how the motion of the optics table itself has changed at various frequencies, and how the resonance peak has shifted thanks to the new legs. For this purpose, we used a seismometer (a PZT accelerometer coupled with a voltage preamplifier (SR560)).



**Figure 21: Seismic spectra for vertical displacement on horizontal table for Crackle2.** Of primary importance here are the traces in red and blue, where the former case corresponds to the older legs, and the latter corresponds to the case with new legs and the vacuum chamber. Notice the difference in noise levels in the 10s of Hz region in both the cases.

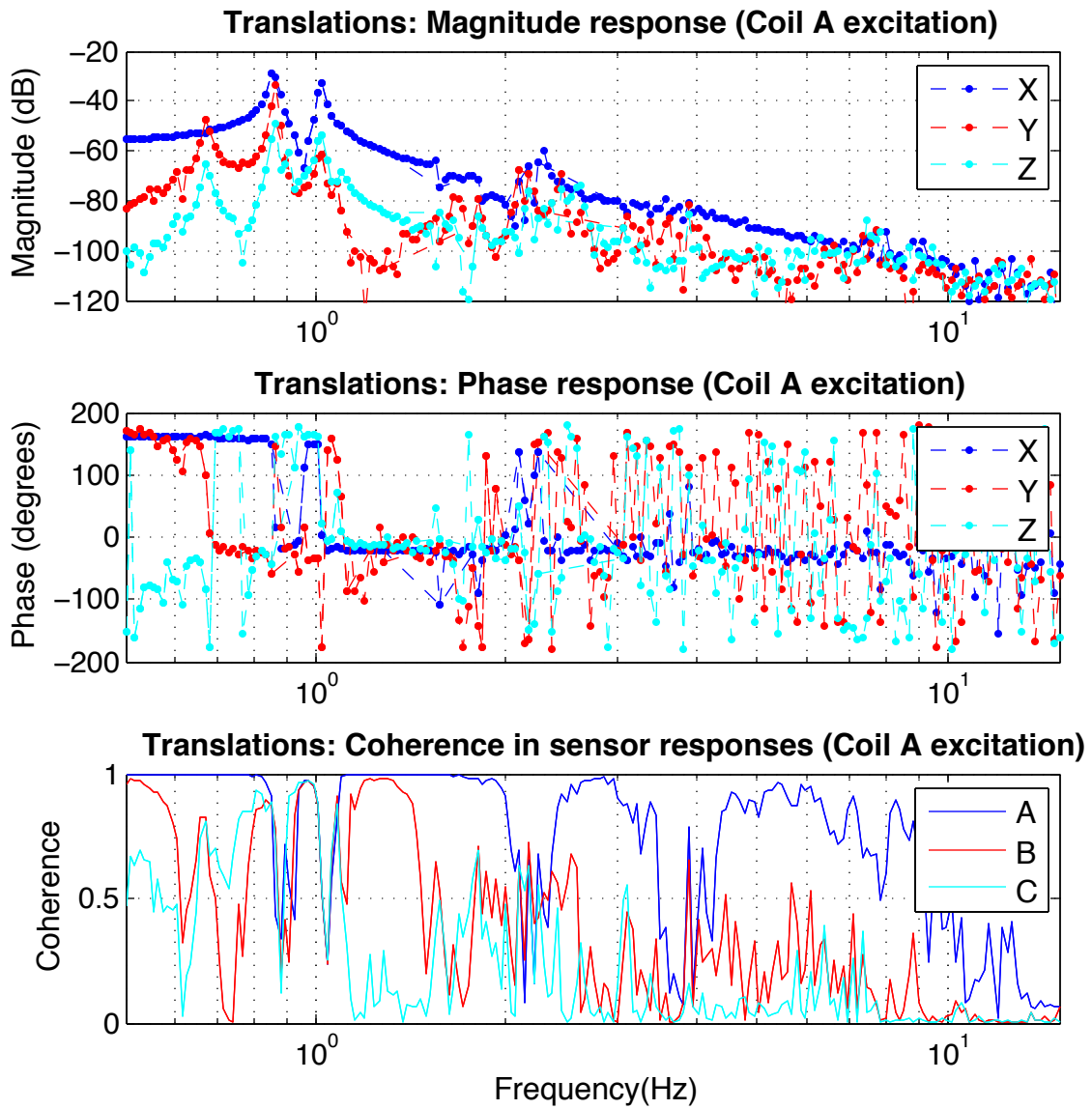
Clearly, the huge bump in the 20-100Hz region is now gone, upon changing the legs. The new table, which can also be "floated" in air, has certainly improved seismic isolation for the whole setup. It is now much more likely to get less noisy transfer function measurements. With this motivation, I started another round of transfer function measurements.

