

LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -

=====

LIGO SCIENTIFIC COLLABORATION

Technical Note	LIGO-T1400587-v1-	2014/09/16
Deriving a MATLAB model for the quad fiber violin modes		
Brett Shapiro		

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 NW22-295
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.org/>

1 Introduction

This note derives a state space model for the fused silica fiber violin modes for a single fiber vibrating along the axis of the interferometer (longitudinal DOF). It then shows how to connect this model to an existing quadruple suspension state space model. This is for use with the MATLAB model.

It is hoped that the derivations throughout this document are complete and sufficient to understand how the fiber dynamics have been modeled. For brevity, some algebraic steps have been left out, but the information is complete enough that the reader could check these if they wished.

Section 2 lists the notation this document uses, Section 3 derives the state space matrices for the fiber one by one, and Section 4 shows how to connect this fiber model to the suspension. Section 5 lists the MATLAB commands for building a quad model with violin modes.

2 Notation

- Bold capital letters denote matrices: \mathbf{M} .
- Bold lower case letters denote column vectors: \mathbf{x} .
- Non-bold letters, lower case or upper case, denote scalars: m, L .

3 Derivation of the fiber state space dynamic equations

It is assumed the fiber vibrates along the x axis defined in Fig. 1. In MATLAB it is convenient to model this fiber in a state space form. This form requires breaking the model into discrete states. There are two choices here illustrated in Fig. 1, a spatial discretization or a modal discretization. The modal discretization is convenient because each mode is decoupled from the others and because the frequency of each mode can be represented exactly with Eq. (11). The modal format also minimizes the number of states required because each mode only requires two, one for displacement and one for velocity. The spacial discretization is convenient because the deflections of the fiber endpoints can be represented exactly. However, the mode frequency will have error, converging to zero as the number of states goes to infinite.

We choose to follow the modal format of the state space equations (1) and (2).

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix}_{2N \times 1} = \mathbf{A}_{2N \times 2N} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \mathbf{B}_{2N \times 2} \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (1)$$

$$\mathbf{f}_{2 \times 1} = \mathbf{C}_{2 \times 2N} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \mathbf{D}_{2 \times 2} \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (2)$$

\mathbf{q} represents the modal displacements of the fiber. It is an N by 1 column vector, where N is the user chosen number of modes to include in the model. u_0 and u_L are the fiber

