



LIGO Laboratory / LIGO Scientific Collaboration

LIGO-T1200418-v3

ADVANCED LIGO

19 April 2013

LLO MC2 Violin Mode Q

Mark Barton, Keiko Kokeyama

Distribution of this document:
DCC

This is an internal working note
of the LIGO Laboratory.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW22-295
185 Albany St
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 1970
Mail Stop S9-02
Richland WA 99352
Phone 509-372-8106
Fax 509-372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone 225-686-3100
Fax 225-686-7189

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

Table of Contents

1 Introduction..... 3

1.1 Purpose and Scope 3

1.2 References..... 3

1.3 Version history 3

2 Measurement..... 3

3 Theory..... 3

4 Model parameter values..... 6

5 Results 6

5.1 Frequencies and tensions..... 7

5.2 Damping..... 9

6 Conclusion..... 11

7 Appendix..... 14

1 Introduction

1.1 Purpose and Scope

This is a writeup of a calculation to understand the measured violin mode Q's of the MC2 suspension at LLO but much of the background is applicable to all suspensions.

1.2 References

LLO alog entries [4470](#), [4472](#), [5097](#), [5280](#)

G. Cagnoli et al., Phys. Lett. A 255 (1999), p230

[T0900415](#): Upper Limit to Suspension Thermal Noise from LIGO 1 and Implications for Wire Suspensions in Advanced LIGO

T070101: [Dissipation Dilution](#)

T080096: [Wire Attachment Points and Flexure Corrections](#)

LIGO-T0900435: [HAM Small Triple Suspension \(HSTS\) Final Design Document](#)

LIGO-D020700: [HSTS Overall Assembly](#)

Cumming et al., Design and development of the advanced LIGO monolithic fused silica suspension, Class. Quantum Grav. 29 (2012) 035003.

1.3 Version history

8/28/12: -v1.

10/31/12: -v2, with analysis of data on additional wires and modes taken by Keiko and posted 10/29 in LLO alog [5097](#).

4/19/13: -v3. Various refinements suggested by Jeff K, including a calculation of net thermal noise. Incorporate refitted Q's from LLO alog 5280.

2 Measurement

The violin mode of one of the wires supporting the optic in the MC2 was measured by Keiko Kokeyama, with advice from Gaby Gonzalez and Peter Fritschel. See LLO alog entries [4470](#) and [4472](#). The frequency and Q were 631.55 Hz and 2.3×10^5 . See -v1 of this document for the analysis of that data point.

Later, Keiko measured the frequencies and Q's of the first three modes of all four wires. See LLO alog entry [5097](#). This data is reproduced in Table 2 (Section 5) below.

Later again, Keiko reprocessed the ringdown curves from alog [5097](#) using an improved Q-fitting procedure to produce a new data set in alog [5280](#), reproduced in Table 6 (Section 6) below. However it turns out that there is probably some problem with this reanalysis because it implies unphysical damping parameters.

3 Theory

3.1 Mode frequencies

To see whether these measurements were reasonable, the frequency and Q were calculated using the Mathematica model of the suspension, specifically case {"mark.barton", "20120120hstsMC2damp"} of the TripleLite2 model. This is equivalent to the Matlab parameter set `~/trunk/Common/MatlabTools/TripleModel_Production/hstsopt_metal.m` revision 2007 which has given a good fit with measured TFs. It also includes modifications, used below, for optionally assigning a separate damping function on each of the four final wires, so as to allow net pendulum mode thermal noise to be calculated from fitted parameters on the respective wires. However since neither the Mathematica nor Matlab models includes violin modes explicitly, calculating these was a matter of using numerical values from the parameter sets in general formulae as described below.

Per Eq. 2.67 of Fletcher and Rossing, to second order in small quantities, the frequency of a violin mode is

$$f_n = nf_1^0 \left(1 + b + b^2 + \frac{n^2 \pi^2}{8} b^2 \right) \quad (1.1)$$

(Their β has been renamed b to avoid confusion with the thermodynamic material property β used below.)

Here $n = 1, 2, 3, \dots$ is the mode number, and

$$f_1^0 = \frac{1}{2L} \sqrt{\frac{T}{\rho_L}}, \quad (1.2)$$

is the frequency of a wire without bending stiffness but the same length L , tension T and mass per length ρ_L .

The dimensionless quantity b (formerly β) is

$$b = \frac{2K}{L} \sqrt{\frac{YA}{T}} \quad (1.3)$$

where K is the radius of gyration of the wire, Y is the Young's Modulus, and A is the cross-sectional area, but it is closely related to the usual flexure length, defined (T080096) as

$$a = \sqrt{\frac{YI}{T}} = \frac{bL}{2} \quad (1.4)$$

Here, I is the second moment of area of the wire in the bending direction, equal to $\pi r^4 / 4$ in any direction for a wire of circular cross-section. (The moments of area of the bottom wires in the longitudinal and transverse directions are called M31 and M32 in the model code.)

It is convenient and instructive to put the above formula in terms of a :

$$f_n = \frac{n}{2L \left(1 - \frac{2a}{L} - \frac{n^2 \pi^2 a^2}{2L^2} \right)} \sqrt{\frac{T}{\rho_L}} \quad (1.5)$$

This makes it obvious that to first order in $a/L = b/2$ (≈ 0.00248 for the HSTS) the effect is simply to shorten the wire by one flexure length a at each end for all harmonics. This is consistent with the fact that a wire of non-zero bending stiffness does not bend sharply at the clamp point but along a curve that for most purposes gives the effect of a pivot a away from the attachment point. In addition, there is also a tiny shortening $n^2\pi^2 a^2/2L$ second order in both a/L and mode number n . The plain b^2 term disappears because it turns out to be an artifact of doing the expansion in the numerator rather than the denominator, i.e.,

$$1/(1-b) = 1 + b + b^2 + O(b^3) \quad (1.6)$$

3.2 Damping

The Q of the violin mode depends on the material damping factor ϕ and the dissipation dilution factor D . The damping factor is modeled as a frequency-independent structural term $\phi_{struct} = 2 \times 10^{-4}$ (Cagnoli et al. 1999; also T0900415) plus a thermoelastic term:

$$\phi(f) = \phi_{struct} + \phi_{thermo} = \phi_{struct} + \frac{2\pi f \tau \Delta}{1 + (2\pi f \tau)^2} \quad (1.7)$$

where (e.g., Cumming et al.)

$$\tau = 0.0732 C r^2 \rho_v / \kappa \quad (1.8)$$

is a time constant for heat diffusion across the wire (C is heat capacity and κ is heat conductivity), and

$$\Delta = \frac{Y T_w}{\rho_v C} \left(\alpha - \frac{\sigma \beta}{Y} \right)^2 \quad (1.9)$$

is twice the thermoelastic damping at the peak frequency $1/2\pi\tau$ (T_w is temperature, α is linear expansion, $\beta = \frac{1}{Y} \frac{dY}{dT_w}$, and $\sigma = T/A$ is stress). The magic number 0.0732 is a geometrical factor

for wires of cylindrical shape, equal to $1/4\xi^2$ where ξ is the first zero of the derivative of the first Bessel function of the first kind:

$$\left. \frac{dJ_1(x)}{dx} \right|_{x=\xi} = \frac{1}{2} (J_0(\xi) - J_2(\xi)) = 0 \quad (1.10)$$

Because the energy in a violin mode is stored in second-order stress changes of the elastic material, dissipation dilution is applicable (T070101) and the quality factor Q is not just $1/\phi$ for the material, but D/ϕ where

$$D = \frac{2a}{L} \left(1 + \frac{n^2\pi^2 a}{2L} \right) \quad (1.11)$$

Again there is a higher order term proportional to n^2 , which turns out to be significant.

4 Model parameter values

The following table gives symbol names and values for key parameters from the “production” HSTS model as of 1/20/2012 through the date of this report, which aims to be a good approximation to a generic HSTS suspension and has given good fits to measured transfer functions. In some cases, e.g., the optic mass m_3 , a value specific to L1:MC2 is available and this has been noted in the table and used where appropriate in the subsequent analysis. The model can be found in the SUS SVN at

`^/trunk/Common/MathematicaModels/TripleLite2/mark.barton/20120120hsts`

Table 1: Key parameter values from Mathematica model “20120120hsts”

Parameter (Theory)	Parameter (Mathematica)	Value (SI Units)	Note
m	m3	2.889	optic mass (generic HSTS value)
m	m3	2.90099	optic mass, L1:MC2 with prisms
L	l3	0.22	wire length
Y	Y3==Ysteel	$2.119 \cdot 10^{11}$	Young’s modulus
r	r3	0.0000597	wire radius
a	flex3	0.000546237	flexure length a (generic HSTS value)
I	M31	$9.97671 \cdot 10^{-18}$	wire second moment of area I
β	betasteel	$-2.5 \cdot 10^{-4}$	logarithmic rate of change of Young’s modulus with temperature
α	alphasteel	$12 \cdot 10^{-6}$	thermal expansion coefficient
ρ_v	rhosteel	7800	density
C	Csteel	486	heat capacity
ϕ_{struct}	phisteel	$2 \cdot 10^{-4}$	structural component of phi
τ	taufibre	0.0000813372	thermoelastic time constant
Δ	deltafibre	0.00263381	thermoelastic half maximum phi
D (n=1)	D1	0.00502663	dissipation dilution (n=1)
D (n=2)	D2	0.00520916	dissipation dilution (n=2)
D (n=3)	D3	0.00551338	dissipation dilution (n=3)

5 Results

The raw data from LLO alog [5097](#) (initial ringdown fitting algorithm) is given in Table 2.

