LLO Calibration for S1 data

Rana Adhikari, Gabriela González, Brian O'Reilly

October 5, 2002

Abstract

A summary of calibration information gathered for S1 data.

1 Model used

We use a Simulink interferometer model, based on Rana's "darm_08.mdl", called "S1darm_08.mdl", shown in Fig1.

The parameters used to load the model are in a file S1darm.m, and are as follow (following the diagram, starting from AS_Q)

- Input matrix constant gain asq2darm=0.014 (from conlog).
- LSC Digital Filters

  We used "Foton" to translate the numbers in the coefficient filters file (/cvs/cds/llo/chans/L1.txt) used during S1, to zeros and poles to use in LTI Matlab models. The filters are digital filters, but we use them in Matlab/Simulink as analog filters (we haven't yet figured out hot make compatible digital and analog filters in the linearization programs). This will introduce errors when getting close to the Nyquist frequency, 8kHz.

  ```
  FM1=zpk([-75.757+i*153.31;-75.757-i*153.31],[0;0],1.99078);
  FM2=zpk([-62.8318;-628.241],[-4389.6-i*4389.6;-4389.6+i*4389.6],976.28);
  FM3=zpk([],[-8485+i*8485;-8485-i*8485],1.43991e+08);
  FM4=zpk([-125.663],[-6.28319],0.996375);
  FM8 = zpk([-6207.83;-6207.84],...
  [-25045.8+i*0.00338783;-25045.8-i*0.00338783],16.2775);
  FM9=zpk([-126.26+i*255.513;-126.26-i*255.513],...
  [-12.6263+i*25.552;-12.6263-i*25.552],1.00005);
  DARMDF= FM1 * FM2 * FM3 * FM8 * FM9;
  ```

- Gain knob: GW_K=0.232 (also from conlog). This gain is negative in the LSC code, but we need a negative sign somewhere to make the Simulink loop stable, as the real loop was (this was probably a minus sign in some whitening/dewhitening board).
- Output matrix darm2lm=-0.33 (from output matrix to ETMs, also in conlog).
- ETM filters: there were steep stopbands near the internal modes (near 7 kHz) and a violin mode notch (near 345 Hz) used for the ETM drive. We could not include the stopband filters without running into what looked like numerical problems, so since they are outside the gw band, we left them out. Thus, we use just the violin notch for the test mass filter:

  ```
  violin = zpk([0+i*2152.47;0-i*2152.47;0+i*2156.2;0-i*2156.2], ...
  ```

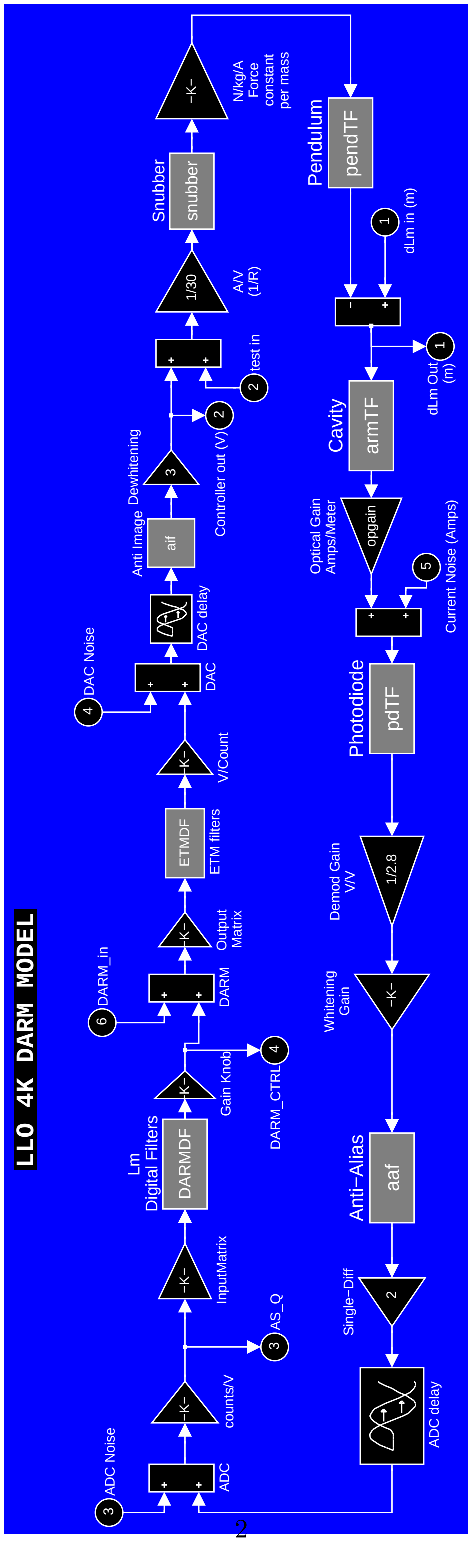


Figure 1: Simulink model S1darm_08.mdl

```
[-10.7623+i*2152.44;-10.7623-i*2152.44;-10.781+i*2156.18; ...
-10.781-i*2156.18],1);
ETMDF=violin;
```

- DAC gain= (5/32768) Volt/count
- DAC delay: 75μsec. This delay was found to get a good fit to the loop gains.
- Anti-imaging filter: `[z,p,k] = ellip(4,4,60,2*pi*7570,'s'); aif = zpk(z,p,k*`$10^{4/20}$`);`
- Dewhitening gain=3
- Voltage to current gain: 1/30 Amp/Volt
- Snubber =0.63 (this factor is fitted to get the measured DC drive of 1.72nm/ct for DARM_CTRL, it probably represents the effective gain in the output matrix of the controller)
- Coil/pendulum gain: 0.064/10.3 N/kg/Amp
- Pendulum transfer function (m/N/kg) `Q = 10; w0 = 2*pi*0.76; pendTF = 2 * tf(1,[1 w0/Q w0`^{2}`]);` The factor of 2 accounts for Ly-Lx.
- Arm Cavity
  ```
  fc = 87.3; % Hz
  armTF = 2*pi*fc * tf(1,[1 2*pi*fc]);
  ```
- Optical gain: 0.47e7 Amps/meter. This gain is multiplied by a fitted factor depending on alignment.
- LSC Photodiode gain: pdTF=400*10 V/Amp.
- Demodulation gain=1/2.8
- Whitening gain: $10^{18/20}$
- Anti-Aliasing filter:
  ```
  [z,p,k] = ellip(8,0.035,80,2*pi*7570,'s');
  aaf = zpk(z,p,k);
  ```
- Single-to-differential gain: 2
- ADC delay: 50μsec.
- ADC gain: 32768/10 counts/Volt