### 3.4.4 Transfer function measurements - again

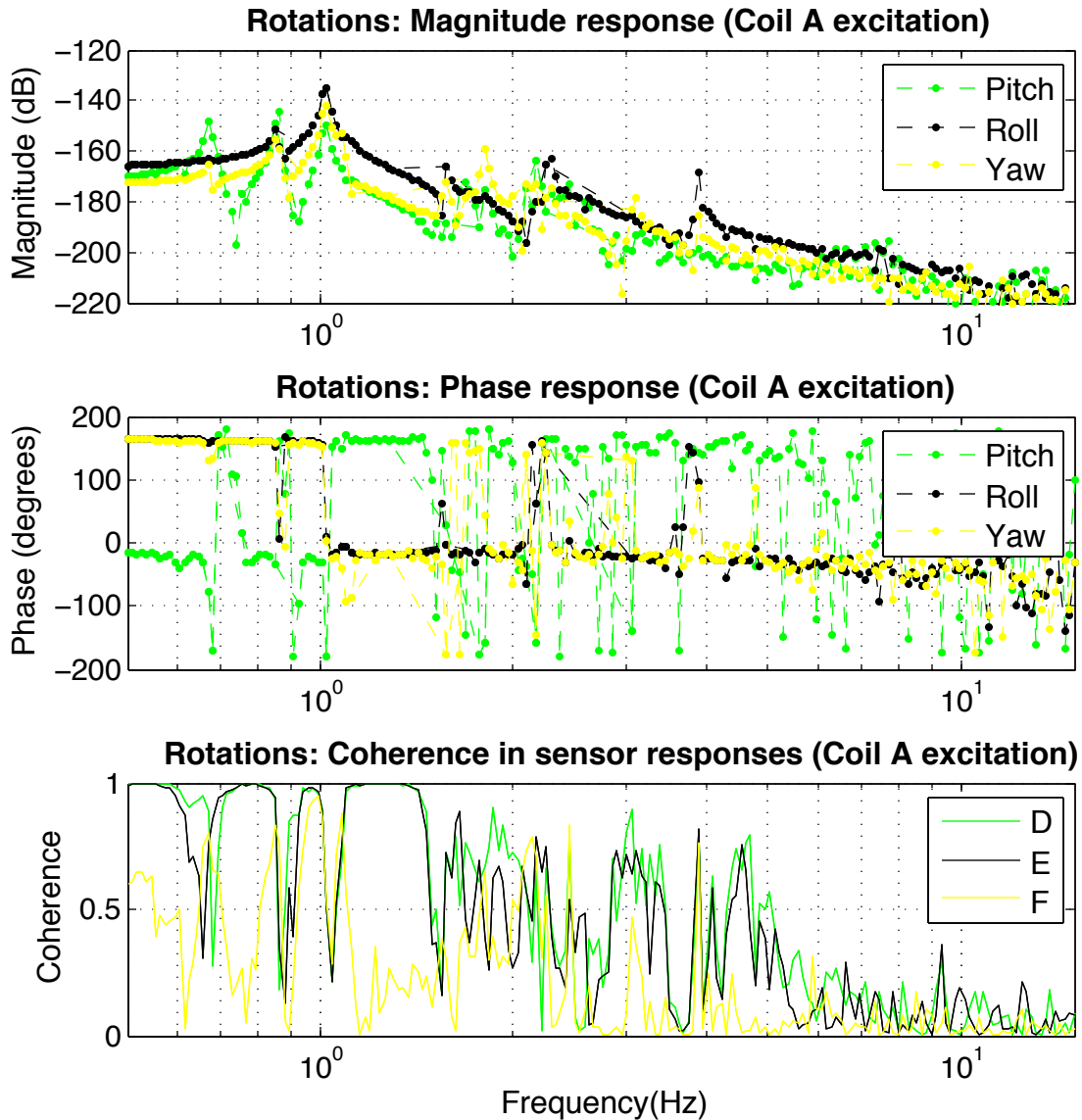
**After shifting to the newer legs,** I performed transfer function measurements again. This time, I could clearly observe two peaks: something that one would expect in a double suspension system and was not seen in earlier measurements. The reason why this wasn't observed earlier, as said above, is not very clearly known, and it is not of interest at this stage either.

