Advanced LIGO global control and ETM suspension violin modes LIGO-T050107-01-K

K.A. Strain

June 30, 2005

1 Introduction and references

It is planned to feedback global control forces to the lower 3 stages of the ETM. This document considers issues related to the violin modes of the lowest suspension stage. Note that there are additional concerns, such as the time required to settle after excitation, that are not considered here, and that this document does not by itself provide a complete answer to the requirements for violin mode damping.

The details presented here are likely to be modified when end-to-end modelling of the global control is carried out. In the meantime the conclusions are thought to be useful.

Some background is given by Fritschel in G010086. Generic requirements for the suspensions are given in T010007-03.

The local control is assumed to use hybrid damping, interaction between global and local control is mild, except as concerns feedback to the upper-intermediate stage. The global controller for that stage has been designed to tolerate a wide range of local control closed-loop responses.

In lieu of a complete model of the suspension model, including violin modes, some rough estimates of the height of violin modes enables evaluation of controllers designed to have small loop gain at the violin mode frequencies. The fundamental modes are required to be above 400 Hz. It is assumed that the modes are matched to better than 10% in frequency within a single suspension (should be much better), and are in a band starting from 400 Hz. No consideration is given to the higher order modes, some mild additional filtering will probably be required to deal with them.

The mass of a ribbon is < 0.1 g, or $\sim 3 \times 10^{-6}$ of the 40 kg test mass. This goes some way to suppressing the effect of the violin modes on the test mass dynamics. The Q of the mode is expected to be $\leq 10^{10}$. Taking these together we can expect the motion of the test mass to be enhanced up to $\sim 3 \times 10^{4}$ times on resonance. Thus any global control loop intended to work with such a system should have gain somewhat less than -90 dB in the band from 400 Hz to 440 Hz (or other higher or narrower and/or higher band to be determined when the mode frequencies are known). In fact it turns out to be quite practical to design a controller that achieves this degree of attenuation while retaining good phase margin around the 'optical spring' frequency – required to ensure system stability against radiation pressure fluctuations.

update in version 01 It was realised that the estimate of the height of violin modes used in the model was not correct as it neglects stiffness of the ribbons. The result of incorporating the correct effect was a need to slightly alter the controllers. The changes are in the detail of the filters for the violin modes – there had been in any case considerable margin, and by tweaking the parameters there is again some margin, even for $Q \sim 10^{10}$ which is higher than expected. It is also realised that the calculations are close to the limit at which the MATLAB algorithms are accurate – pole locations were used as the final to check of stability, and the accuracy seems to be about 10^{-12} real part on the violin modes. Thus to ensure stability additional margin was allowed (i.e. considered stable if real part $< 10^{-11}$) for all closed loop poles. The controller does not damp the violin modes. The details of the controller and wording of a few sentences have been changed.

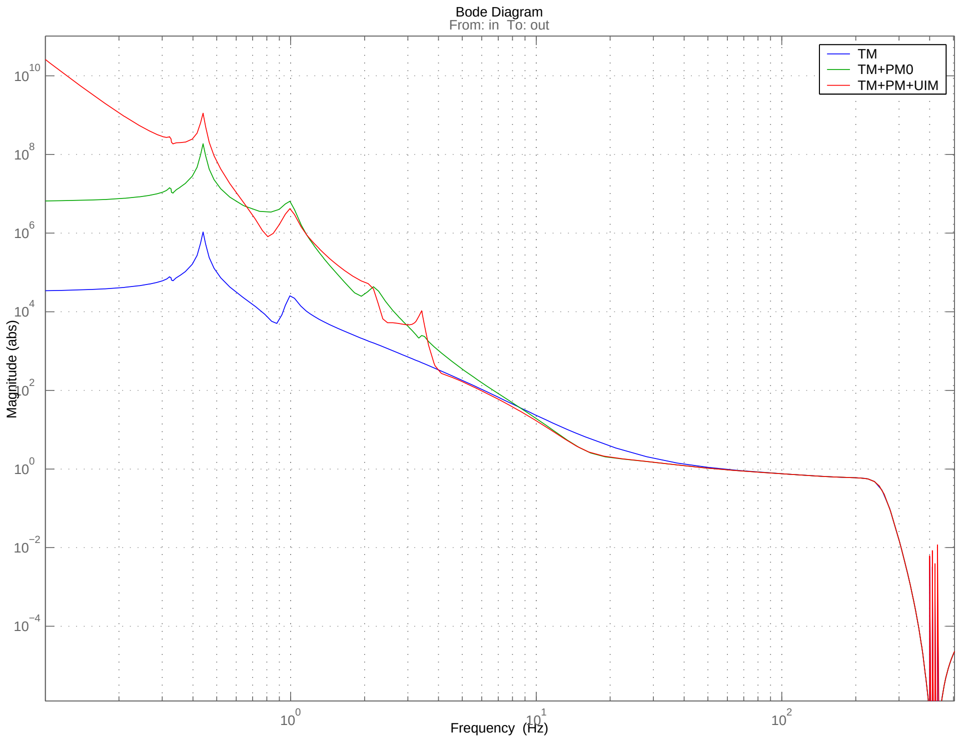

Figure 1: Example open loop responses of a practical global control servo. More detail is given in the text. The plot resolution is not sufficient to show the violin modes. The least-attenuated one extends to -3 dB open loop gain. The TM loop is shown near maximum design gain. Change in -01 the new controller is different in the fine detail

2 Evaluation of the controller

The controller has unity gain up to about 80 Hz, with phase margin at least 30 degrees. This should be adequate for optical spring control.