The measured DC calibration for DARM_CTRL into effective mirror motion (for (Ly-Lx)/2?) was 1.72nm/count. In this model, that gain is made up of the output matrix (0.33), times the DC gain of the ETM filters (1.0), times the DAC gain (5/32768 Volts/counts), past the DAC delay (unity gain), the antiimaging filter (unity gain), times the dewhitening gain (3.0), times the voltage-to-current gain (1/30 Amp/Volt), times the force constant (0.064/10.3 N/m/Amp), times the DC pendulum gain (1/w0^2=0.044 m/N/kg). This product is:

x= 0.33*(5/32768 V/ct)*3*(1/30 A/V)*(0.064/10.3 N/kg/A)*(0.044m/M/kg)*DARM[ct] =2.74 (nm/ct)* DARM[ct]

Since the measured value is 1.72nm/count, we use the snubber “fudge factor” of 1.72/2.74=0.63. As mentioned before, this factor is probably due to the coefficients in the output matrix for the length to each coil drive, which average 85% of the full value (for minimizing length to angle drive at DC).

2 Model and Measurement comparison

We compare the results of this model to the measured swept sine on Sept 06, 2002 at 23:02 UTC, saved in `darm_loop_020906.xml`. We fit a factor to multiply the optical gain so that the loop gains measured and modelled agree.

The swept sine was done injecting a signal in DARM_ERR_EXC. The response of MICH_CTRL, AS_I, AS_Q, REFL_I, REFL_Q, DARM_ERR, CARM_CTRL and DARM_CTRL were measured in DTT, with 7 “A” channels. The measurement was done for frequencies between and 36.7 Hz and 7.4kHz. The open loop gain is the transfer function of AS_Q times the darm2asq gain (0.014) to DARM_ERR. In DTT, this can be plotted using a Yslope=0.014 in the “Units” menu, and is a fast way to check the loop gain, especially near the unity gain frequency. In this measurement, the unity gain is at about 250 Hz. From DTT, we exported the transfer function AS_Q/DARM_ERR, and compared it with our Matlab/Simulink model.

To use the model, we first linearize it using the “linmod” function, and create a state-space representation. All the parameters of the Simulink model from the previous section are collected in the “S1darm.m” file, then the Simulink model is called, and the linearization is done in Matlab as follows:

```
S1darm
S1darm_08
[A,B,C,D] = linmod(’S1darm_08’);
darmsys = ss(A,B,C,D);
```

The open and closed loop gains can then be calculate in Matlab using bode plot functions (the function “mybodesys” is similar to “bode”, but uses frequency in Hertz):

```
f=logspace(log10(f1),log10(f2),1e4);
F=mybodesys(darmsys(2,2),f);
CLG=mybodesys(darmsys(1,1),f);
OLG=-F./CLG;
```

The comparison between measured and modelled open loop gains is shown in Fig2. The best agreement is found with an alignment gain (multiplying the optical gain) of 1.2. The unity gain found in the model with a finely space frequency vector is 248 Hz. The ratio of the magnitudes is within ten percent between 40 Hz and 1 KHz. We believe the growing discrepancy at higher frequencies is due to the analog treatment of the digital filters. The comparison of the measured and modelled phase is within 5 degrees up to 2 kHz.

3 Calibrated noise

We now use the model we have for the interferometer transfer functions, and compare the calibrated noise for AS_Q, using a cavity pole; and DARM_CTRL, using a pendulum transfer function.

We measured with DTT (and then exported to use in Matlab), amplitude spectral densities for AS_Q and DARM_CTRL, in counts/rtHz. These spectra were measured at the closest time to the time the swept sine was done, on Sept 06 at 22:53, in the same locked stretch, so we assume the gains were the same as measured and fitted with the model. For DARM_CTRL, we use the measured DC calibration of 1.72 nm/count; for AS_Q, we use a calibration such that it intersects DARM_CTRL at the measured unity gain frequency of 250 Hz. This calibration results in 2.34×10^{-15} m/count. The resulting noise spectra are the residual noise, or error length, when the loops are closed (from AS_Q) and the correcting noise, or control length, applied by the servo (from DARM_CTRL). We do this in Matlab as follows:

```
load 020906_2253_cal %load raw measured spectra asqc and dc, in counts/rtHz,
                     %and frequency f in Hertz.

w0 = 2 * pi * 0.76; %pendulum frequency
Q = 10;
fc = 87.3;          %cavity pole

darmcal = 1.72e-9; %1.72nm/count, measured calibration for DARM_CTRL
asqcal = 2.34e-15;  %m/count, to meet DARM_CTRL at ugf at 250 Hz
```