Since the setup is not fully in shape yet for a series of measurements and has been undergoing a changes, modifications and realignment, I haven't recorded all transfer functions of interest yet.





**Figure 22: Bode plots for translational d.o.f. for coil A excitation (after shifting to shorter legs)**



**Figure 23: Bode plots for rotational d.o.f. for coil A excitation (after shifting to shorter legs)**

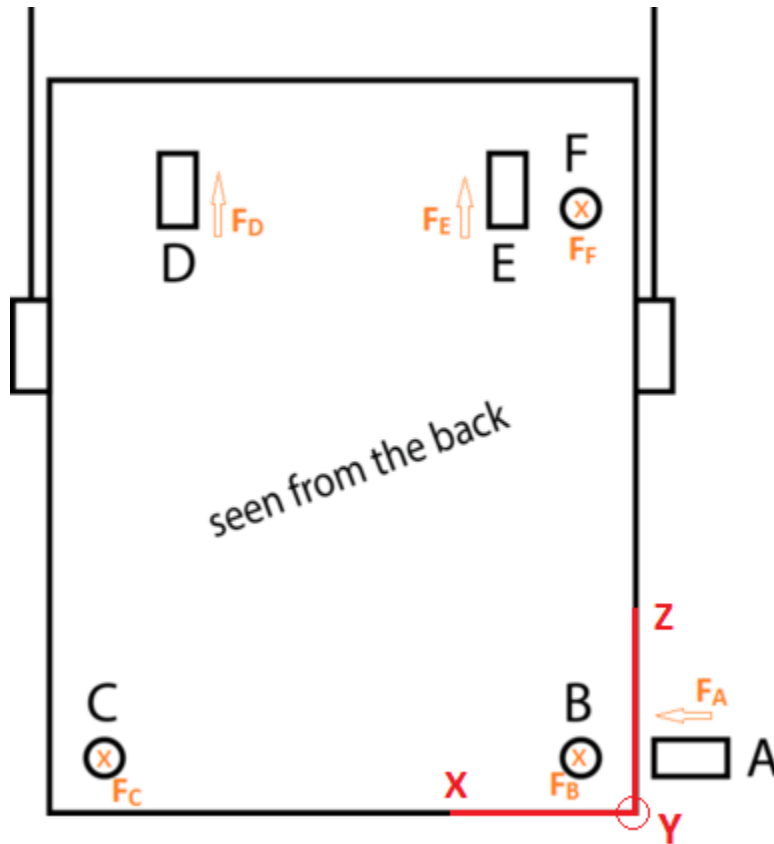
Figures 21 and 22 show bode plots of transfer functions for coil A excitation to physical d.o.f. motion. However, the coherence shown is still that of coil-A-excitation to sensor-reading, for sensors A through F. In other words, I haven't done any transformation on the coherence values.

One can see that coherences in responses of sensors other than A are a little poor, and this can be attributed to the fact that there is less coupling between orthogonal motions. But then, it does make sense: when A is excited, I expect to see some translation along X-axis along with some roll; and roll is measured by D and E. It is this coupling that gives better coherence in D and E measurements, as can be clearly seen in figure 23.

### 3.4.5 Driving motion along physical d.o.f.: Driving matrix

Using the sensing matrix in (6), it is possible to sense motion along physical d.o.f. So far, driving has been only along each OSEM axis. We would, however, like to drive the suspended breadboard along physical d.o.f.: these are the d.o.f.'s that will be relevant in the control system I hope to establish.

For this purpose, I have constructed a transformation equation, simply based on Newton's 2nd law along various d.o.f.'s.



**Figure 24: Scheme of OSEM's with force sign convention shown**

Figure 24 shows OSEM scheme along with the sign convention of driving force for each OSEM. The convention is that **for a positive drive, the magnet moves OUT of the OSEM.**

One can, by looking at figure 24, easily write down the following equations:

$$F_x = F_A \quad (7)$$

$$F_y = -F_B - F_C - F_F \quad (8)$$

$$F_z = F_D + F_E \quad (9)$$

Similarly for the net torque:

$$\tau_{net} = \sum_{i=A}^F \vec{r}_i \times \vec{F}_i \quad (10)$$

where  $\vec{r}_i$  and  $\vec{F}_i$  are position and force vectors of OSEM 'i'.