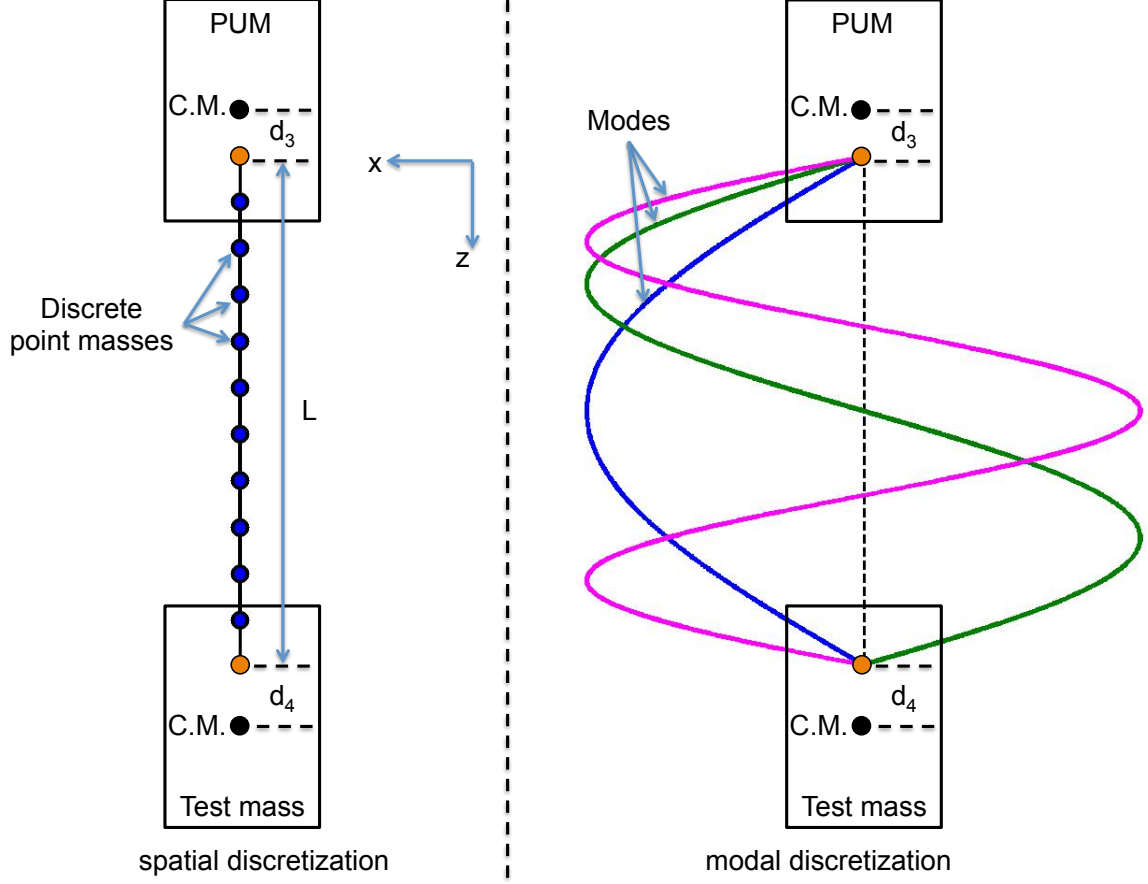


Figure 1: Two models for a silica fiber between the penultimate mass (PUM) and test mass. The fiber is of length L between the orange endpoints. Left: spatially discretized fiber model. The fiber is modeled as a set of N equal point masses spaced evenly along the length of the fiber, under tension T . Right: modally discretized fiber model. The fiber is modeled as a set of N vibrational modes, under tension T .

endpoint displacements in the x direction at the PUM and test mass respectively. \mathbf{f} is the 2 by 1 column vector of reaction forces (along x) that the fiber exerts on suspension at its endpoints. The following derivation will proceed to solve for the \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} matrices.

3.1 A matrix and mode frequencies and shapes

We begin by modeling the fiber as a taut continuous string in the spatial coordinate system

$$\rho \ddot{u} - T \frac{\partial^2 u}{\partial x^2} = g[u_0, gL] \quad (3)$$

where ρ is the mass per unit length of the fiber, T is the tension of the fiber and $u = u[t, z]$ is the transverse deflection of the fiber along the x axis. g is the function that transforms the endpoint deflections u_0 and u_L to forces on the fiber. We will consider these endpoint deflections in Section 3.2.

For the homogeneous equation

$$\rho\ddot{u} - T\frac{\partial^2 u}{\partial x^2} = 0 \quad (4)$$

Assume a given vibrational mode has a solution of

$$u[t, z] = a \sin(\omega t) \sin(\lambda z) \quad (5)$$

where ω is the mode frequency in units of rad/s, λ is the spacial mode frequency (inverse of the wavelength) in units of rad/m, and a is the amplitude of the mode in m.

The boundary conditions are

$$u[t, 0] = u[t, L] = 0 \quad (6)$$

Plugging the second boundary condition into the assumed solution solves for λ

$$\sin(\lambda L) = 0 \quad (7)$$

$$\lambda = \frac{n\pi}{L}, \quad n = 1, 2, 3, \dots etc \quad (8)$$

We can now relate the mode frequency ω to λ by plugging the solution into the equation of motion (4). This gives us

$$-\rho\omega^2 a \sin(\omega t) \sin(\lambda z) + T\lambda^2 a \sin(\omega t) \sin(\lambda z) = 0 \quad (9)$$

Simplifying by canceling common terms

$$-\rho\omega^2 + T\lambda^2 = 0 \quad (10)$$

$$\omega = \lambda \sqrt{\frac{T}{\rho}} = \frac{n\pi}{L} \sqrt{\frac{T}{\rho}} \quad (11)$$