Table 2: Raw data from initial ringdown fitting algorithm

Wire	f1 (Hz)	Q1	f2 (Hz)	Q2	f3 (Hz)	Q3
#1	631.56	247105	1263.3	163943	1895.19	141496
#2	644.977	251198	1290.09	172007	1935.45	150543
#3	660.305	249889	1320.62	187364	1981.46	177172
#4	671.023	229827	1342.16	175060	2013.56	169019

5.1 Frequencies and tensions

Using values from the model in the frequency formula of Section 1 gives $f = 650.55$ Hz. This represented a mystery in -v1 of this report when the only frequency value was for wire #1, i.e., 631.56 Hz or a 4% discrepancy. However with the luxury of frequency values for all four wires (Table 2) it becomes obvious that the problem was simply that there is a spread of values, presumably due to uneven tensions, and that the theoretical value is in the centre of the cluster.

To make this precise, the tension in each wire was inferred from the $n=1$ mode frequency using the above equation for f_n with $n=1$ - see Table 3. The total tension over 4 wires is 28.4791 N (frequency is proportional to the square root of tension and could not have been sensibly averaged or added). This is within 0.07% of the total load force from a mass of 2.90099 kg, i.e., 28.4587 N, which is excellent agreement.

Table 3: Tensions inferred from measured fundamental violin mode frequencies

Wire number	f1 (Hz)	Tension (N)	% difference from mean
#1	631.56	6.67533	-6.24
#2	644.977	6.96346	-2.20
#3	660.305	7.30009	+2.53
#4	671.023	7.54019	+5.90
Sum		28.4791	
MC2mass*g		28.4587	

Besides the sum, there are three other linear combinations of interest, which are related to the pitch, roll, and “pringle” forces on the optic:

$$\text{“pitch”} = BL+BR-FL-FR- \text{ (BL = back left, etc)}$$

$$\text{“roll”} = FL+BL-FR-BR$$

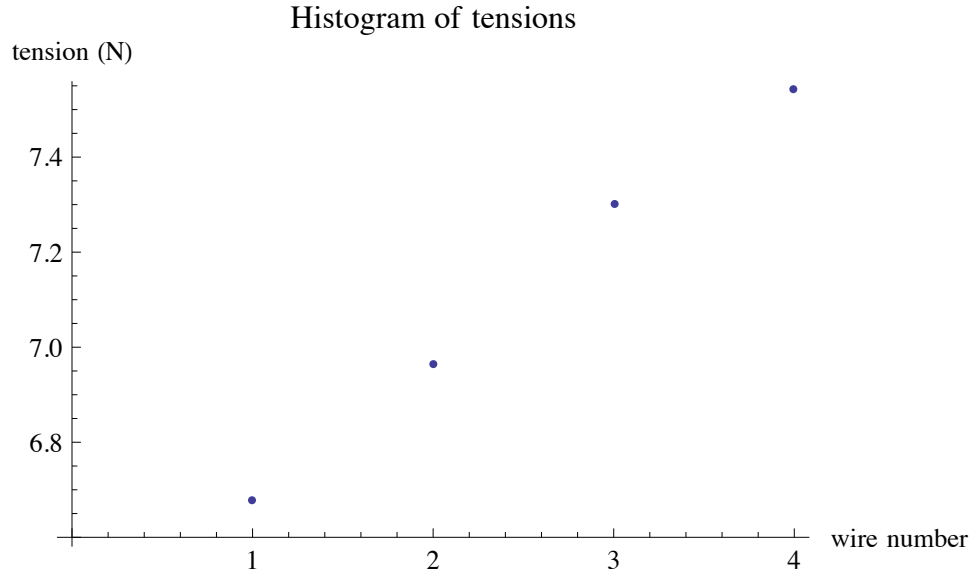
$$\text{“pringle”} = FL+BR-FR-BL$$

Because the modes are measured by exciting the whole mass, it is difficult to identify which wire position corresponds to which frequency and the wire numbers have been assigned purely by

ascending order of frequency. Thus we need to consider multiple possible mappings from wire number to position.

If the tensions are plotted as a function of wire number, as in Figure 1, they fall on a straight line to a good approximation, that is, they are equally spaced. This is a striking pattern but the significance of it is not clear and it may be a coincidence.

Figure 1: Tensions inferred from measured fundamental violin mode frequencies (plotted)



The “pringle” imbalance (between the sums over the two diagonals) can be arbitrarily large because the optic is very stiff and resists it. However it is not possible to assign four equally spaced tensions to the corners so that *only* the “pringle” imbalance is large. In some order or other, the three linear combinations will be 0.048 N, 0.528 N and 1.201 N, i.e., one small and two large.

The small imbalance is presumably “roll”. There is a 0.5° horizontal wedge which gives a left-right COM shift of 0.17 mm. This creates a torque of 0.00482 N.m, which is respectably close to 0.048 N acting over a lever arm of $r_5 = 0.08$ m, i.e., 0.0038 N.m.

The mid-sized imbalance is *probably* “pitch”. A 0.093 mm displacement forward or backward of the prism (slightly less than the gluing accuracy goal of 0.1 mm) would give a slightly smaller torque than in roll, of 0.264 N.m, but the lever arm is smaller ($r_1 = 0.005$ m), so this would correspond to a 0.528 N force imbalance. Matching the largest imbalance of 1.2 N with a 0.21 mm prism offset is not totally unthinkable but neither especially plausible.

There aren’t enough clues to determine the sign of pitch, but we can constrain roll: the optic is installed with the thick side of the wedge on the right as viewed from the back (upside down with respect to the substrate drawing). This gives a positive roll torque in standard SUS coordinates, so we need “roll” <0 as defined above to balance it, i.e., $FR+BR>FL+BL$.

All this implies either the following arrangement or its front-back mirror:

Table 4: Mapping of wire numbers to corners

Note that the front-back assignment is not constrained and could be flipped.

				Left-right sum	Front-back difference (“pitch”)	Pitch torque (N.m)
FL #1 631.56 Hz	6.67533 N	FR #3 660.305 Hz	7.30009 N	13.97542 N	-0.52823 N	-0.00264 N.m
BL #4 671.023 Hz	7.54019 N	BR #2 644.977 Hz	6.96346 N	14.50365 N		
Front-back sum:	14.21552 N		14.26355 N			
Right-left difference (“roll”):	-0.04803 N					
Roll torque	-0.00384 N.m					

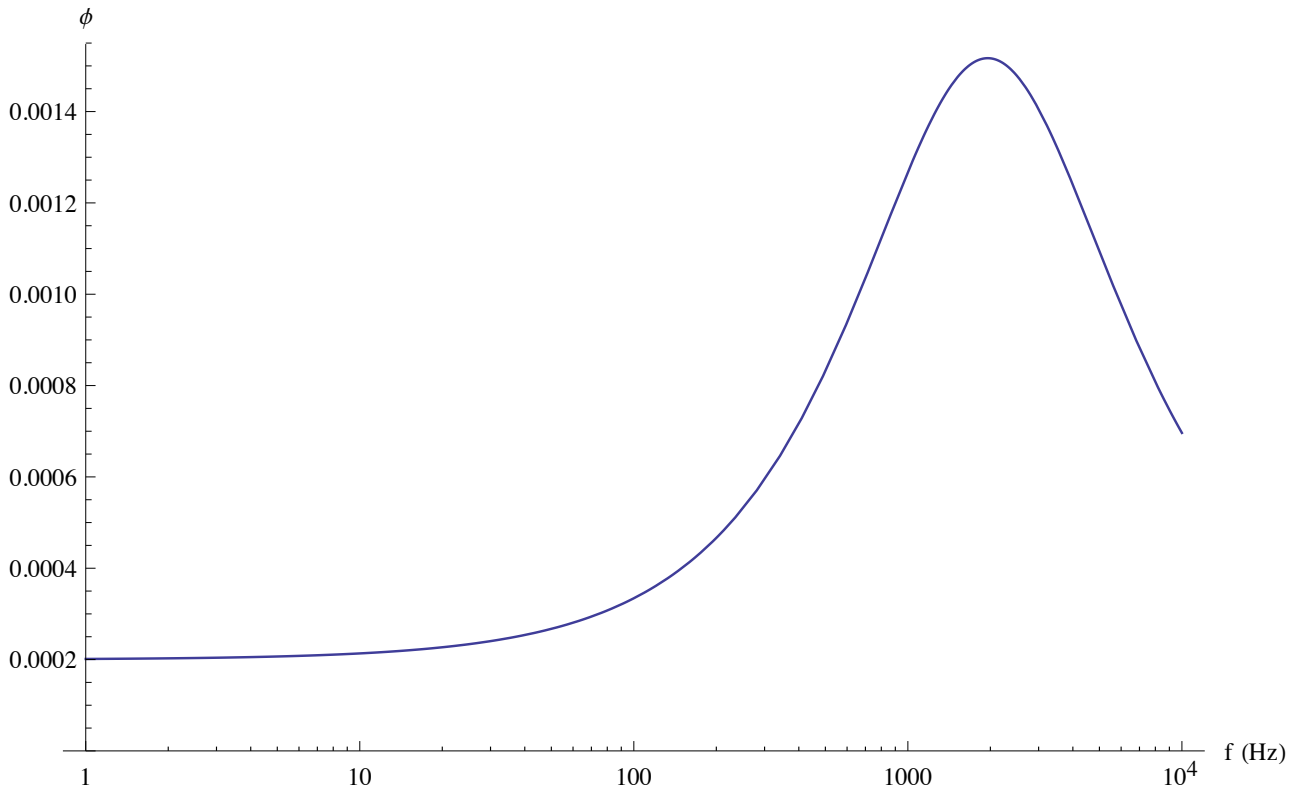
5.2 Damping

The stock damping function included with the model has two terms, one for structural and one for thermoelastic damping. The thermoelastic term turns out to contribute the bulk of the damping in the frequency range of the violin modes but the structural term is not negligible. See Figure 2.