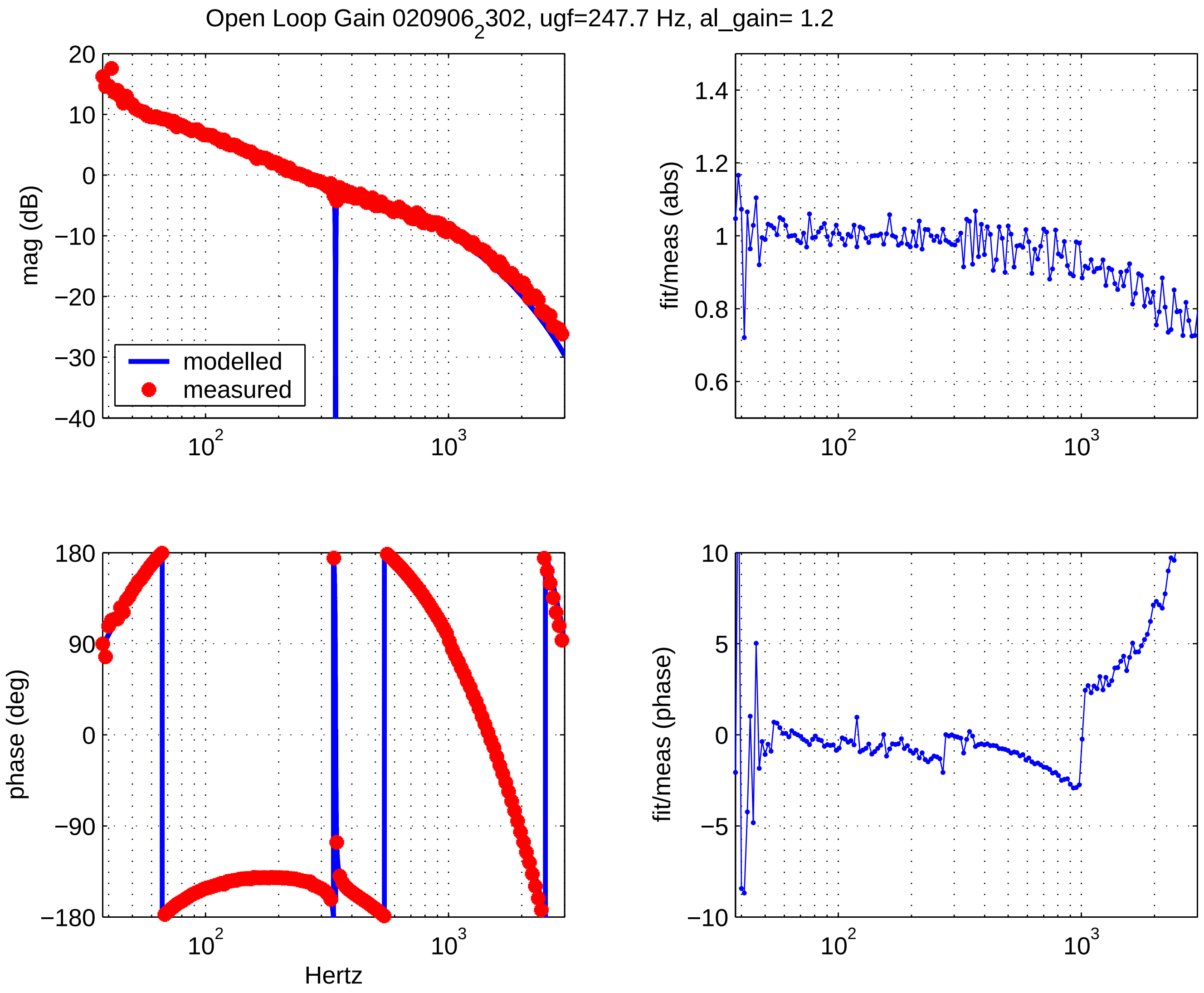


Figure 2: Comparison between swept sine and modelled gain.

```
darmtf = darmcal * tf(w0^2,[1 w0/Q w0^2]);
asqtf = asqcal * tf([1 2*pi*fc],[2*pi*fc]);
at = mybodesys(asqtf,f);
dt = mybodesys(darmtf,f);

%noise in m/rtHz
darmn = abs(dt) .* dc;
asqn = abs(at) .* asqc;
```

We then get a calibration function to use for AS_Q from the Simulink model in m/counts, and use it on the loaded spectral density to create a “true” input displacement noise, which for Rana-historical reasons we called “gwp1” in meters/rtHz:

```
Lmin2asqout=abs(mybodesys(darmsys(3,1),f));
gwp1 = asqc./Lmin2asqout;
```

The comparison of the spectra is shown in Fig3. The calibrated spectrum should coincide with DARM_CTRL at low frequencies, and with AS_Q at high frequencies, and it does!

4 Calibration Lines

Throughout the run calibration lines were injected in ETMX at 04:28 August 25th UTC (cf. LLO ELOG entry by Giaime). Initially the lines were injected at 37.75Hz and 972.8Hz but the lower line was later moved to 51.3Hz which is where it is for this calibration. The change was made at 16:09 Aug. 25th UTC at the suggestion of Rana (cf. LLO ELOG entries around that time), to allow for a smaller line with a large signal to noise ratio (the LLO spectrum has a large value near 35 Hz, so a line with good signal to noise needed to be too large).

The hope is that the power in the lines will track gain changes in the interferometer. Our starting point is the same amplitude spectral density data described in a previous section. The data are squared so that we can fit the power spectral density (counts^2/Hz). We take the error on each point in the PSD ”p” to be ”p/10” since it is calculated using 10 averages. When the points lie in the calibration peaks this results in an overestimate of the error since we are implicitly assuming Gaussian background noise, which is obviously not true for the injected signal. To compensate we use the errors from the adjacent points for the errors on the peak values. This may be a slight underestimate since it does not account for any noise contribution from the injection process.

In Figures 4 and 4 we plot the ASQ and DARM PSD in the region of each calibration line. Each distribution is fit to a Gaussian over a constant background. The parameters returned from the fit are:

- P1 = Counts rms in the peak.
- P2 = Mean frequency of the peak in Hz.
- P3 = Sigma (width) of the peak in Hz.
- P4 = Level of the background in Counts^2/Hz.