Equation (10) can be expanded as below:

$$\tau_{net} = \vec{r}_A \times (F_A \hat{e}_x) + \vec{r}_B \times (F_B (-\hat{e}_y)) + \vec{r}_C \times (F_C (-\hat{e}_y)) + \vec{r}_D \times (F_D \hat{e}_z) + \vec{r}_E \times (F_E \hat{e}_z) + \vec{r}_F \times (F_F (-\hat{e}_y)) \quad (11)$$

$F_i$  corresponds to force by each OSEM; the negative sign in some terms is because of the orientation of the OSEM's. (For example, positive force – which we defined as when the magnet moves out – along B would mean a displacement in negative y direction.)

The x, y, z components of torque can be separated and a **matrix transformation equation** can be written from linear equations 6 through 10 as follows:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & z_{cm,B} & z_{cm,C} & y_{cm,D} & y_{cm,E} & z_{cm,F} \\ z_{cm,A} & 0 & 0 & -x_{cm,D} & -x_{cm,E} & 0 \\ -y_{cm,A} & -x_{cm,B} & -x_{cm,C} & 0 & 0 & -x_{cm,F} \end{pmatrix} \begin{pmatrix} F_A \\ F_B \\ F_C \\ F_D \\ F_E \\ F_F \end{pmatrix} \quad (12)$$

Recall:  $x_{cm,A}$ , for instance, is the x-coordinate of OSEM A in the CM frame.

On plugging in values of the necessary coordinates from table 2 **in meter**, the equation can be written down as:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -0.1606 & -0.1636 & 0.0527 & 0.0527 & 0.17391 \\ -0.1606 & 0 & 0 & -0.11182 & 0.12018 & 0 \\ -0.0270 & 0.18018 & -0.17182 & 0 & 0 & 0.18018 \end{pmatrix} \begin{pmatrix} F_A \\ F_B \\ F_C \\ F_D \\ F_E \\ F_F \end{pmatrix} \quad (13)$$

Therefore, the Driving Matrix equation is:

$$\begin{pmatrix} F_A \\ F_B \\ F_C \\ F_D \\ F_E \\ F_F \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.0774 & -1 & -1 & 0 & 0 & -1 \\ -0.0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -0.1606 & -0.1636 & 0.0527 & 0.0527 & 0.17391 \\ -0.1606 & 0 & 0 & -0.11182 & 0.12018 & 0 \\ -0.0270 & 0.18018 & -0.17182 & 0 & 0 & 0.18018 \end{pmatrix} \begin{pmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} \quad (14)$$

Here, forces are in  $\mu\text{N}$  and Torque in  $\mu\text{Nm}$ . **I have not yet started using the driving matrix in transfer function measurements, I will start doing so only in the next set of measurements.**

### 3.4.6 Automation of (multiple) simultaneous transfer function measurements

All of the above discussed transfer function plots were arrived at after running measurements in series: that is, one coil excitation after another. Each measurement took about 6-8hrs, and 6 such measurements meant a week's time. It is then required to bring this time down by adjusting the parameters of DTT measurements, such as number of averages, number of cycles of measurements etc. I will not go into more details about those here, refer to Appendix A.

One can then say, it would be even more efficient if all measurements could be taken in parallel in, say, a night.

Turns out that, in principle, it is possible with our system for the simple reason that it is a linear system. Why is it a linear system? Because Newton's laws – which govern its behavior – are linear.

**As long as the system is not excited by the same frequency through different coils at the same instant of time**, due to linearity, the measurements should be as good as they were taken independently. The difficult part, however, is to implement it.

It is certainly not possible for one to do it manually each time by opening 6 parallel sessions of DTT. Automation is needed. To **automate** parallel sessions of transfer functions, the following are some possible methods:

1. Use the Diagnostic Tool ("diag") through the workstation computer along with a python script
2. Use Python and MATLAB to run DTT measurements

Both of these methods have been tried out before at LLO/LHO. I have initially attempted to work on the second solution. However, I faced some issues with getting the Python-MATLAB interface working. (To be specific, the problem was with Python XML parsing and its interface with MATLAB.) The problem didn't even make much sense, and so I switched to the first method.