Putting in entirely stock values, at the nominal violin mode frequency of 650.56, the total ϕ is 9.9×10^{-4} , almost 5 times the structural term, 2×10^{-4} . Together with a D of 0.00503, this gives an effective ϕ of 4.97×10^{-6} or a Q of 2.01×10^5 . Interestingly the measured range of 2.3×10^5 to 2.5×10^5 is very much in the same ballpark but slightly *better* (lower ϕ , higher Q).

This discrepancy persists when the stock damping function is used but allowance is made for the uneven tensions. Tension affects the mode Q in three ways: (i) via the thermoelastic Δ , which has a (small, negative) term proportional to stress, (ii) via pushing the frequency to a different point on the damping function, which is moderately steeply increasing in the range of the first three violin modes, and (iii) via the dissipation dilution factor, which falls (more dilution) with tension. In fact the last effect dominates and the net ϕ falls with frequency. See the small coloured dots in Figure 3, which lie above the larger coloured dots representing the measured values. However within each cluster ($n=1$, $n=2$, $n=3$) both the measured points and the stock theory with tension correction show the same characteristic downward trend with frequency from the dependency of the dissipation dilution on tension.

Figure 2: Bending loss angle ϕ (before dissipation dilution) as a function of frequency for the wire. The thermoelastic peak is visible on the right.



Since three data points (for three different harmonics) are available per wire, we can potentially say something about three different parameters of the damping function, the obvious candidates of interest being the structural ϕ and the thermoelastic τ and Δ . The Mathematica function `FindRoot[]` was used to find the parameter values that would reproduce the observed Q 's in conjunction with dissipation dilution incorporating the tensions derived earlier. (`FindFit[]` and `NSolve[]` were tried first but even when fairly narrow constraints on the sensible solutions were provided, they failed to give curves passing through or even particularly near the data.)

The parameter values obtained are listed in Table 5, together with the stock values for comparison. Curves based on the parameters have been plotted in Figure 3, and do in fact pass precisely through the data points they're based on. As an aid to the eye, the mode number n has been treated as a continuous function of frequency and interpolated so as to give a smooth curve that can be compared to the underlying damping function. Despite the fit being based on only three points on a somewhat unrepresentative section of the curve (the upward slope just to the left of the thermoelastic peak) the consistency of the fitted parameters and the naturalness of the fitted curves is quite good. The fitted values of structural ϕ are clustered around the stock value of 2×10^{-4} (mean 2.06×10^{-4}), whereas the fitted values of thermoelastic Δ are clustered around about 71% of the stock value.

Figure 3: Fitted damping curves with Q 's from original ringdown fitting algorithm

Black dots are prediction from stock model. Wire #1=red, #2=blue, #3=green,#4=brown. Large coloured dots are measured $1/Q$. Small dots are predictions using stock values for all parameters

except wire tensions. Dashed curves use fitted parameters (the mode number n has been interpolated to give a continuous curve).

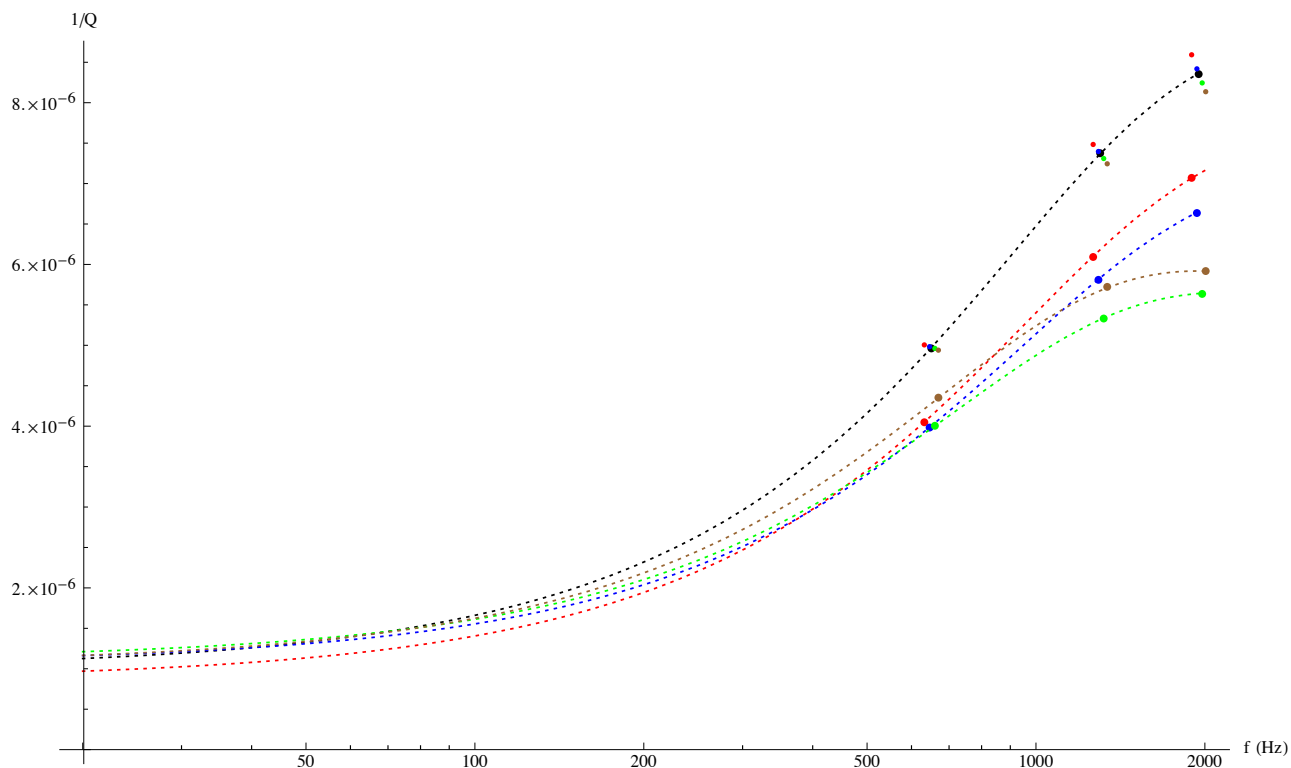


Table 5: Fitted damping parameters with Q's from original ringdown fitting algorithm

	f1 (Hz)	phi	delta	tau (s)	$1/(2*\pi*\tau)$ (Hz)
stock	650.585	0.0002	0.00263381	0.0000813372	1956.73
#1	631.56	0.00016832	0.00214845	0.0000789362	2016.25
#2	644.977	0.000212517	0.00196291	0.0000794312	2003.68
#3	660.305	0.000227039	0.00166091	0.0000984017	1617.4
#4	671.023	0.000218143	0.00184757	0.000103027	1544.78
mean	651.966	0.000206505	0.00190496	0.0000899491	1795.53
stdev	17.3009	0.0000261493	0.000204534	0.000012575	249.431

6 Refitted data

Later, Keiko reanalyzed the original ringdown curves with a different fitting algorithm that was hoped to be better. See LLO alog [5280](#). The new data is reproduced in Table 6. The frequencies are the same (and thus all the above conclusions about tensions) but the Qs are somewhat different, especially for the $n=3$ mode.

Table 6: Raw data with Q's from "improved" fitting algorithm

Wire number	f1 (Hz)	Q1	f2 (Hz)	Q2	f3 (Hz)	Q3
#1	631.56	234598	1263.3	160933	1895.19	195142
#2	644.977	238817	1290.09	165250	1935.45	178511
#3	660.305	239867	1320.62	167053	1981.46	174990

#4	671.023	229072	1342.16	164167	2013.56	177825
----	---------	--------	---------	--------	---------	--------

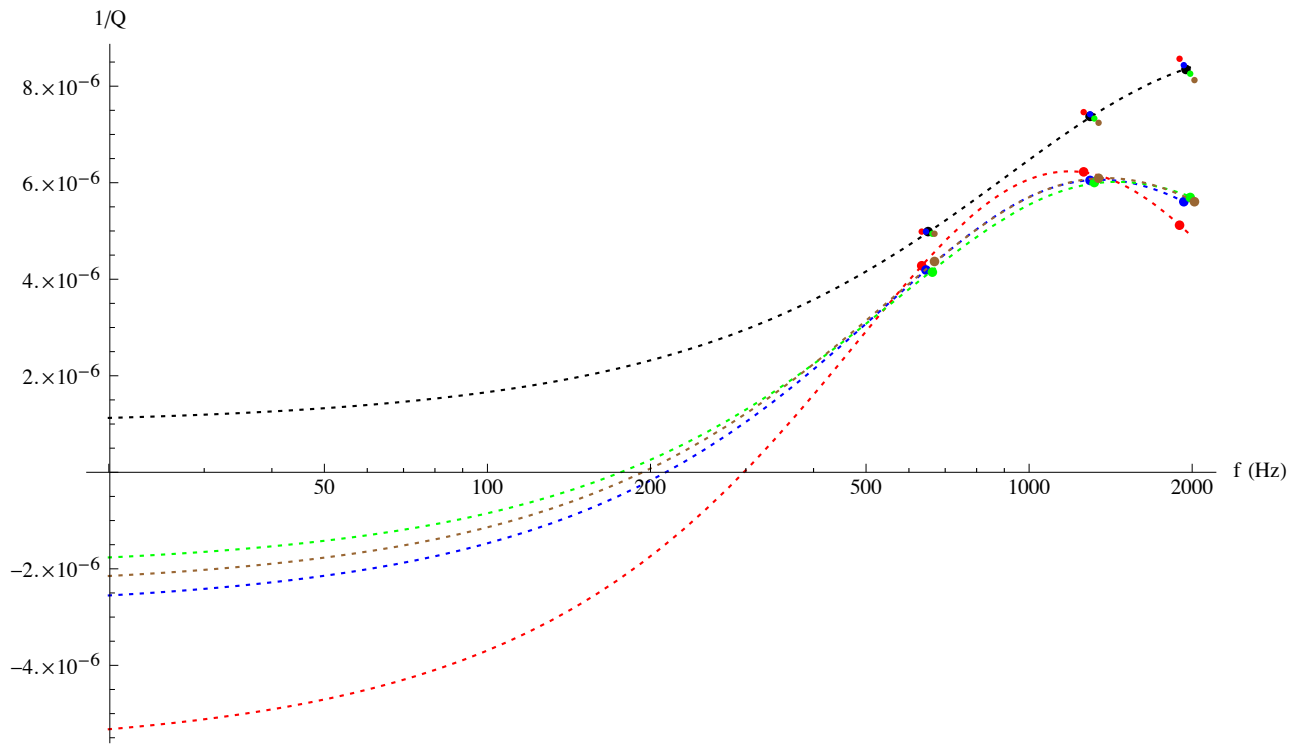
This rather disturbs the subsequent curve fitting of ϕ , τ and Δ . The τ 's and Δ 's are substantially larger than before (the thermoelastic peak frequency is lower), and the ϕ values come out negative, as in Table 7 and Figure 4. Since a negative ϕ is unphysical, there is presumably something wrong with the new algorithm, and we persist with the original Q's.