Since there are infinite vibrational modes, the full solution will be a sum of all the modes.

$$u[t, z] = \sum_{n=1}^{\infty} a_n \sin(\omega_n t) \sin(\lambda_n z) \quad (12)$$

Here a_n is the amplitude of mode n , and $\sin(\lambda_n z)$ is the mode shape of mode n . Or to simplify the notation,

$$u[t, z] = \sum_{n=1}^{\infty} q_n \sin(\lambda_n z) \quad (13)$$

where

$$q_n = a_n \sin(\omega_n t) \quad (14)$$

q_n is the displacement of mode n , and the mode shape $\sin(\lambda_n z)$ can be thought of as the transformation from mode n to the spatial coordinates of the fiber. Since each mode of vibration is independent, we can write the equation of motion for each mode as if it were a single DOF oscillator.

$$m_{m,n}\ddot{q}_n + k_{m,n}q_n = 0 \quad (15)$$

where $m_{m,n}$ is the modal mass for mode n , $k_{m,n}$ is the modal stiffness. We can derive the \mathbf{A} matrix from this equation. For this matrix, we do not need to know the modal mass or stiffness because we can simply divide out the modal mass, resulting in

$$\ddot{q}_n + \omega_n^2 q_n = 0 \quad (16)$$

where

$$\omega_n = \sqrt{\frac{k_{m,n}}{m_{m,n}}} \quad (17)$$

which has already been defined in Eq. (11).

We can then write the \mathbf{A} matrix as

$$\mathbf{A}_{2N \times 2N} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \\ -\mathbf{\Omega}^2 & \mathbf{0}_{N \times N} \end{bmatrix} \quad (18)$$

where $\mathbf{\Omega}^2$ is the N by N diagonal matrix listing ω_1^2 through ω_N^2 .

3.2 B matrix for accepting endpoint displacements as inputs to the fiber

Since the fiber endpoint displacements are along the x axis, we must transform them from this spatial coordinate to the modal coordinates. To this end, it is convenient to return to the spatial coordinates and discretize Eq. (3) with the left side of Fig. 1 in mind. For good agreement with the modal form, we start out considering the limits where $N \rightarrow \infty$, but we'll see eventually that once in the modal form N does not need to be infinitely large.

In the discrete spacial form, Eq. (3) becomes

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{K}_e \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (19)$$

The mass matrix \mathbf{M} is diagonal with each diagonal element representing a point of mass m along the fiber length.

$$\mathbf{M} = m\mathbf{I}_{N \times N} \quad (20)$$

where

$$m = \frac{\rho L}{N} \quad (21)$$

The stiffness matrix \mathbf{K} is a symmetric matrix where k represents the stiffness between adjacent point masses. This matrix can be derived by inspection of Fig. 1 by displacing each mass by 1 unit in turn and observing the forces required to hold that configuration (think $\mathbf{f} = \mathbf{K}\mathbf{u}$, where \mathbf{u} is all zeros except for a single element). For example, a unit displacement on mass 1 (at the top) requires $2k$ of force. The force required to hold the mass below in its zero position is $-k$, the reaction force exerted on mass 1. The third mass and those below require no force because the second has not moved. This gives you column 1. Repeat for columns 2 to N .

$$\mathbf{K} = k \begin{bmatrix} 2 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ 0 & 0 & -1 & 2 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}_{N \times N} \quad (22)$$

where

$$k = N \frac{T}{L} \quad (23)$$

The endpoint stiffness matrix \mathbf{K}_e is derived in a similar way by displacing the endpoints and observing the forces exerted on the point masses.

$$\mathbf{K}_e = k \begin{bmatrix} 1 & 0 \\ 0 & \dots \\ \dots & 0 \\ 0 & 1 \end{bmatrix}_{N \times 2} \quad (24)$$

The conversion to the modal form is then realized through the transformation

$$\mathbf{u} = \mathbf{\Phi} \hat{\mathbf{q}} \quad (25)$$

where $\hat{\mathbf{q}}$ is the modal deflection of the spatially discretized fiber and $\mathbf{\Phi}$ is the matrix of eigenvectors or mode shapes.

$$\mathbf{\Phi} = [\phi_1 \quad \phi_2 \quad \phi_3 \quad \dots \quad \phi_N]_{N \times N} \quad (26)$$

Note, as $N \rightarrow \infty$ then $\hat{\mathbf{q}} \rightarrow \mathbf{q}$ and each eigenvector $\phi_n \rightarrow \sin(\lambda_n z)$.