In this method, a DTT template for the measurement of interest is first prepared and saved as an XML file. The desired parameters are set; 'Start time' set as 'Now'.

A measurement based on this DTT template can be run by using a short 'diag' script. The script itself is simple, and goes as follows:

```
open
restore PATH-TO-DIRECTORY/IN_FILE_NAME.xml
run -w
save PATH-TO-DIRECTORY/OUT_FILE_NAME.xml
exit
```

Here, **path to directory and file names are to be set appropriately**. Appendix B explains the codes in more detail.

The above script is for the Diagnostic Tools "diag". In order to automate this for all 6 coil excitations, I have written a short Python script based on the 'subprocess' module. In short, the whole process can be summarized as:

1. Input DTT templates are saved in directory
2. 'diag' scripts are prepared appropriately and saved in the same directory
3. A python script to execute all the 6 'diag' scripts in parallel terminal sessions is then executed

### 3.4.7 Fitting multiple transfer functions with common set of poles

The way the collected transfer function data is used is in the zpk (zero-pole- gain) form. It then becomes necessary to fit these transfer functions; and not just fit them, but to fit them with common poles. Why common poles? Because it is the same system, and we expect resonances (peaks) seen in different measurements to coincide.

For this, I have used the **vectfit3** function in MATLAB. [3][4][5] The function works for both single transfer function as well as a vector of transfer functions. However, given the way I have been using coherence data of transfer function measurements, it was not straightforward.

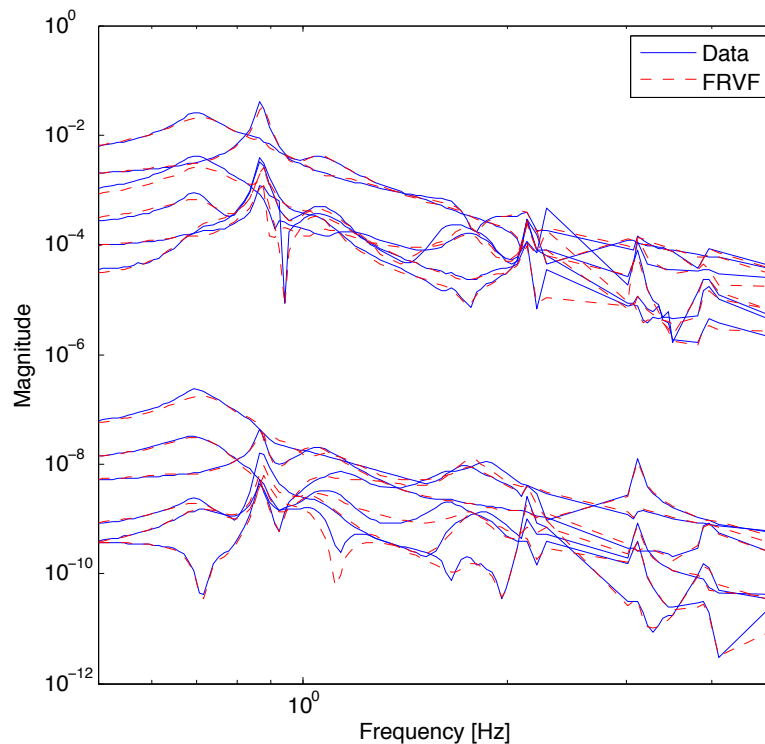
The problem is subtle: For the vectfit3 function to work, data for each transfer function in the transfer function array must have the same length. In general, I measure 36 transfer functions as groups of 6 each, 1 group for each coil excitation. Each of these groups have a set of finite points for which the coherence of all the 6 measurements in the group is above a set threshold (say 0.7), and the points of which all coherence values are above the threshold are different for each group. Put in simple terms, the frequency axes were different for each group. Clearly, simply inputting this to the vectfit3 function won't work.

How I solved this was to identify points on the frequency axis, for each group, where all coherences were above the set threshold; and then find the intersection of such collected points for all groups. This set the common frequency axis for the measurements. I extracted transfer function values corresponding to these and then made use of the `vectfit3` function.