Table 7: Fitted damping parameters with Q's from "improved" ringdown fitting algorithm

	f1 (Hz)	phi	delta	tau (s)	1/(2*pi*tau) (Hz)
stock	650.585	0.0002	0.00263381	0.0000813372	1956.73
#1	631.56	-0.00112109	0.00457979	0.000139954	1137.2
#2	644.977	-0.00056371	0.00343057	0.000126003	1263.11
#3	660.305	-0.000407667	0.00315013	0.000118878	1338.81
#4	671.023	-0.00049905	0.00341633	0.000122863	1295.38
mean	651.966	-0.00064788	0.0036442	0.000126925	1258.62
stdev	17.3009	0.000321904	0.000636916	9.16241×10^{-6}	86.6895

Figure 4: Fitted damping curves with Q's from "improved" ringdown fitting algorithm

Black dots are prediction from stock model. Wire #1=red, #2=blue, #3=green,#4=brown. Large coloured dots are measured 1/Q. Small dots are predictions using stock values for all parameters except wire tensions. Dashed curves use fitted parameters (the mode number n has been interpolated to give a continuous curve).



7 Effect on net thermal noise

To examine the effect that different amounts of damping at different corners would have, the stock Mathematica model was altered to allow different damping functions on each of the four lower wires, and the fitted parameters from the initial Q's were plugged in. The modified model is in the SUS SVN at

`^/trunk/Common/MathematicaModels/TripleLite2/mark.barton/20120120hstsMC2damp`

The results are compared with those from the stock model in Figure 5 and Table 8. In fact the thermal noise is almost identical up to 30 Hz and about 11% lower at 1000 Hz.

Figure 5: Net pendulum thermal noise

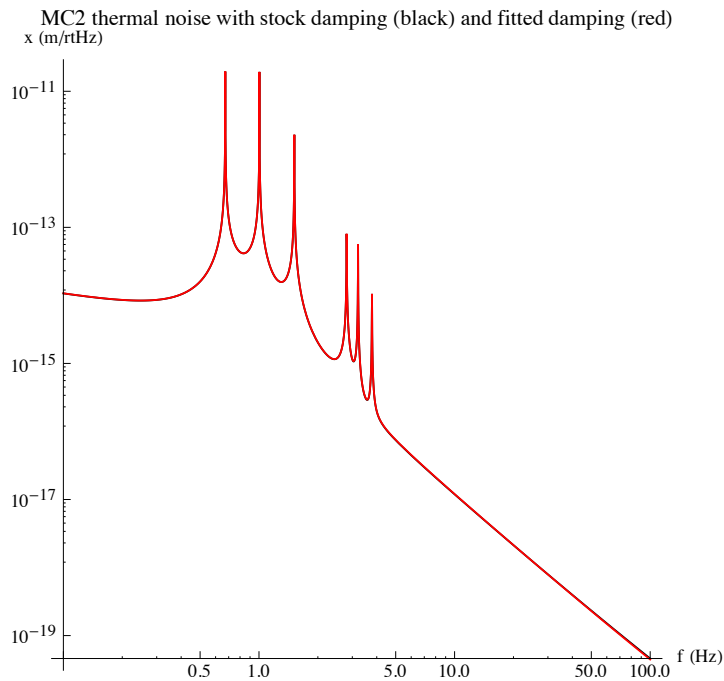


Table 8: Net thermal noise at representative frequencies

f (Hz)	x (m/rHz), stock	x (m/rHz), fitted	fitted/stock
0.1	1.07×10^{-14}	1.0738×10^{-14}	1.00356
0.316228	8.73506×10^{-15}	8.75147×10^{-15}	1.00188
1.	7.82564×10^{-13}	7.8366×10^{-13}	1.0014
3.16228	2.13397×10^{-15}	2.16377×10^{-15}	1.01396
10.	1.19274×10^{-17}	1.20306×10^{-17}	1.00865
31.6228	6.98528×10^{-19}	6.95182×10^{-19}	0.99521
100.	4.60104×10^{-20}	4.45137×10^{-20}	0.96747
316.228	3.5083×10^{-21}	3.26725×10^{-21}	0.931292
1000.	2.83237×10^{-22}	2.53209×10^{-22}	0.893982

8 Conclusion

The observed frequencies tell a consistent story. The tensions implied by the frequencies sum to almost exactly the known payload force, and the “roll” imbalance matches the expected value based on the horizontal wedge. It’s difficult to say definitively which of the two remaining imbalances is “pitch” and which is “pringle”, but the smaller of the two is what would be expected if the prisms were at the limit of their 0.1 mm front-back gluing tolerance and so is probably pitch. Conversely the larger is an additional factor two, consistent with the fact that there is little constraint on the “pringle” imbalance.

The damping from the first ringdown fitting algorithm ties up quite well with the model and what small discrepancy there is (of order 25%) is (i) in the advantageous direction and (ii) better explained by the thermoelastic term (rather than the structural term) being low.

The Q’s from the second ringdown fitting algorithm imply unphysical values of ϕ , and we propose to ignore them.

To the extent the initial Q values can be trusted, the effect on thermal noise at the optic is very small (<11%) and in the favorable direction compared to the stock model.

9 Appendix

In the PDF version of this report, a printout of the Mathematica notebook containing the calculation will be appended.

calculations will be appended.

Calculation of TripleLite2 model with violin mode stuff (for T1200418-v3)

Note: this notebook needs to be in a calculation directory of case {"mark.barton","20120120hstsMC2damp"} of the TripleLite2 model to be run.

■ Setup

Switches to enable loading of previously saved results instead of recalculating from scratch

```
useprecomputed = True; (* set to True to use saved results from precomputed subdirectory *)
If[useprecomputed,
  exceptdamping = True, (* False by default, True to recalculate just damping-dependent stuff*)
  exceptdamping = True (* DON'T CHANGE *)
];
loadcasefromuser["ASUS3L2ModelCaseDefn.m"];

overridesorig = overrides;

modelcase
{mark.barton, 20120120hstsMC2damp}

modelcasecomment
Equivalent to Jeff K's hstsopt_metal.m revision
  2007 of 1/19/12, but with independent damping on lower wires.

Reset[All]

Calculate[constval];
```

■ Parameters of interest

Wire length

l3 /. constval

0.22

Young's modulus

y3 /. constval

2.119×10^{11}

Wire radius

r3 /. constval

0.0000597

Mass (nominal value from model)

m3 /. constval

2.889

Mass of bare IMCC-03 (L1, MC2) from nebula

MC2bare = 2.898;

Mass of primary prism

prismmass1 = 3980 * (0.197 * 0.0254) * (0.223 * 0.0254) * (0.79 * 0.0254) / 2

0.00113175

Mass of secondary prism

prismmass2 = rhosteel * (0.079 * 0.0254) * (0.091 * 0.0254) * (0.787 * 0.0254) / 2 /. constval

0.000361584

Mass of MC2 with prisms

MC2mass = MC2bare + 2 * prismmass1 + 2 * prismmass2

2.90099

Flexure length (nominal)

flex3 / . constval

0.000546237

Twice flexure length as a fraction of total (cf. beta=2a/L)

2 * flex3 / l3 / . constval

0.00496579

Half front-back wire separation

s1 / . constval

0.005

Half left-right wire separation

n5 / . constval

0.08

COM offset due to 0.5° horizontal wedge

COMy = 0.17 / 1000;

Wire tension (nominal value from model)

tension = m3 * g / nw3 / . constval

7.08527

Roll torque due to wedge

rolltorque = m3 * COMy * g / . constval

0.00481799

rolltorque * n5 / . constval

0.000385439

Extra tension to offset roll torque

deltatensionroll = rolltorque / n5 / 4 / . constval

0.0150562

Extra tension as a percentage

100 * deltatensionroll / tension

0.2125

Pitch torque due to 0.1 mm prism offset (gluing tolerance)

pitchtorque = 0.0001 * m3 * g / . constval

0.00283411

Extra tension to offset pitch torque

deltatensionpitch = pitchtorque / s1 / . constval

0.566822

0.528 * s1 / pitchtorque / . constval

0.93151

0.528 * s1 / . constval

0.00264

1.201 * sl / pitchtorque / . constval

2.11883

100 * deltatensionpitch / tension

8.