Plugging Eq. (25) into Eq. (19) we get

$$\mathbf{M} \mathbf{\Phi} \ddot{\hat{\mathbf{q}}} + \mathbf{K} \mathbf{\Phi} \hat{\mathbf{q}} = \mathbf{K}_e \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (27)$$

Then multiply on the left by the matrix transpose of the eigenvectors

$$\mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} \ddot{\hat{\mathbf{q}}} + \mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} \hat{\mathbf{q}} = \mathbf{\Phi}^T \mathbf{K}_e \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (28)$$

The notation is then simplified to modal form

$$\mathbf{M}_m \ddot{\hat{\mathbf{q}}} + \mathbf{K}_m \hat{\mathbf{q}} = \mathbf{\Phi}^T \mathbf{K}_e \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (29)$$

where \mathbf{M}_m is the diagonal matrix of modal masses and \mathbf{K}_m is the diagonal matrix of modal stiffnesses,

$$\mathbf{M}_m = \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} \quad (30)$$

$$\mathbf{K}_m = \mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} \quad (31)$$

To work towards a state space form that is compatible with the \mathbf{A} matrix from Section 3.1, we multiply on the left by \mathbf{M}_m^{-1} .

$$\ddot{\hat{\mathbf{q}}} + \hat{\mathbf{\Omega}}^2 \hat{\mathbf{q}} = \mathbf{M}_m^{-1} \mathbf{\Phi}^T \mathbf{K}_e \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (32)$$

where,

$$\hat{\mathbf{\Omega}}^2 = \mathbf{M}_m^{-1} \mathbf{K}_m \quad (33)$$

is a diagonal matrix with the squared mode frequencies (in rad/s) of this spatially discretized system. Note, as $N \rightarrow \infty$ then $\hat{\Omega}^2 \rightarrow \Omega^2$.

Next we derive the matrix product $\mathbf{M}_m^{-1} \Phi^T \mathbf{K}_e$. For \mathbf{M}_m , it turns out that all modal masses are the same. In the limit of a continuous system, the modal mass m_n is given by

$$m_m = \int_0^L \rho \sin^2(\lambda_n z) dz \quad (34)$$

$$m_m = \frac{\rho}{2} \int_0^L \left[1 - \cos\left(\frac{2n\pi}{L} z\right) \right] dz \quad (35)$$

$$m_m = \frac{1}{2} \rho L - \frac{1}{2} \rho L \int_0^L \cos\left(\frac{2n\pi}{L} z\right) dz \quad (36)$$

The integrand in the second term is harmonic, so the integral goes to zero. Thus,

$$m_m = \frac{1}{2} \rho L \quad (37)$$

Note, if you derive the modal mass from the discrete equation $\mathbf{M}_m = \Phi^T \mathbf{M} \Phi$, you will see that you get a value equal to $\frac{1}{N} \rho L$, smaller by a factor of $0.5N$ (hint $\Phi^T \Phi = \mathbf{I}$). However, for N larger than just a few, \mathbf{K}_m is also too small by the same factor (MATLAB observations). All we need is to scale both sides of (19) by $0.5N$ to make it consistent with the continuous string results. This has no impact on our derivation, so we ignore it moving forward.

The problem with Φ is that we do not know exactly what it is because it depends on whatever value was chosen for N . However, since \mathbf{K}_e is only nonzero in the top left and bottom right corners, we only need to approximate the first and last elements for each eigenvector.