The fits were performed using the PAW utility from the CERN libraries. It was difficult to get a decent fit and as a consequence the errors on the above quantities should not be taken too seriously. The results obtained are consistent with a rough estimate made by Gaby. This coupled with the fact that different fits yielded similar results gives us confidence that the central values are robust.

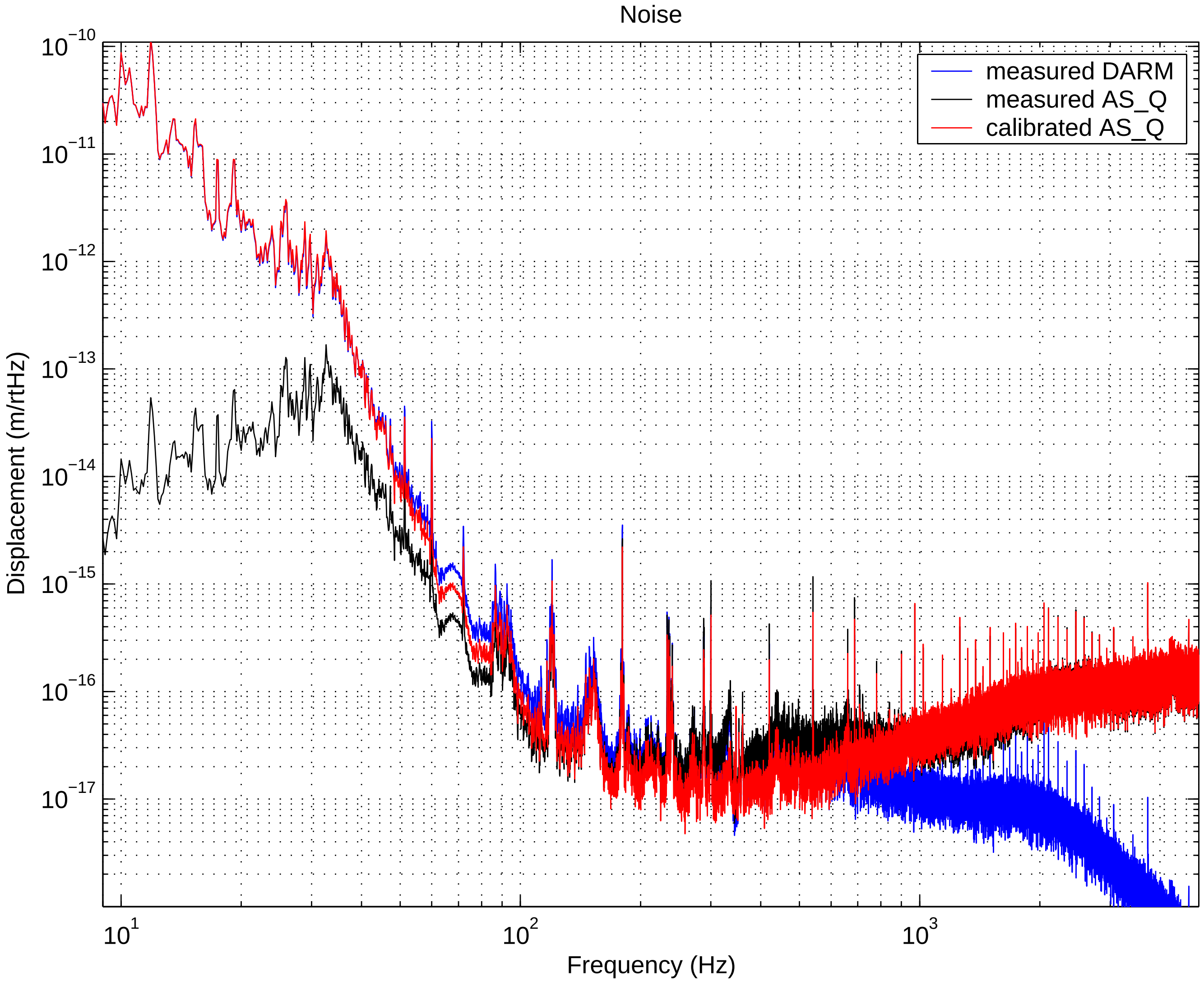
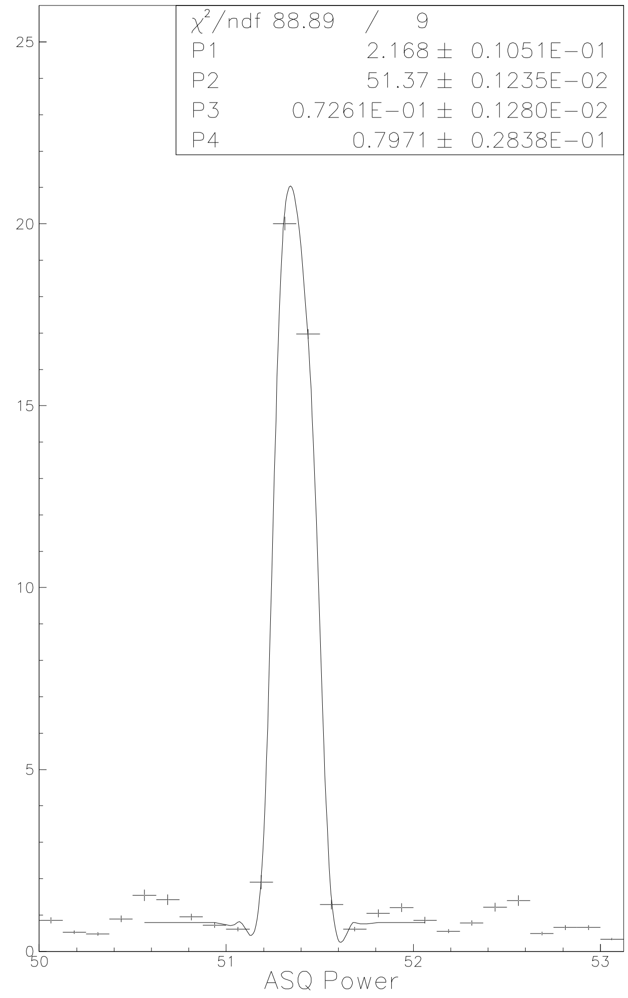
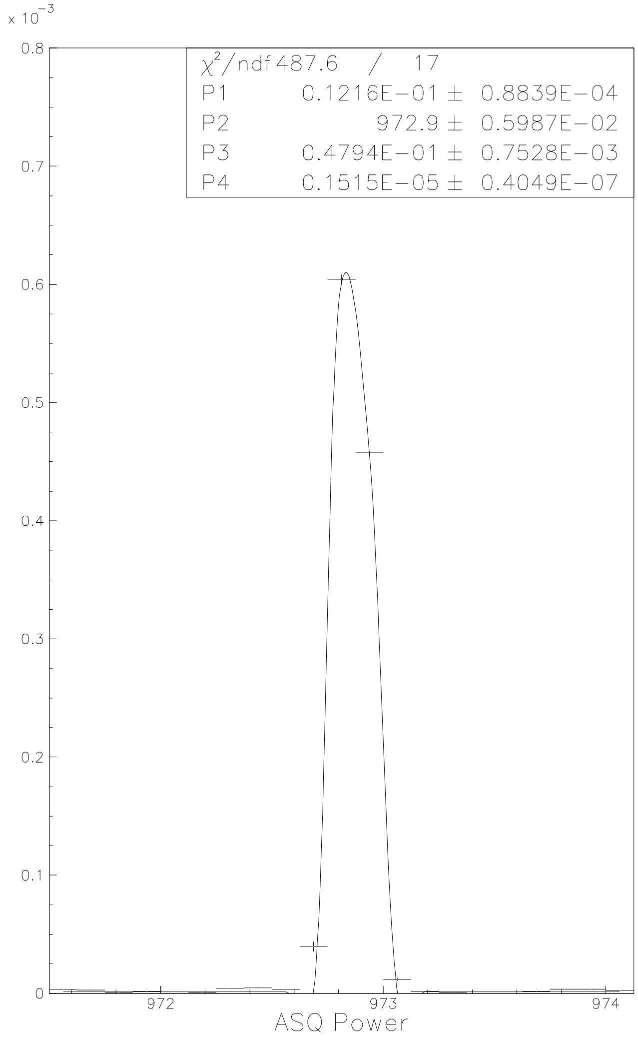
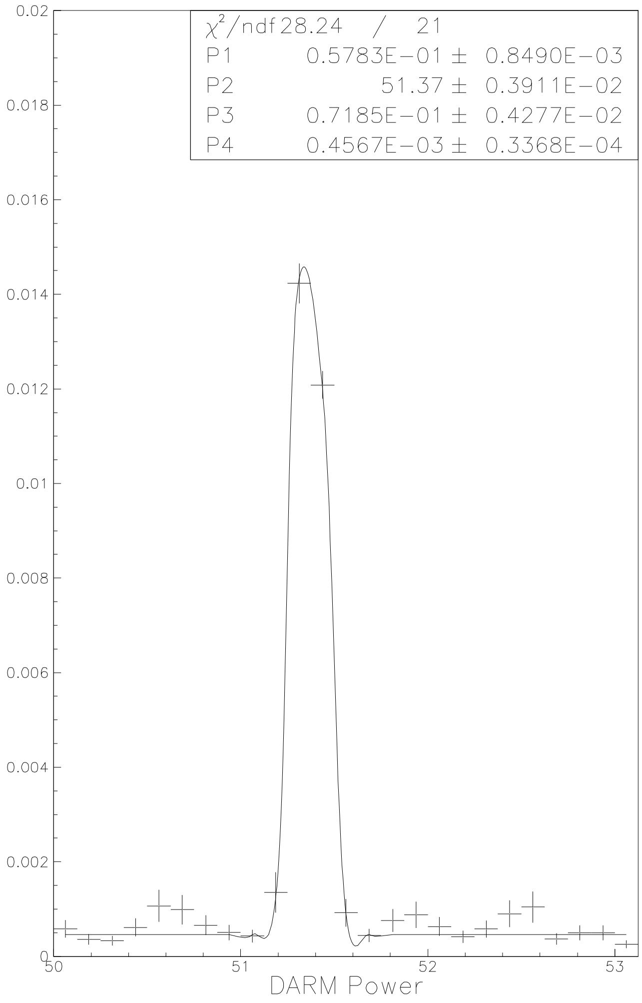
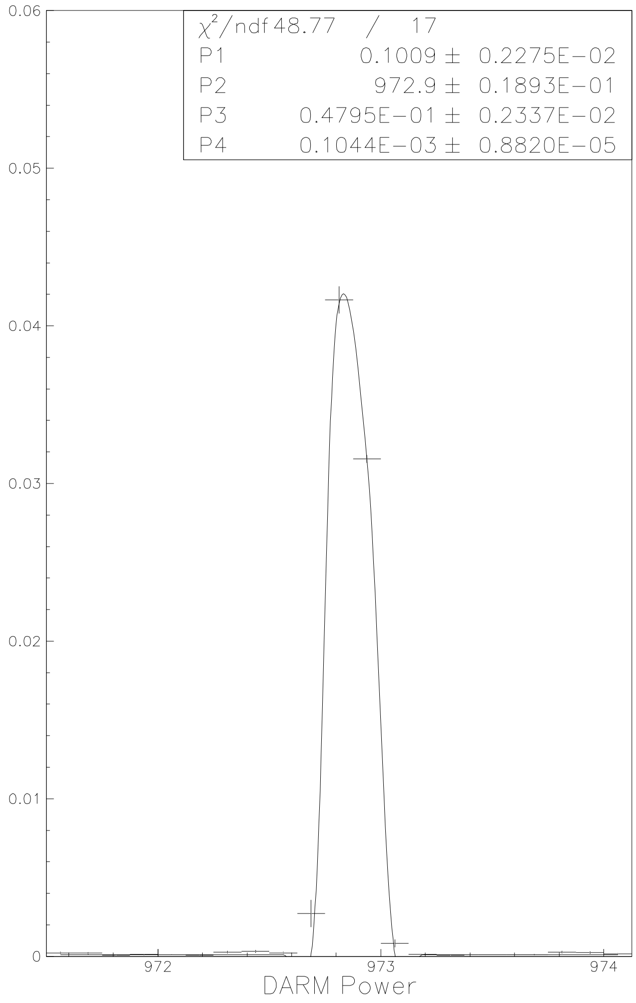


Figure 3: Spectral densities for error length (measured AS_Q), control length (measured DARM_CTRL), and true input noise (calibrated AS_Q).