Wire second moment of area

M31 / . constval (* a.k.a. I *)

9.97671 × 10⁻¹⁸

Fundamental violin mode for wires above mass 3 (the optic)

**f3n = n * Sqrt [(m3 * g / c3 / nw3) / (rhosteel * A3)] /
(2 * l3 * (1 - 2 * flex3 / l3 - n^2 * Pi^2 * flex3^2 / 2 / l3^2))**

$$\frac{n \sqrt{\frac{g m_3}{A_3 c_3 n w_3 \rho_{\text{steel}}}}}{2 l_3 \left(1 - \frac{2 \text{flex}_3}{l_3} - \frac{\text{flex}_3^2 n^2 \pi^2}{2 l_3^2} \right)}$$

f31 = f3n / . n → 1 / . constval

650.585

f32 = f3n / . n → 2 / . constval

1301.29

f33 = f3n / . n → 3 / . constval

1952.23

Giles' formula for dissipation dilution, D

**D1 = (2 / l3) * Sqrt [Y2 * M31 / tension] *
(1 + (1 / 2 / l3) * (Pi * n)^2 * Sqrt [Y2 * M31 / tension]) / . constval / . n → 1**

0.00502663

**D2 = (2 / l3) * Sqrt [Y2 * M31 / tension] *
(1 + (1 / 2 / l3) * (Pi * n)^2 * Sqrt [Y2 * M31 / tension]) / . constval / . n → 2**

0.00520916

**D3 = (2 / l3) * Sqrt [Y2 * M31 / tension] *
(1 + (1 / 2 / l3) * (Pi * n)^2 * Sqrt [Y2 * M31 / tension]) / . constval / . n → 3**

0.00551338

Structural component of wire phi (before dissipation dilution)

phisteel / . constval

0.0002

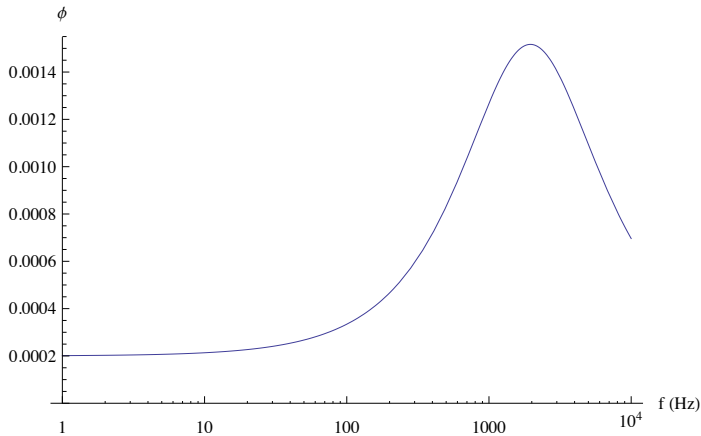
Expected Q at fundamental violin mode neglecting thermoelastic damping

Qstruct = 1 / (D1 * phisteel) / . constval

994 702.

Total phi(f) including thermoelastic (note damping tag is "fibreatype" but model really does have wire parameters)

```
LogLinearPlot[damping[imag, fibreatype][f] /. constval,
  {f, 1, 10000}, PlotRange -> {0, Automatic}, AxesLabel -> {"f (Hz)", "phi"}]
```



Total phi at fundamental violin mode (before dissipation dilution)

```
damping[imag, fibreatype][f31] /. constval
```

0.000988535

Total phi at fundamental violin mode (after dissipation dilution)

```
D1 * damping[imag, fibreatype][f31] /. constval
```

4.969×10^{-6}

Q at fundamental violin mode (after dissipation dilution)

```
Q1 = 1 / (D1 * damping[imag, fibreatype][f31]) /. constval
```

201248.

```
Q2 = 1 / (D2 * damping[imag, fibreatype][f32]) /. constval
```

135720.

```
Q3 = 1 / (D3 * damping[imag, fibreatype][f33]) /. constval
```

119571.

■ Keiko's data

Raw data from <https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=5097>

```
keikodata20121029 = {
  {{631.56, 247105}, {1263.3, 163943}, {1895.19, 141496}},
  {{644.977, 251198}, {1290.09, 172007}, {1935.45, 150543}},
  {{660.305, 249889}, {1320.62, 187364}, {1981.46, 177172}},
  {{671.023, 229827}, {1342.16, 175060}, {2013.56, 169019}}
};
```

Refitted data from <https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=5280>

```
keikodata20121114 = {
  {{631.56, 234598}, {1263.3, 160933}, {1895.19, 195142}},
  {{644.977, 238817}, {1290.09, 165250}, {1935.45, 178511}},
  {{660.305, 239867}, {1320.62, 167053}, {1981.46, 174990}},
  {{671.023, 229072}, {1342.16, 164167}, {2013.56, 177825}}
};
```

■ Analysis of tensions

The frequency expressed as a function of tension

```
f1s = keikodata20121029[[All, 1, 1]];
```

```
foftension = (Sqrt[(m3 * g / nw3) / (rhosteel * A3)] / (2 * (13 - 2 * flexoften))) /.
```

$$\text{flexoften} \rightarrow \sqrt{\frac{M31 \text{ nw3 } Y3}{g \text{ m3}}} / . g * m3 / nw3 \rightarrow \text{ten} / . nw3 / (g * m3) \rightarrow 1 / \text{ten} / . \text{constval}$$

$$\frac{53.5024 \sqrt{\text{ten}}}{0.22 - 0.00290796 \sqrt{\frac{1}{\text{ten}}}}$$

The tensions required to account for the observed frequencies (N)

```
tensions = Table[ten /. NSolve[f1s[[i]] == foftension, ten][[1]], {i, 4}]
{6.67533, 6.96346, 7.30009, 7.54019}
```

Deviations from the mean (N)

```
tensions - Mean[tensions]
{-0.444438, -0.156303, 0.18032, 0.420421}
```

Percentage deviations from the mean

```
100 * (tensions - Mean[tensions]) / Mean[tensions]
{-6.24231, -2.19535, 2.53266, 5.90499}
```

Net tension

```
netten = Plus@@tensions
28.4791
```

Net load force

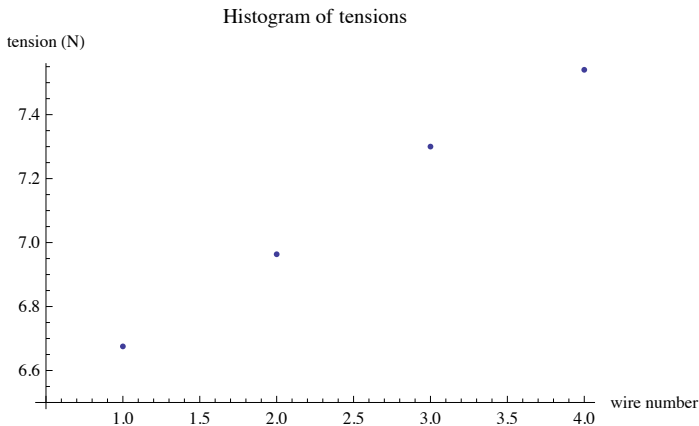
```
load = MC2mass * g /. constval
28.4587
```

Percent tension-load discrepancy

```
100 * (netten / load - 1)
0.0716248
```

Plot of tensions

```
ListPlot[tensions, PlotLabel -> "Histogram of tensions",
  AxesLabel -> {"wire number", "tension (N)"}, AxesOrigin -> {0.5, 6.5}]
```



Functions giving the three linear combinations representing tension imbalances

```
pitch[{{FL_, FR_}, {BL_, BR_}}] := BL + BR - FL - FR
roll[{{FL_, FR_}, {BL_, BR_}}] := FL + BL - FR - BR
pringle[{{FL_, FR_}, {BL_, BR_}}] := FL + BR - FR - BL
```

Most likely mapping of tension values to the four corners of the optic, in {{FL,FR},{BL,BR}} order

```
corners = {{tensions[[1]], tensions[[3]]}, {tensions[[4]], tensions[[2]]}};
```

The imbalances for the likely mapping

```
{pitch[corners], roll[corners], pringle[corners]}
```

```
{0.528236, -0.0480325, -1.20148}
```

The torque corresponding to the pitch imbalance

```
pitch[corners] * s1 /. constval
```

```
0.00264118
```

The roll torque corresponding to the roll imbalance

```
roll[corners] * n5 /. constval
```

```
-0.0038426
```

■ Damping comparison using mostly stock parameters but correcting for tension differences

```
Doftension[n_] =
```

```
(2 / 13) * Sqrt[Y3 * M31 / ten] * (1 + (1 / 2 / 13) * (Pi * n) ^ 2 * Sqrt[Y3 * M31 / ten]) /. constval
```

$$0.013218 \left(1 + 0.0326142 n^2 \sqrt{\frac{1}{\text{ten}}} \right) \sqrt{\frac{1}{\text{ten}}}$$

```
Doftension[1] /. ten → tensions[[1]]
```

```
0.00518057
```

The dissipation dilution factors allowing for different tensions, for the four wires and three harmonics

```
Ds = Table[(2 / 13) * Sqrt[Y3 * M31 / tensions[[i]]] *
```

```
(1 + (1 / 2 / 13) * (Pi * n) ^ 2 * Sqrt[Y3 * M31 / tensions[[i]])] /. constval, {i, 4}, {n, 3}]
```

```
{{0.00518057, 0.00537432, 0.00569722}, {0.00507094, 0.00525666, 0.0055662},  
{0.00495123, 0.00512839, 0.00542366}, {0.00487083, 0.00504235, 0.00532822}}
```

The thermoelastic tau expressed as a function of tension (actually tension-independent)

```
tauoftension = taufibre /. constval
```

```
0.0000813372
```

```
taus = Table[tauoftension /. ten → tensions[[i]], {i, 4}]
```

```
{0.0000813372, 0.0000813372, 0.0000813372, 0.0000813372}
```

The thermoelastic delta parameter expressed as a function of tension

```
deltaoftension = deltafibre /. overrides /. g * m3 / nw3 → ten /. constval
```

$$1.62106 \times 10^7 (0.000012 + 1.05368 \times 10^{-7} \text{ten})^2$$

The delta values for the four wires, allowing for the tension differences

```
deltas = Table[deltaoftension /. ten → tensions[[i]], {i, 4}]
```

```
{0.00261599, 0.00262851, 0.00264317, 0.00265366}
```