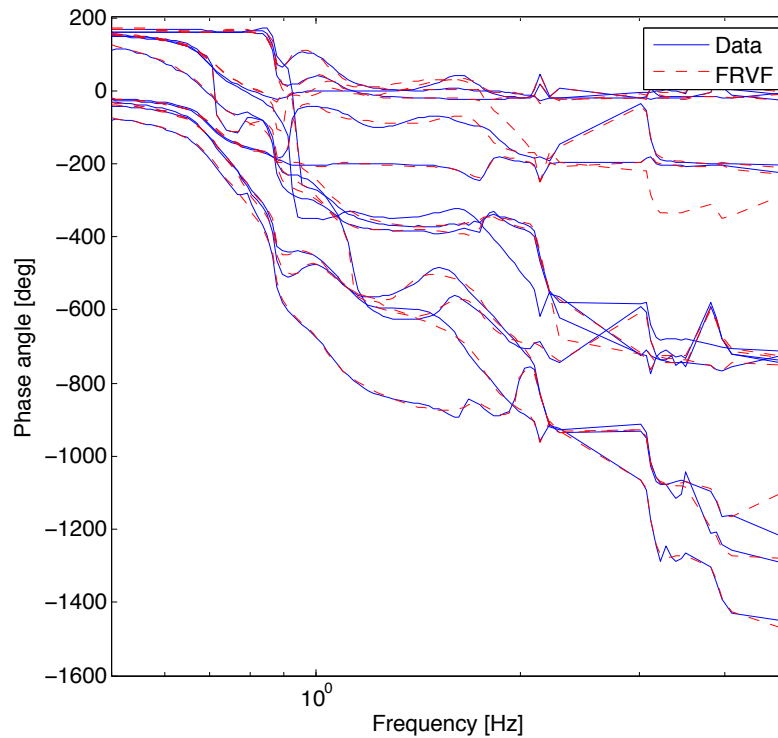
A more elegant way, where the problem wouldn't even arise, is to directly measure and use the coherence of coil excitation to physical d.o.f.; and this is what I would be implementing in further measurements.

In either case, the MATLAB code for fitting multiple TF's can be a bit messy, because it has a number of parameters to be set. I have written a wrapper function for convenient use. See Appendix C for more.

For the purpose of illustration, I have shown in figure 25 an example fit, where I tried to fit 12 transfer functions with a common set of poles. Though the order of the fit was on the higher side: 16.



**Figure 25(a): Simultaneous vector fitting of 12 transfer functions: Magnitude plot**



**Figure 25(b): Simultaneous vector fitting of 12 transfer functions: Phase plot**

The exact locations of the poles and other details are not of much interest to us here. These plots were for a purely representational purpose.

### 3.4.8 Comparison of experimental data with analytical predictions

One other purpose of fitting poles and zeros to experimentally obtained data is to compare against analytical predictions. Having an analytical model - one based on Newton's Laws - often helps in ensuring the credibility of the experimental results obtained; this is something that's true for any experiment in Physics.

I have spent about a couple of weeks working on analytical modeling of Crackle2. I started off with an existing model (which I will refer to as Model 1 from now on) built on Mathematica, and obtained some results. However, the results weren't satisfactory. I then started building a new model from scratch. (I will refer to the model as 'Model 2' from now on, in this paper.) Here, I discuss the results I have obtained in Model 1 and talk a bit about the progress I have made in Model 2.

#### 1. Model 1

This one was developed about an year ago and was used to compute a few transfer functions



of interest in the past. I will not talk about the full details of the Mathematica code here. Refer to Appendix D for the same.

In summary, the Mathematica simulation (as detailed in Appendix D) saves state-space matrices  $A$ ,  $B$  and  $F$ , in a format that MATLAB can understand, on running a 'ToMatlab.m' script.  $A$  represents dynamics,  $B$  is the input coupling matrix and  $F$  represents external disturbances. These matrices can be used to compute the required transfer functions.

In the current case of interest, the state equation can be written as:

$$\ddot{\vec{x}} = A\vec{x} + F\vec{x}_0 + B\vec{\tau} \quad (15)$$

where  $\vec{x}$  is the state vector,  $\vec{x}_0$  corresponds to exogenous inputs from the environment and  $\vec{\tau}$  corresponds to the driving vector. The use of the letter 'x' in  $\vec{x}_0$  suggests that these exogenous inputs are state vectors, or so they may be regarded;  $\vec{x}_0$  corresponds to the *state of the environment*. (This is consistent with the notation in [6].)