Recall that for large N , ϕ_n is a good approximation for the true mode shape $\sin(\lambda_n z)$. The first element in eigenvector n represents the displacement of first (top) point mass for mode n . Therefore,

$$\phi_{1,n} \approx \sin\left(\lambda_n \frac{L}{N}\right) \approx \frac{n\pi}{L} \frac{L}{N} \quad (38)$$

Similarly, for the last (bottom) point mass of the fiber,

$$\phi_{N,n} \approx (-1)^{n-1} \frac{n\pi}{L} \frac{L}{N} \quad (39)$$

The alternating minus sign is there because the even numbered modes displace the last point mass in the negative x direction (see the right side of Fig. 1).

If we then multiply Eq. (38) by $k = N \frac{T}{L}$ from Eq. (23), we get

$$\phi_{1,n} k \approx T \frac{n\pi}{L} \quad (40)$$

Similarly for Eq. (39),

$$\phi_{N,n} k \approx (-1)^{n-1} T \frac{n\pi}{L} \quad (41)$$

Note that N cancels out. This fact, and that the modal mass is independent of N means that N does not need to be a large value to produce a good \mathbf{B} matrix. It need only be large enough to contain the modes of interest.

The full matrix product is then,

$$\mathbf{M}_m^{-1} \Phi^T \mathbf{K}_e = T \frac{2\pi}{\rho L^2} \begin{bmatrix} 1 & 1 \\ 2 & -2 \\ 3 & 3 \\ 4 & -4 \\ \dots & \dots \\ N & (-1)^{N-1} N \end{bmatrix}_{N \times 2} \quad (42)$$

where N now refers to the desired number of modes rather than the number of discrete point masses, since it need not be arbitrarily large.

The \mathbf{B} matrix is then

$$\mathbf{B}_{2N \times 2} = \begin{bmatrix} \mathbf{0}_{N \times 2} \\ \mathbf{M}_m^{-1} \Phi^T \mathbf{K}_e \end{bmatrix} \quad (43)$$

3.3 C matrix and the fiber reaction forces on the suspension

The \mathbf{A} matrix gives us the fiber dynamics, and the \mathbf{B} matrix receives the displacement inputs from the endpoints. The final piece we need to fully couple the fiber system to a suspension is the reaction forces of the fiber endpoints, where it will pull on the PUM and test mass. We use the \mathbf{C} matrix to compute these.

The reaction force of the fiber at an endpoint in the x direction is equal to the fiber tension T times the slope of the fiber. For the top endpoint reaction force for mode n , $f_{n,0}$

$$f_{n,0} = T q_n \frac{\partial u}{\partial z} \sin(\lambda_n z)|_{z=0} \quad (44)$$

$$f_{n,0} = T q_n \lambda_n \cos(\lambda_n z)|_{z=0} \quad (45)$$

$$f_{n,0} = \frac{n\pi}{L} T q_n \quad (46)$$

Similarly, for the mode n bottom force $f_{n,L}$

$$f_{n,L} = -T q_n \frac{\partial u}{\partial z} \sin(\lambda_n z)|_{z=L} \quad (47)$$

The minus sign is so odd modes pull the endpoints in the same direction and even modes pull in opposite directions.

$$f_{n,L} = (-1)^{n-1} \frac{n\pi}{L} T q_n \quad (48)$$

Therefore the \mathbf{C} is

$$\mathbf{C}_{2 \times 2N} = \frac{\pi}{L} T \left[\begin{array}{cccc|c} 1 & 2 & 3 & \dots & N \\ 1 & -2 & 3 & \dots & (-1)^{n-1} N \end{array} \right] \mathbf{0}_{2 \times N} \quad (49)$$

3.4 D matrix and the subtraction of DC reaction forces on the suspension

The fiber model now has all the pieces required to attach it to another system: its own internal dynamics (\mathbf{A}), inputs (\mathbf{B}), and outputs (\mathbf{C}).

However, this fiber model is intended to attach to an existing quadruple suspension model. This model already computes some of the fiber forces. If it did not, then the test mass would not be connected to the PUM. Thus, to not double count these forces, we must subtract them off from this new model, using the \mathbf{D} matrix.