The phi function from the standard damping model in terms of the three constants to be fitted

```
phifn[f_, phi_, tau_, delta_] =
```

```
damping[imag, fibreatype][f] /. overrides /. taufibre → tau /. deltafibre → delta /.  
phisteel → phi
```

$$\text{phi} + \frac{6.28319 \text{delta f tau}}{1 + 39.4784 \text{f}^2 \text{tau}^2}$$

Dissipation diluted phi as a function of tension

```
fitfn[f_, n_, ten_, phi_, tau_, delta_] = DofTension[n] * phifn[f, phi, tau, delta]
```

$$0.013218 \left(\text{phi} + \frac{6.28319 \text{ delta f tau}}{1 + 39.4784 \text{ f}^2 \text{ tau}^2} \right) \left(1 + 0.0326142 \text{ n}^2 \sqrt{\frac{1}{\text{ten}}} \right) \sqrt{\frac{1}{\text{ten}}}$$

■ Analysis of “improved” Q’s

```
keikodata = keikodata20121114;
```

Equations expressing the requirement that the fitted curves pass through the data points

```
Do[
  wireeqns[w] = Table[fitfn[keikodata[[w, n, 1]], n, tensions[[w]], phi, tau, delta] ==
    1 / keikodata[[w, n, 2]], {n, 3}],
  {w, 1, 4}
]
```

Convert to expressions, LHS-RHS

```
Do[
  wireexp[w] = wireeqns[w] /. x_ == y_ -> x - y,
  {w, 1, 4}
]
```

Solve using FindRoot[] (which turns out to do a better job than NSolve[] and Minimize[])

```
wireroot[1] = FindRoot[wireexp[1], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[2] = FindRoot[wireexp[2], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[3] = FindRoot[wireexp[3], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[4] = FindRoot[wireexp[4], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
```

Quick summary table

```
TableForm[Table[{phi, tau, delta} /. wireroot[w], {w, 1, 4}],
  TableHeadings -> {keikodata[[All, 1, 1]], {phi, "tau (s)", delta}}]
```

	phi	tau (s)	delta
631.56	-0.00112109	0.000139954	0.00457979
644.977	-0.00056371	0.000126003	0.00343057
660.305	-0.000407667	0.000118878	0.00315013
671.023	-0.00049905	0.000122863	0.00341633

A function to give the expected phi from the stock model, except with corrections for observed tension.

```
netphitencorronly[w_, n_] :=
  Ds[[w, n]] * damping[imag, fibreatype][keikodata[[w, n, 1]]] /. overrides /.
  taufibre -> taus[[w]] /. deltafibre -> deltas[[w]] /. constval
```

Summary plot

```
Show[
  LogLinearPlot[fitfn[f, f / f31, m3 * g / 4, phisteel, taufibre, deltafibre] /. constval,
    {f, 20, 2000}, PlotStyle -> {Black, Dotted}],
  ListLogLinearPlot[{{f31, 1 / Q1}, {f32, 1 / Q2}, {f33, 1 / Q3}},
    PlotStyle -> {Black, PointSize[Medium]}],

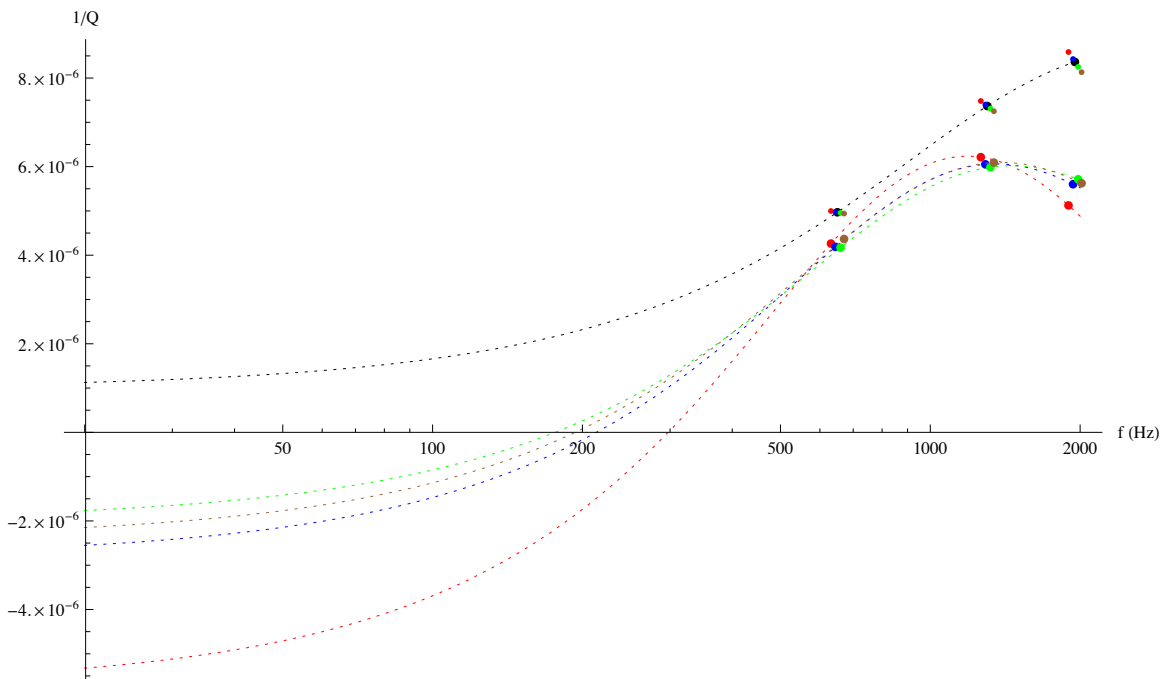
  ListLogLinearPlot[keikodata[[1]] /. {f_, Q_} :> {f, 1 / Q},
    PlotStyle -> {Red, PointSize[Medium]}],
  ListLogLinearPlot[Table[{keikodata[[1, n, 1]], netphitencorronly[1, n]}, {n, 3}],
    AxesOrigin -> {20, 0}, PlotStyle -> {Red}],
  LogLinearPlot[fitfn[f, f / keikodata[[1, 1, 1]], tensions[[1]], phi, tau, delta] /.
    wireroot[1], {f, 20, 2000}, PlotStyle -> {Red, Dotted}],

  ListLogLinearPlot[keikodata[[2]] /. {f_, Q_} :> {f, 1 / Q},
    PlotStyle -> {Blue, PointSize[Medium]}],
  ListLogLinearPlot[Table[{keikodata[[2, n, 1]], netphitencorronly[2, n]}, {n, 3}],
    AxesOrigin -> {20, 0}, PlotStyle -> {Blue}],
  LogLinearPlot[fitfn[f, f / keikodata[[2, 1, 1]], tensions[[2]], phi, tau, delta] /.
    wireroot[2], {f, 20, 2000}, PlotStyle -> {Blue, Dotted}],

  ListLogLinearPlot[keikodata[[3]] /. {f_, Q_} :> {f, 1 / Q},
    PlotStyle -> {Green, PointSize[Medium]}],
  ListLogLinearPlot[Table[{keikodata[[3, n, 1]], netphitencorronly[3, n]}, {n, 3}],
    AxesOrigin -> {20, 0}, PlotStyle -> {Green}],
  LogLinearPlot[fitfn[f, f / keikodata[[3, 1, 1]], tensions[[3]], phi, tau, delta] /.
    wireroot[3], {f, 20, 2000}, PlotStyle -> {Green, Dotted}],

  ListLogLinearPlot[keikodata[[4]] /. {f_, Q_} :> {f, 1 / Q},
    PlotStyle -> {Brown, PointSize[Medium]}],
  ListLogLinearPlot[Table[{keikodata[[4, n, 1]], netphitencorronly[4, n]}, {n, 3}],
    AxesOrigin -> {20, 0}, PlotStyle -> {Brown}],
  LogLinearPlot[fitfn[f, f / keikodata[[4, 1, 1]], tensions[[4]], phi, tau, delta] /.
    wireroot[4], {f, 20, 2000}, PlotStyle -> {Brown, Dotted}],

  AxesOrigin -> {Automatic, 0},
  PlotRange -> All,
  AxesLabel -> {"f (Hz)", "1/Q"}
]
```



Averages and standard deviations

```

fmean = Mean[
  {keikodata[[1, 1, 1]], keikodata[[2, 1, 1]], keikodata[[3, 1, 1]], keikodata[[4, 1, 1]]}]
651.966

fstdev = StandardDeviation[
  {keikodata[[1, 1, 1]], keikodata[[2, 1, 1]], keikodata[[3, 1, 1]], keikodata[[4, 1, 1]]}]
17.3009

phimean =
  Mean[{phi /. wireroot[1], phi /. wireroot[2], phi /. wireroot[3], phi /. wireroot[4]}]
-0.00064788

phistdev = StandardDeviation[
  {phi /. wireroot[1], phi /. wireroot[2], phi /. wireroot[3], phi /. wireroot[4]}]
0.000321904

deltamean = Mean[
  {delta /. wireroot[1], delta /. wireroot[2], delta /. wireroot[3], delta /. wireroot[4]}]
0.0036442

deltastdev = StandardDeviation[
  {delta /. wireroot[1], delta /. wireroot[2], delta /. wireroot[3], delta /. wireroot[4]}]
0.000636916

taumean =
  Mean[{tau /. wireroot[1], tau /. wireroot[2], tau /. wireroot[3], tau /. wireroot[4]}]
0.000126925

taustdev = StandardDeviation[
  {tau /. wireroot[1], tau /. wireroot[2], tau /. wireroot[3], tau /. wireroot[4]}]
9.16241 × 10-6

ffmean = Mean[{1 / (2 * Pi * tau) /. wireroot[1], 1 / (2 * Pi * tau) /. wireroot[2],
  1 / (2 * Pi * tau) /. wireroot[3], 1 / (2 * Pi * tau) /. wireroot[4]}]
1258.62

ffstdev = StandardDeviation[{1 / (2 * Pi * tau) /. wireroot[1], 1 / (2 * Pi * tau) /. wireroot[2],
  1 / (2 * Pi * tau) /. wireroot[3], 1 / (2 * Pi * tau) /. wireroot[4]}]
86.6895