In Laplace domain,

$$s^2 I\vec{x}(s) = A\vec{x}(s) + F\vec{x}_0(s) + B\vec{\tau}(s) \quad (16)$$

This gives,

$$\vec{x}(s)/\tau(s) = (s^2 I - A)^{-1} B \quad (17)$$

$\vec{x}(s)/\tau(s)$  is a  $12 \times 12$  matrix of transfer functions. ( $12 \times 12$  because the Crackle2 mechanical system has 12 degrees of freedom: 6 in the intermediate mass stage and 6 in the suspended breadboard (lower mass, in aLIGO jargon) stage. Of interest to me is the last 6, and so I have made use of the corresponding  $6 \times 6$  sub-matrix to compare with experimental data.

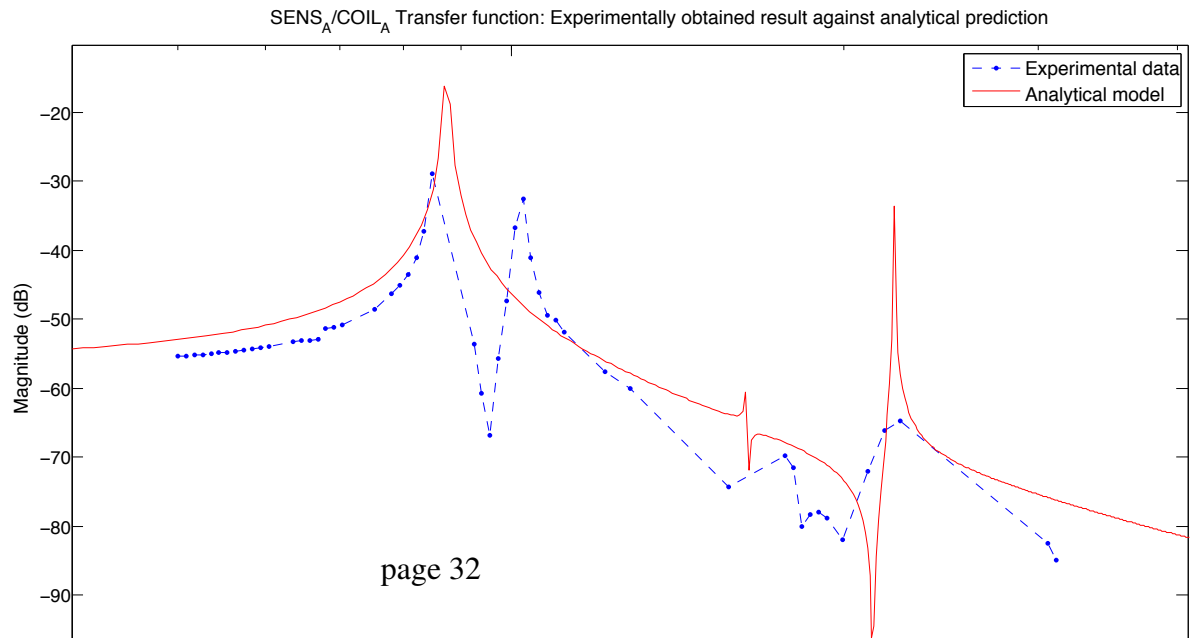
Like I said before, the output of the Mathematica simulation is a set of three matrices **in symbolic representation**, and in order to be able to plot transfer functions, appropriate parameters must be plugged in. Table 4 talks about some of the parameters that I have used in the simulation. I have, in fact, played around with these values and varied them in a range where they still made physical sense. However, as will be seen in the result plot below, the results are not satisfactory.

Where necessary and possible, I have given enough description of what each parameter means. Refer to Appendix D for further details.

Figure 25 shows the analytical prediction of Coil A excitation to Sensing along X transfer function against experimentally obtained data.