χ^2/ndf 88.89 / 9
P1 2.168 ± 0.1051E−01
P2 51.37 ± 0.1235E−02
P3 0.7261E−01 ± 0.1280E−02
P4 0.7971 ± 0.2838E−01
25
20
15
10
5
0
50
51
52
53
ASQ Power


x 10^{-3}
χ^2/ndf 487.6 / 17
P1 0.1216E−01 ± 0.8839E−04
P2 972.9 ± 0.5987E−02
P3 0.4794E−01 ± 0.7528E−03
P4 0.1515E−05 ± 0.4049E−07
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
972
973
974
ASQ Power


χ^2/ndf 28.24 / 21
P1 0.5783E-01 ± 0.8490E-03
P2 51.37 ± 0.3911E-02
P3 0.7185E-01 ± 0.4277E-02
P4 0.4567E-03 ± 0.3368E-04
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
50
51
52
53
DARM Power


χ^2/ndf 48.77 / 17
P1 0.1009 ± 0.2275E-02
P2 972.9 ± 0.1893E-01
P3 0.4795E-01 ± 0.2337E-02
P4 0.1044E-03 ± 0.8820E-05
0.06
0.05
0.04
0.03
0.02
0.01
0
972
973
974
DARM Power

5 Results

Finally, we present a calibration, valid for Sept 06, 22:52 UTC (when a clean locked section started) to 23:02 (when the swept sine started). Notice that this is NOT a "science" segment, since calibration procedures were being made. However, the alignment and sensitivity during the time of observation for noise spectra were typical.

1. **Calibration function Cb(f) for AS_Q, in strain/counts.**

   This is obtained from the Simulink model, for a frequency vector evenly spaced from 0Hz to 2048Hz, with 1/64 Hz spacing:

   ```
   S1darm
   S1darm_08
   [A,B,C,D] = linmod('S1darm_08');
   darmsys = ss(A,B,C,D);
   fd=[0:1/64:2048];
   calib=1./mybodesys(darmsys(3,1),fd);
   calib=calib/4e3;
   ```

   The output is plotted in Figure4, and the frequency triplets (frequency, magnitude and phase) written in the file ASQCalibration.txt.

2. **Sensing function C(f) for AS_Q, in strain/counts.**

   This function is the same calibration function calculated above, except that now the loop is open. This function involves essentially just the cavity pole transfer function. To keep consistency and include the effects of the ADC time delay included in the model, we obtain the sensing function with another Simulink model that has the servo links broken. The function is plotted in Figure5, and the frequency triplets written in a file ASQSensing.txt

3. **Open Loop Gain H(f)**

   We calculate the open loop gain, again from the Simulink model, that is related to the calibration and sensing functions through $Cb(f) = C(f)/(1+H(f))$. This function is calculated as the forward loop function darmsys(2,2) divided by the closed loop function darmsys(1,1). Since this function has a very large dynamic range, there are numerical errors visible at low frequencies. We can better estimate the function from the model by using the analytical expressions known for each block, but for consistency we use the same method as used when fitting the alignment gain to the loop gains. The (noisy) curve is plotted in Figure6, and the triplets (always for the same frequency vector) saved in the file OpenLoopGain.txt

4. **Amplitudes of calibration lines in AS_Q and DARM_CTRL.**

   As described in the previous section, the amplitudes of the calibration lines in AS_Q were 2.16 ± 0.01 counts rms (51.37 Hz) and 0.012 ± 0.001 counts rms (972.9 Hz); and in DARM_CTRL, $0.058 \pm .001$ counts rms (51.37 Hz) and $0.101 \pm .002$ counts rms (972.9 Hz).

This output characterizes the calibration of LLO at a given time (09/06/02, 22:50-23:00 UTC). A procedure described elsewhere (?) will be used to propagate this calibration to the rest of S1.