```

Summary table for publication

```

fitted = {
  {f31, phisteel, deltafibre, taufibre, 1 / taufibre / 2 / Pi} /. constval,
  {keikodata[[1, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[1],
  {keikodata[[2, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[2],
  {keikodata[[3, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[3],
  {keikodata[[4, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[4],
  {fmean, phimean, deltamean, taumean, ffmean},
  {fstdev, phistdev, deltastdev, taustdev, ffstdev}
};

TableForm[fitted, TableHeadings → {"stock", "#1", "#2", "#3", "#4", "mean", "stdev"},
  {"f1 (Hz)", "phi", "delta", "tau (s)", "1/(2*pi*tau) (Hz)"}]

```

	f1 (Hz)	phi	delta	tau (s)	1/(2*pi*tau) (Hz)
stock	650.585	0.0002	0.00263381	0.0000813372	1956.73
#1	631.56	-0.00112109	0.00457979	0.000139954	1137.2
#2	644.977	-0.00056371	0.00343057	0.000126003	1263.11
#3	660.305	-0.000407667	0.00315013	0.000118878	1338.81
#4	671.023	-0.00049905	0.00341633	0.000122863	1295.38
mean	651.966	-0.00064788	0.0036442	0.000126925	1258.62
stdev	17.3009	0.000321904	0.000636916	9.16241 × 10 ⁻⁶	86.6895

Average fitted delta as percentage

```
100 * deltamean / deltafibre /. constval
```

```
138.362
```

■ Analysis of older Q's

```
keikodata = keikodata20121029;
```

Equations expressing the requirement that the fitted curves pass through the data points

```
Do[
  wireeqns[w] = Table[fitfn[keikodata[[w, n, 1]], n, tensions[[w]], phi, tau, delta] ==
    1 / keikodata[[w, n, 2]], {n, 3}],
  {w, 1, 4}
]
```

Convert to expressions, LHS-RHS

```
Do[
  wireexp[w] = wireeqns[w] /. x_ == y_ -> x - y,
  {w, 1, 4}
]
```

Solve using FindRoot[] (which turns out to do a better job than NSolve[] and Minimize[])

```
wireroot[1] = FindRoot[wireexp[1], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[2] = FindRoot[wireexp[2], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[3] = FindRoot[wireexp[3], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
wireroot[4] = FindRoot[wireexp[4], {phi, phisteel /. constval},
  {delta, deltafibre /. constval}, {tau, taufibre /. constval}];
```

Quick summary table

```
TableForm[Table[{phi, tau, delta} /. wireroot[w], {w, 1, 4}],
  TableHeadings -> {keikodata[[All, 1, 1]], {phi, "tau (s)", delta}}]
```

	phi	tau (s)	delta
631.56	0.00016832	0.0000789362	0.00214845
644.977	0.000212517	0.0000794312	0.00196291
660.305	0.000227039	0.0000984017	0.00166091
671.023	0.000218143	0.000103027	0.00184757

A function to give the expected phi from the stock model, except with corrections for observed tension.

```
netphitencorronly[w_, n_] :=
  Ds[[w, n]] * damping[imag, fibreatype][keikodata[[w, n, 1]]] /. overrides /.
  taufibre -> taus[[w]] /. deltafibre -> deltas[[w]] /. constval
```

Summary plot


```
Show[
LogLinearPlot[fitfn[f, f / f31, m3 * g / 4, phisteel, taufibre, deltafibre] /. constval,
  {f, 20, 2000}, PlotStyle → {Black, Dotted}],
ListLogLinearPlot[{{f31, 1 / Q1}, {f32, 1 / Q2}, {f33, 1 / Q3}},
  PlotStyle → {Black, PointSize[Medium]}],

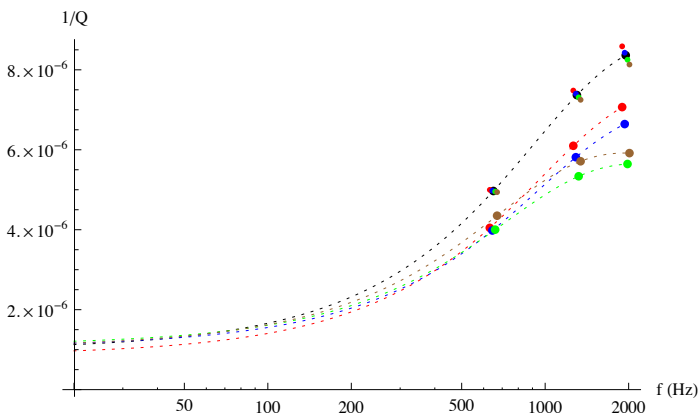
ListLogLinearPlot[keikodata[[1]] /. {f_, Q_} := {f, 1 / Q},
  PlotStyle → {Red, PointSize[Medium]}],
ListLogLinearPlot[Table[{keikodata[[1, n, 1]], netphitencorronly[1, n]}, {n, 3}],
  AxesOrigin → {20, 0}, PlotStyle → {Red}],
LogLinearPlot[fitfn[f, f / keikodata[[1, 1, 1]], tensions[[1]], phi, tau, delta] /.
  wireroot[1], {f, 20, 2000}, PlotStyle → {Red, Dotted}],

ListLogLinearPlot[keikodata[[2]] /. {f_, Q_} := {f, 1 / Q},
  PlotStyle → {Blue, PointSize[Medium]}],
ListLogLinearPlot[Table[{keikodata[[2, n, 1]], netphitencorronly[2, n]}, {n, 3}],
  AxesOrigin → {20, 0}, PlotStyle → {Blue}],
LogLinearPlot[fitfn[f, f / keikodata[[2, 1, 1]], tensions[[2]], phi, tau, delta] /.
  wireroot[2], {f, 20, 2000}, PlotStyle → {Blue, Dotted}],

ListLogLinearPlot[keikodata[[3]] /. {f_, Q_} := {f, 1 / Q},
  PlotStyle → {Green, PointSize[Medium]}],
ListLogLinearPlot[Table[{keikodata[[3, n, 1]], netphitencorronly[3, n]}, {n, 3}],
  AxesOrigin → {20, 0}, PlotStyle → {Green}],
LogLinearPlot[fitfn[f, f / keikodata[[3, 1, 1]], tensions[[3]], phi, tau, delta] /.
  wireroot[3], {f, 20, 2000}, PlotStyle → {Green, Dotted}],

ListLogLinearPlot[keikodata[[4]] /. {f_, Q_} := {f, 1 / Q},
  PlotStyle → {Brown, PointSize[Medium]}],
ListLogLinearPlot[Table[{keikodata[[4, n, 1]], netphitencorronly[4, n]}, {n, 3}],
  AxesOrigin → {20, 0}, PlotStyle → {Brown}],
LogLinearPlot[fitfn[f, f / keikodata[[4, 1, 1]], tensions[[4]], phi, tau, delta] /.
  wireroot[4], {f, 20, 2000}, PlotStyle → {Brown, Dotted}],

AxesOrigin → {Automatic, 0},
PlotRange → All,
AxesLabel → {"f (Hz)", "1/Q"}
]
```



Averages and standard deviations

```
fmean = Mean[
  {keikodata[[1, 1, 1]], keikodata[[2, 1, 1]], keikodata[[3, 1, 1]], keikodata[[4, 1, 1]]}]
651.966

fstdev = StandardDeviation[
  {keikodata[[1, 1, 1]], keikodata[[2, 1, 1]], keikodata[[3, 1, 1]], keikodata[[4, 1, 1]]}]
17.3009
```

```

phimean =
  Mean[{phi /. wireroot[1], phi /. wireroot[2], phi /. wireroot[3], phi /. wireroot[4]}]
0.000206505

phistdev = StandardDeviation[
  {phi /. wireroot[1], phi /. wireroot[2], phi /. wireroot[3], phi /. wireroot[4]}]
0.0000261493

deltamean = Mean[
  {delta /. wireroot[1], delta /. wireroot[2], delta /. wireroot[3], delta /. wireroot[4]}]
0.00190496

deltastdev = StandardDeviation[
  {delta /. wireroot[1], delta /. wireroot[2], delta /. wireroot[3], delta /. wireroot[4]}]
0.000204534

taumean =
  Mean[{tau /. wireroot[1], tau /. wireroot[2], tau /. wireroot[3], tau /. wireroot[4]}]
0.0000899491

taustdev = StandardDeviation[
  {tau /. wireroot[1], tau /. wireroot[2], tau /. wireroot[3], tau /. wireroot[4]}]
0.000012575

ffmean = Mean[{1 / (2 * Pi * tau) /. wireroot[1], 1 / (2 * Pi * tau) /. wireroot[2],
  1 / (2 * Pi * tau) /. wireroot[3], 1 / (2 * Pi * tau) /. wireroot[4]}]
1795.53

ffstdev = StandardDeviation[{1 / (2 * Pi * tau) /. wireroot[1], 1 / (2 * Pi * tau) /. wireroot[2],
  1 / (2 * Pi * tau) /. wireroot[3], 1 / (2 * Pi * tau) /. wireroot[4]}]
249.431

```

Summary table for publication

```

fitted = {
  {f31, phisteel, deltafibre, taufibre, 1 / taufibre / 2 / Pi} /. constval,
  {keikodata[[1, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[1],
  {keikodata[[2, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[2],
  {keikodata[[3, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[3],
  {keikodata[[4, 1, 1]], phi, delta, tau, 1 / tau / 2 / Pi} /. wireroot[4],
  {fmean, phimean, deltamean, taumean, ffmean},
  {fstdev, phistdev, deltastdev, taustdev, ffstdev}
};

TableForm[fitted, TableHeadings → {"stock", "#1", "#2", "#3", "#4", "mean", "stdev"},
  {"f1 (Hz)", "phi", "delta", "tau (s)", "1/(2*pi*tau) (Hz)"}]