Table 4: Parameters used in Model 1

Parameter	Description	Value
M1	Mass of intermediate mass stage	20 kg
M2	Mass of lower mass stage	20kg
$I_{1,xx}$	Moment of Inertia of intermediate mass stage about x-axis	0.406 kgm <sup>2</sup>
$I_{1,yy}$	Moment of Inertia of intermediate mass stage about y-axis	0.399 kgm <sup>2</sup>
$I_{1,zz}$	Moment of Inertia of intermediate mass stage about z-axis	0.643 kgm <sup>2</sup>
$I_{2,xx}$	Moment of Inertia of lower mass stage about x-axis	0.338 kgm <sup>2</sup>
$I_{2,yy}$	Moment of Inertia of lower mass stage about y-axis	0.604 kgm <sup>2</sup>
$I_{2,zz}$	Moment of Inertia of lower mass stage about z-axis	0.267 kgm <sup>2</sup>
$L_1, L_4$	Lengths of suspension wires joining the intermediate mass stage to the cage	0.16 m
$L_2, L_3$	Lengths of suspension wires joining the intermediate mass stage to the cage	0.16 m
$L_5, L_6$	Lengths of suspension wires joining the lower mass stage to the intermediate mass stage	0.20 m
$k_1, k_2, k_3, k_4, k_5, k_6$	Spring constants corresponding to all suspension blades	1500 Nm <sup>-1</sup>
$R_{1,x}, R_{1,y}$		0.34 m
$d$		0.07 m
$g$	Acceleration due to gravity	9.8 ms <sup>-2</sup>



**Figure 25: Analytical prediction vs experimental data for Coil A excitation to Sensing along X transfer function**

It can be clearly seen that the two plots don't fit well: the position of the first zero and second pole do not match at all. The shape, too, is not very similar. What's more, the analytical result does not show the second pole as expected; it only shows it as a minor local peak. Clearly, this needs more working: it was concluded that Model 1 is "insufficient" in explaining full motion and all modes of the system. Therefore, I have started working on building another model of Crackle2 from scratch, this time on a Mathematica-based platform that was developed by personnel who have worked at Advanced LIGO and KAGRA.

## 2. Model 2

This platform was built specially to simulate suspension systems in gravitational wave detectors. It is convenient to add mass stages, define suspension spring and string parameters, and compute the necessary transfer functions. Work on this part is still underway, and so far, I haven't produced any results. Only the next (and final) report will contain results corresponding to this part.

## 4 Timeline

### 4.1 How am I doing on time?

Things didn't go as (optimistically) planned while I made my first progress report. A lot of time went into getting the setup up and running each time there was a modification. While this did contribute considerably to my learning and gave me an opportunity to know the Crackle2 setup in and out, it did eat into some time that I had earlier planned for other things. Roughly speaking, I am lagging behind the earlier planned timeline by 3-5 days. Though it does look a little tough now to catch up and finish things as I wanted to earlier, I think it is still possible, with some extra effort, to at least implement the control system before leaving.

### 4.2 Up next

Given the way Crackle2 is going at the moment, we expect to pump down vacuum in about a couple of weeks from now. Till then, while other work is on, I will attempt to finish design (and perhaps implementation) of the suspension damping system. I have about 3 weeks left, which includes Caltech Gravitational Wave Astrophysics School (CGWAS) and a field trip to the LIGO Livingston Observatory (LLO). On taking these into account, I think here is how I will spend my time.

Table 5: Timeline

Week	Plan
1 - 5 July	Finish acquisition of transfer function data; Attempt to produce first transfer function plots from analytical model
6 - 10 July	CGWAS 2015; Finish comparison of analytical model results with experimental data; Groundwork towards implementing feedback control
12 - 18 July	Design and implement digital feedback control; Final presentation
20 - 22 July	LLO Visit

## References

- [1] Xiaoyue Ni, E. Quintero, G. Vajente, "Proposal for an upgrade of the Crackle experiment", LIGO-T1400407-v1 (2015)
- [2] E. Quintero, E. Gustafson, R. Adhikari, "Experiment to investigate crackling noise in maraging steel blade springs", LIGO-T1300465-v2 (2013)
- [3] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain response by Vector Fitting", *IEEE Trans. Power Delivery*, vol. 14, no. 3, pp. 1052-1061, July 1999.
- [4] B. Gustavsen, "Improving the pole relocating properties of vector fitting", *IEEE Trans. Power Delivery*, vol. 21, no. 3, pp. 1587-1592, July 2006.
- [5] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of Multiport Systems Using a Fast Implementation of the Vector Fitting Method", *IEEE Microwave and Wireless Components Letters*, vol. 18, no. 6, pp. 383-385, July 2008.
- [6] Bernard Friedland, "Control System Design: An Introduction to State-Space Methods", *Dover Publications Inc.*, 2005.