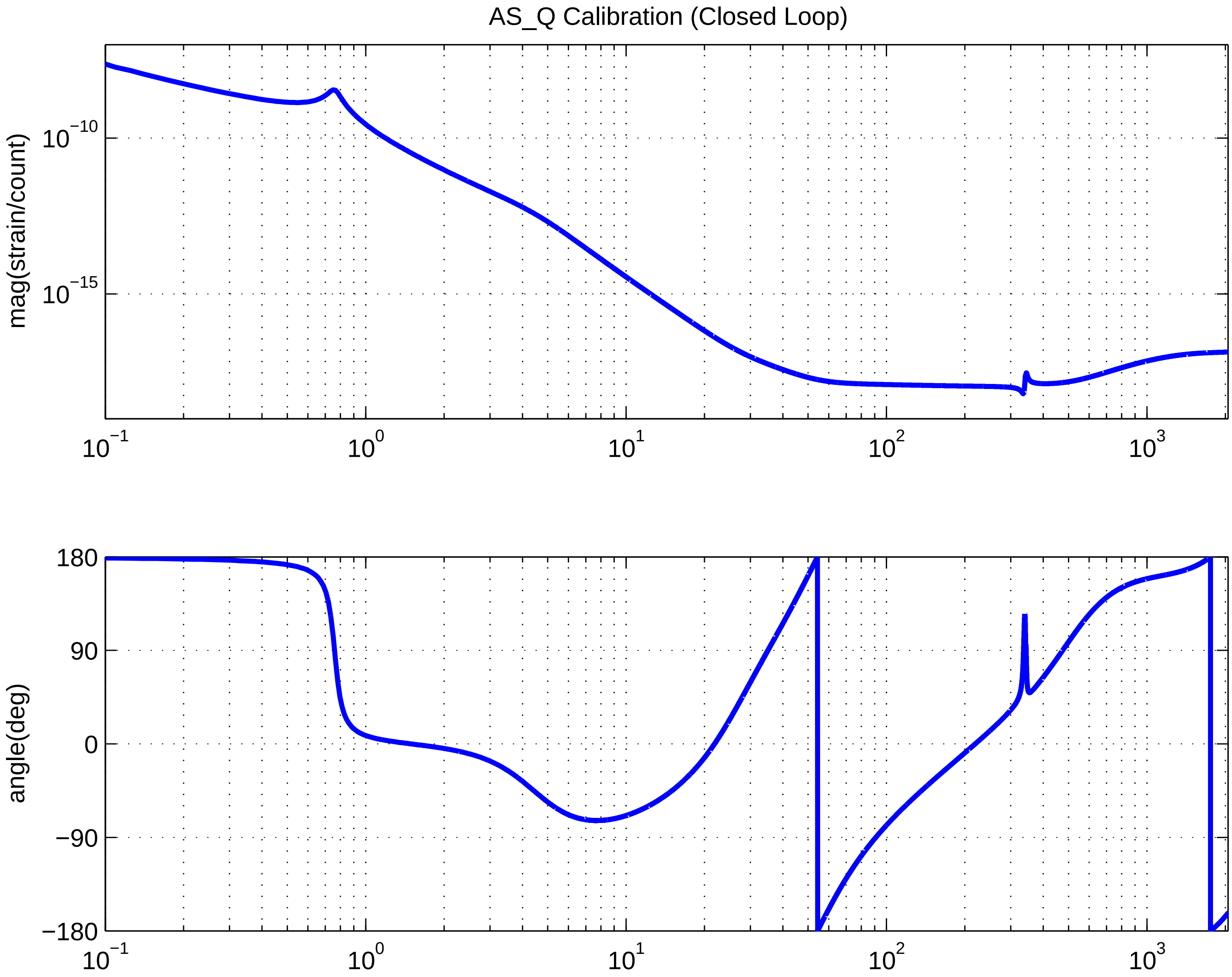


Figure 4: Calibration function for ASQ.

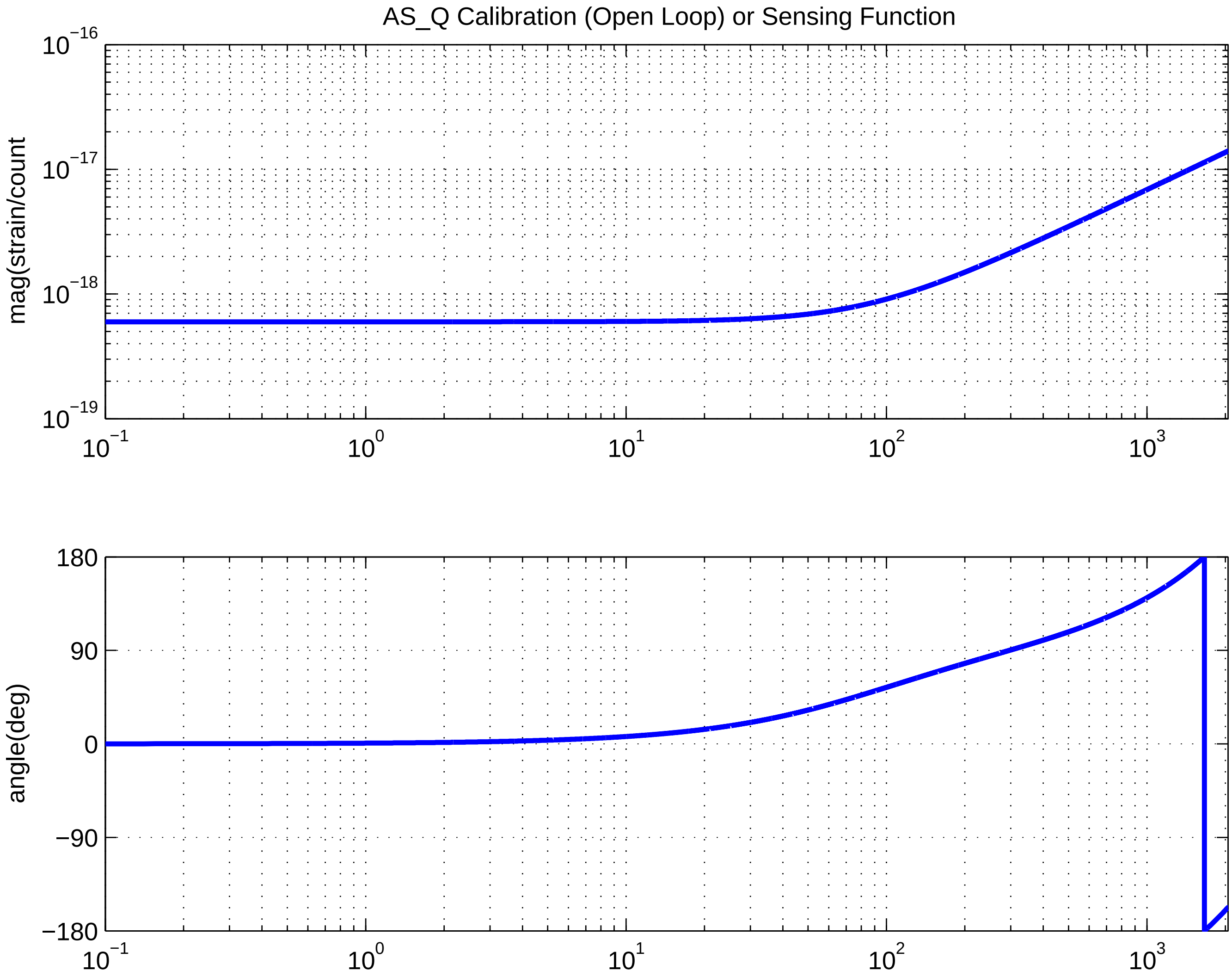


Figure 5: Calibration function for ASQ.