```

	f1 (Hz)	phi	delta	tau (s)	1/(2*pi*tau) (Hz)
stock	650.585	0.0002	0.00263381	0.0000813372	1956.73
#1	631.56	0.00016832	0.00214845	0.0000789362	2016.25
#2	644.977	0.000212517	0.00196291	0.0000794312	2003.68
#3	660.305	0.000227039	0.00166091	0.0000984017	1617.4
#4	671.023	0.000218143	0.00184757	0.000103027	1544.78
mean	651.966	0.000206505	0.00190496	0.0000899491	1795.53
stdev	17.3009	0.0000261493	0.000204534	0.000012575	249.431

Average fitted delta as percentage

```

100 * deltamean / deltafibre /. constval
72.3272

```

■ Plots with damping

```

tablefs = Table[N[10^logf], {logf, -1, 3, 1/2}]
{0.1, 0.316228, 1., 3.16228, 10., 31.6228, 100., 316.228, 1000.}

```

■ Calculate/plot thermal noise with stock damping

```

useprecomputed = True; (* set to True to use saved results from precomputed subdirectory *)
If[useprecomputed,
  exceptdamping = True, (* False by default, True to recalculate just damping-dependent stuff*)
  exceptdamping = True (* DON'T CHANGE *)
];
overrides = overridesorig;

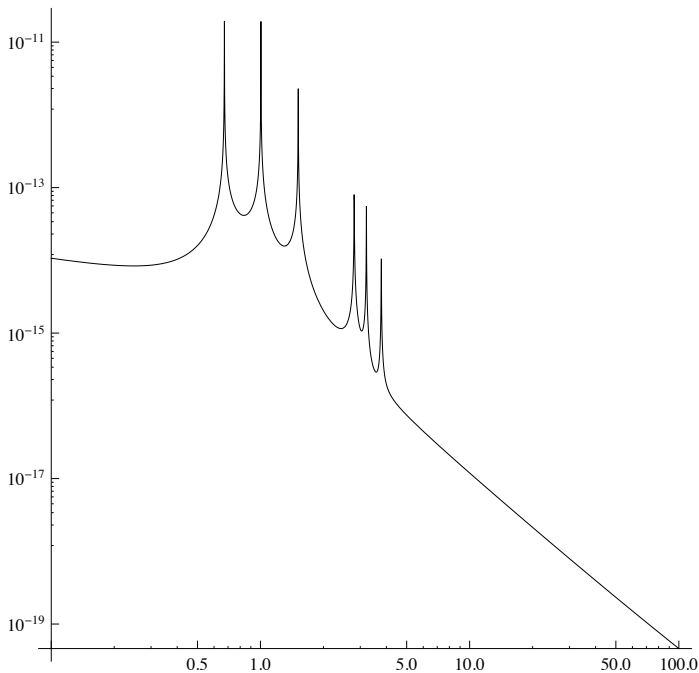
Reset[All]; Calculate[Stage2]

stocktn = Table[Sqrt[noise2[eom2, opticxoutput, tablefs[[i]]]], {i, Length[tablefs]}]

{1.07 × 10-14, 8.73506 × 10-15, 7.82564 × 10-13, 2.13397 × 10-15,
 1.19274 × 10-17, 6.98528 × 10-19, 4.60104 × 10-20, 3.5083 × 10-21, 2.83237 × 10-22}

DoWithStatus["Plotting stage 2 x thermal noise",
stocktnplot = plotTN[eom2, opticxoutput, 0.1, 100, 1, PlotStyle->{Black}]]

```



■ Calculate/plot thermal noise with fitted damping parameters

```

overrides = Override[
  overridesorig,
  {
    phisteellf -> phi /. wireroot[1],
    deltafibrelf -> delta /. wireroot[1],
    taufibrelf -> tau /. wireroot[1],
    phisteellb -> phi /. wireroot[3],
    deltafibrelb -> delta /. wireroot[3],
    taufibrelb -> tau /. wireroot[3],
    phisteelrf -> phi /. wireroot[4],
    deltafibrerf -> delta /. wireroot[4],
    taufibrerf -> tau /. wireroot[4],
    phisteelrb -> phi /. wireroot[2],
    deltafibrerb -> delta /. wireroot[2],
    taufibrerb -> tau /. wireroot[2]
  }
];

useprecomputed = True; (* set to True to use saved results from precomputed subdirectory *)
If[useprecomputed,
  exceptdamping = True, (* False by default, True to recalculate just damping-dependent stuff*)
  exceptdamping = True (* DON'T CHANGE *)
];

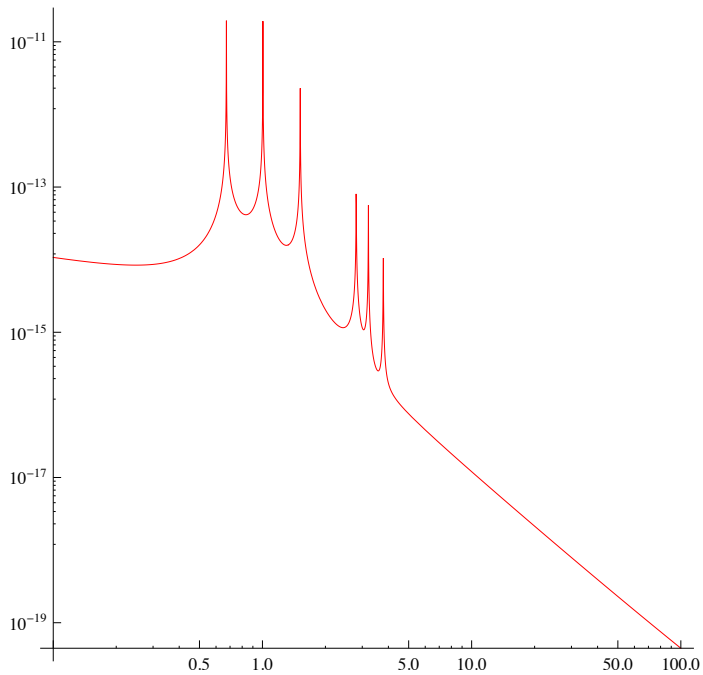
Reset[All]; Calculate[Stage2]

```

```
fittedtn = Table[Sqrt[noise2[eom2, opticxoutput, tablefs[[i]]]], {i, Length[tablefs]}]
```

```
{1.0738 × 10-14, 8.75147 × 10-15, 7.8366 × 10-13, 2.16377 × 10-15,  
1.20306 × 10-17, 6.95182 × 10-19, 4.45137 × 10-20, 3.26725 × 10-21, 2.53209 × 10-22}
```

```
fittedtnplot = plotTN[eom2, opticxoutput, 0.1, 100, 1, PlotStyle->{Red}]
```



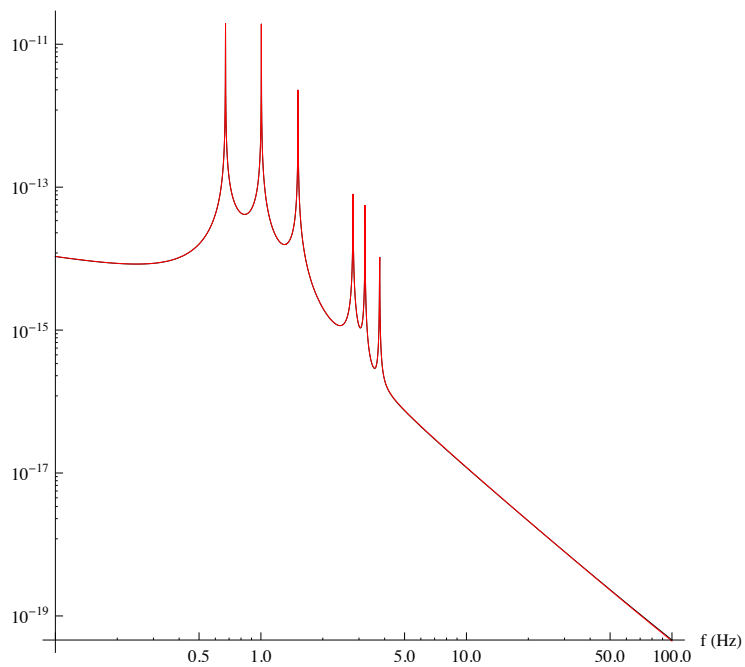
■ Comparison

```
Show[stocktnplot, fittedtnplot,
```

```
PlotLabel -> "MC2 thermal noise with stock damping (black) and fitted damping (red)",
```

```
AxesLabel -> {"f (Hz)", "x (m/rtHz)"}]
```

MC2 thermal noise with stock damping (black) and fitted damping (red)
x (m/rtHz)



TableForm[Transpose[{{tablefs, stocktn, fittedtn, fittedtn / stocktn}}, TableHeadings -> {None, {"f (Hz)", "x (m/rtHz), stock", "x (m/rtHz), fitted", "fitted/stock"}}]]

f (Hz)	x (m/rtHz), stock	x (m/rtHz), fitted	fitted/stock
0.1	1.07×10^{-14}	1.0738×10^{-14}	1.00356
0.316228	8.73506×10^{-15}	8.75147×10^{-15}	1.00188
1.	7.82564×10^{-13}	7.8366×10^{-13}	1.0014
3.16228	2.13397×10^{-15}	2.16377×10^{-15}	1.01396
10.	1.19274×10^{-17}	1.20306×10^{-17}	1.00865
31.6228	6.98528×10^{-19}	6.95182×10^{-19}	0.99521
100.	4.60104×10^{-20}	4.45137×10^{-20}	0.96747
316.228	3.5083×10^{-21}	3.26725×10^{-21}	0.931292
1000.	2.83237×10^{-22}	2.53209×10^{-22}	0.893982