The gain around 1 Hz is over 130 dB. Residual seismic motion at the suspension point of order 1 pm would be reduced to mirror motion of order 0.3 am. This is adequate.

The gain around 0.1 Hz is over 200 dB. Residual seismic motion at the suspension point of order 0.1 μm would be reduced to mirror motion of order 10 am. This is certainly adequate.

The crossovers are at about 15 Hz (TM to PM) and about 2 Hz (PM to UIM), in the latter case the residual effect of suspension modes means there are multiple crossings. Both crossovers could be improved given further work.

The controller is not extremely complex and is designed to be simple to implement digitally – see the appendix for details.

3 Conclusion

Control of the TM can be adequate, in the steady state, even with undamped violin modes. Something else will determine whether damping is needed (probably settling time after a disturbance).

4 Appendix: details of the controller: changed in -01

The controller given connects the sensor for TM position to the appropriate force input on the suspension model (TM, PM or UIM). (Standard MATLAB model for silica quad suspension ref: N.A. Robertson.) The controller primitive blocks are standard ones distributed with the MATLAB model. They can be understood remembering the frequencies are all in Hz and the last parameter is just a gain factor: transint – real pole then real zero; transdif – real zero then real pole; sculte – complex pole (freq,Q) then complex notch (freq,Q). Thus transdif provides controlled phase lead, transint provides phase lag and sculte provides a, very effective, combination of low pass filter and stop-band notch. Doubtless the controllers can be improved.

First the controller for the TM, depending on the spread of violin modes and their mean frequency, it may be possible to simplify this controller by removing one or more 'sculte' stages and modifying the remaining ones (to retain good gain margin).

```
gain = -5e5;
ag = 0; bg = 0; cg = 0; dg = gain;
fid1  = 25;
[a,b,c,d] = transdif(0.8*fid1,fid1*8,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transdif(0.8*fid1,fid1*8,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid1/5,fid1,5);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid1/5,fid1,5);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(190,0.6,400,20,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(240,1.4,410,10,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(250,3,424,10,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(260,4,430,60,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
gc = ss(ag,bg,cg,dg);
```

Next the very simple controller for the PM. Note that modified and/or additional notches could possibly be needed depending on the spread of violin modes.

```
gain = -6e8;
ag = 0; bg = 0; cg = 0; dg = gain;
fid3  = 4;
[a,b,c,d] = transint(fid3/10,fid3,10);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transdif(fid3,2*fid3,1);
```

```
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(200,10,405,10,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = sculte(250,10,425,10,1);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
hc = ss(ag,bg,cg,dg);
```

Finally the controller for the UIM. No notches are required (note that wire and fibre violin modes are not permitted to couple significantly). The aim was to obtain large gain around 0.1 Hz where there is considerable residual suspension point motion.

```
gain = -2e8;
ag = 0; bg = 0; cg = 0; dg = gain;
fid4  = 1.15;
[a,b,c,d] = transdif(fid4,5*fid4,1); %deal with not well damped long mode
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transdif(fid4,5*fid4,1); %deal with not well damped long mode
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid4/100,fid4,100);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid4/100,fid4,100);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid4/100,fid4,100);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid4/100,fid4,100);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
[a,b,c,d] = transint(fid4/100,fid4,100);
[ag,bg,cg,dg] = series(ag,bg,cg,dg,a,b,c,d);
kc = ss(ag,bg,cg,dg);
```

Note that the controllers all have the same number of poles and zeros and are amenable to conversion to IIR form if the poles and zeros are arranged in pairs of closest spacing.

The fibres were modelled as follows (changed in -01) note had to be fudged to give the answer calculated from the ribbon shape. Cannot be correct for round fibres (problem scales with dilution factor).

```
mmirror = 40;
mfibre  = 2200*0.6*0.001*0.0001;
aw = 0; bw = 0; cw = 0; dw = 1;
fo = 400;
df = 10;    %spread of modes
q  = 1e10; %includes factor ~10 safety
ddf = mfibre/mmirror;  %was version -00 replaced by next line in -01
ddf = 0.0003; %stiffness model needs to be tweaked to give right answer
for index = 1:4
    f = fo + (index-1)*df;
    zs  = pi*f*(-1/q + i*sqrt(4 - 1/(q^2)));
    z   = [conj(zs) zs]';
    k   = 1;
    ps  = pi*f*(1+ddf)*(-1/q + i*sqrt(4 - 1/(q^2)));
```

```
    p   = [conj(ps) ps]';
    [a,b,c,d] = zp2ss(z,p,k);
    [aw,bw,cw,dw] = series(aw,bw,cw,dw,a,b,c,d);
end
wires = ss(aw,bw,cw,dw);
```

This works by spacing pairs of high Q zeros then poles at the right separation to represent the mass ratio. The response is very hard to plot in MATLAB, but the poles and zeros are in the correct locations to within $\sim 10^{-12}$, and the peak of the response has the expected value when zoomed in to micro-hertz resolution. Note that a very conservative figure was taken for the true Q of the violin modes.