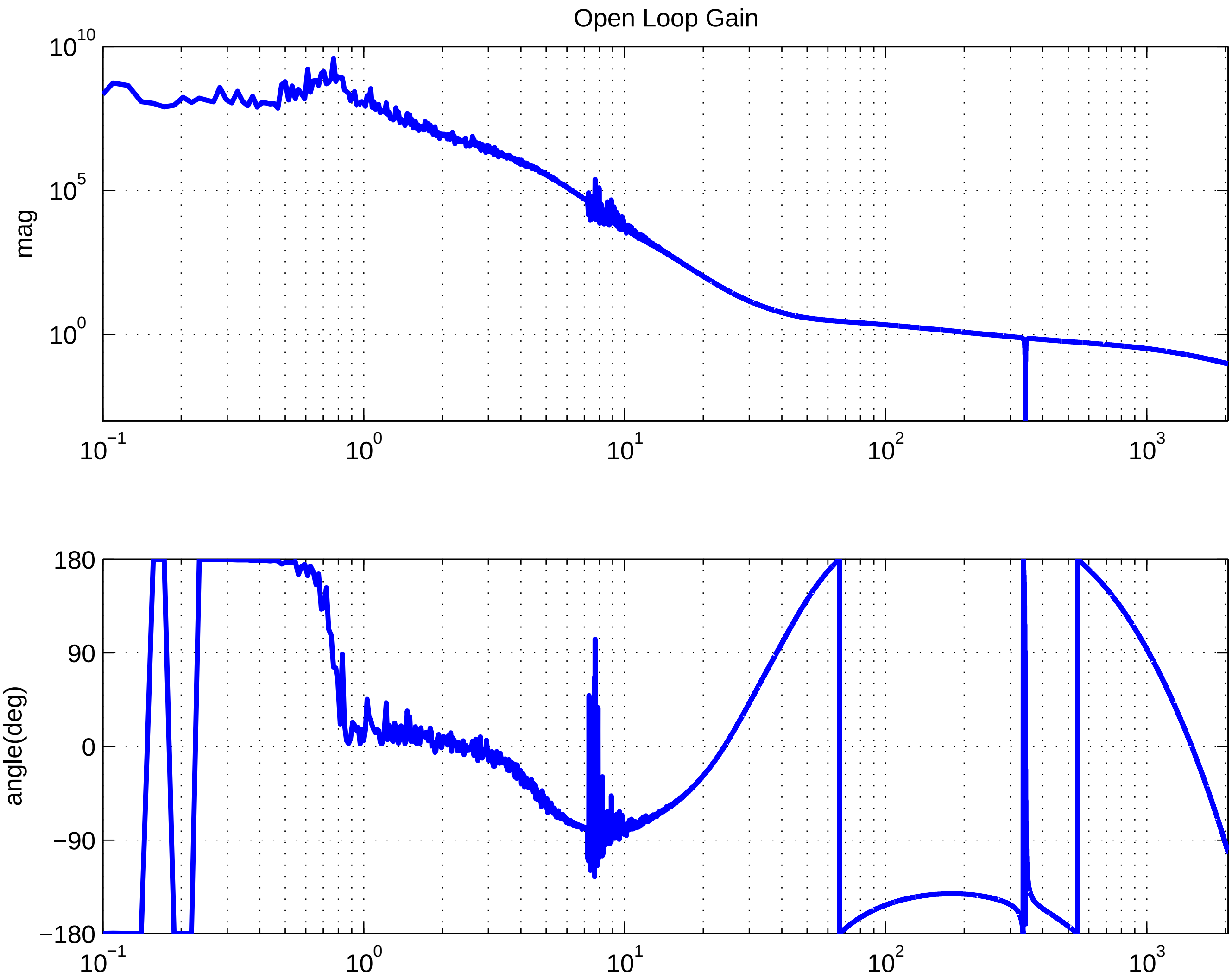


Figure 6: Open Loop Gain from S1darm_08.

Appendix: List of files

The following files can be found in
`http:\\www.phys.lsu.edu\gonzalez\S1Calibration :`

- S1calibOct4.tex, *.eps figs called for, and S1calibOct4.pdf: this document.
- Parameters and filters Matlab file: S1darm.m
- Simulink models: S1darm_08.mdl and OpenS1darm.mdl
- Data files:
  - Swept Sine: 020906_2302coh.txt (coherence) and 020906_2302.tf.txt (transfer functions). The numbers are for for AS_Q/DARM_ERR and DARM_CTRL/DARM_ERR.
  - Noise spectra (with calibration lines): 020906_22_53_cal.mat (f,dc,and asqc vectors, for frequency in Hertz, DARM_CTRL and AS_Q in counts/rtHz).
  - Calibration results: ASQCalibration.txt; ASQSensing.txt; OpenLoopGain.txt.
- Matlab programs:
  - CompareMeasGain.m : Compare meaured gains and modelled gains with Simulink.
  - np.m: Plot noise spectral densities of error and control lengths, and calibrated AS_Q.
  - FinalCalib.m: produces plots and text files mentioned in the previous section about results.
- PAW macros (for calibration lines fits): fit_asq.kumac, fit_darm.kumac; and FORTRAN fit function gp0.for