The existing suspension model does not contain any dynamic behavior of the fibers (thus this note). So the only overlap between this new fiber model and the existing suspension that must be subtracted is the static reaction forces the fiber exerts on the suspension.

The calculation of these static forces is straight forward. We start with the state space form in Eq. (1) and set all time derivatives to zero.

$$\begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{0}_{N \times 1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{q} \\ \mathbf{0}_{N \times 1} \end{bmatrix} + \mathbf{B} \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (50)$$

We then solve for \mathbf{q} .

$$\begin{bmatrix} \mathbf{q} \\ \mathbf{0}_{N \times 1} \end{bmatrix} = -\mathbf{A}^{-1} \mathbf{B} \begin{bmatrix} u_0 \\ u_L \end{bmatrix} \quad (51)$$

Multiplying \mathbf{q} through the \mathbf{C} matrix gives us the static forces. We need to subtract the forces however, so we employ $-\mathbf{C}$ giving us \mathbf{D} ,

$$\mathbf{D}_{2 \times 2} = \mathbf{C} \mathbf{A}^{-1} \mathbf{B} \quad (52)$$

This completes the derivation of the fiber state space system.

4 Connecting the fiber to the suspension

Now that the fiber model has its own state space system, it must be connected to the suspension. This is done through a transformation of the fiber model inputs and outputs. For the inputs, we convert the suspension displacements to fiber endpoint displacements. For the outputs, we convert the fiber endpoint forces to forces and torques on the suspension.

Consider pitch and yaw of the PUM and test mass. Fig. 1 shows us that there is a lever arm of d_3 between the PUM center of mass and the top endpoint. Similarly, the test mass has a lever arm of d_4 . Denoting a positive pitch as counterclockwise rotation in Fig. 1, we

can transform suspension pitch and x displacements into endpoint displacements,

$$\begin{bmatrix} u_0 \\ u_L \end{bmatrix} = \begin{bmatrix} 1 & -d_3 & 0 & 0 \\ 0 & 0 & 1 & d_4 \end{bmatrix} \begin{bmatrix} x_3 \\ \theta_3 \\ x_4 \\ \theta_4 \end{bmatrix} \quad (53)$$

where x_3 and θ_3 are the x and pitch displacements of the PUM respectively, and x_4 and θ_4 are the x and pitch displacements of the test mass respectively.

The fiber endpoint reaction forces f_1 and f_2 are then transformed to forces and torques on the suspension,

$$\begin{bmatrix} F_3 \\ \tau_3 \\ F_4 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -d_3 & 0 \\ 0 & 1 \\ 0 & d_4 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (54)$$

where F_3 and τ_3 are the force and torque on the PUM respectively, and F_4 and τ_4 are the force and torque on the test mass respectively.

5 Building the model with violin modes in MATLAB

The relevant scripts, as of the writing of this doc, live in the svn at

...SusSVN/sus/trunk/QUAD/Common/MatlabTools/QuadModel_Production

The MATLAB script that includes all features for the quad model is

generate_QUAD_Model_Production.m

To call a model that includes violin modes, the command is

quadModel = generate_QUAD_Model_Production(freq_vec, 'fiber', svnDir, 0, 0, [], 0, [], 1, 3)

The first input specifies the vector of frequencies over which to evaluate the frequency response. The second specifies that the model is a fiber model. The model must include fibers to compute fiber violin modes. The third input specifies the SVN directory, which is simply the directory listed above. The fourth through eighth inputs represent other features, including damping which are set off here. The second to last input specifies that violin modes are requested here. To add modes, this input is either 1 or 'true'. The final input is optional. It specifies the desired number of modes, starting from the fundamental mode. If left empty, the default is two. The example here is a request for 3.

The function adds violin modes by calling

makequad_with_modal_fibers.m

One can call this function directly as well.

quadModel = makequad_with_modal_fibers

With no inputs this function compiles a model with the default 2 violin modes. One can specify a different number of modes with an optional input.