Tilt Improvement with a CRS on Stage 2 of the ISI
T2200393

Brian Lantz, Michael Ross

Nov. 2022

1 Introduction

The Cylindrical Rotation Sensor (CRS) is a new inertial sensor under development at the Univ. of Washington. We make a simple estimate of how much we could lower the tilt motion on stage 2 of the BSC-ISI if we include this sensor. The control model is simple, we include an 'ideal' CRS in the stage 2 control with reasonable blend filters. We do not follow up with a calculation of the horizontal performance. We do not consider more sophisticated models which use the stage 2 tilt to stabilize stage 1. We see that the tilt can be improved significantly - about 10x from 0.1 Hz to 3 Hz.

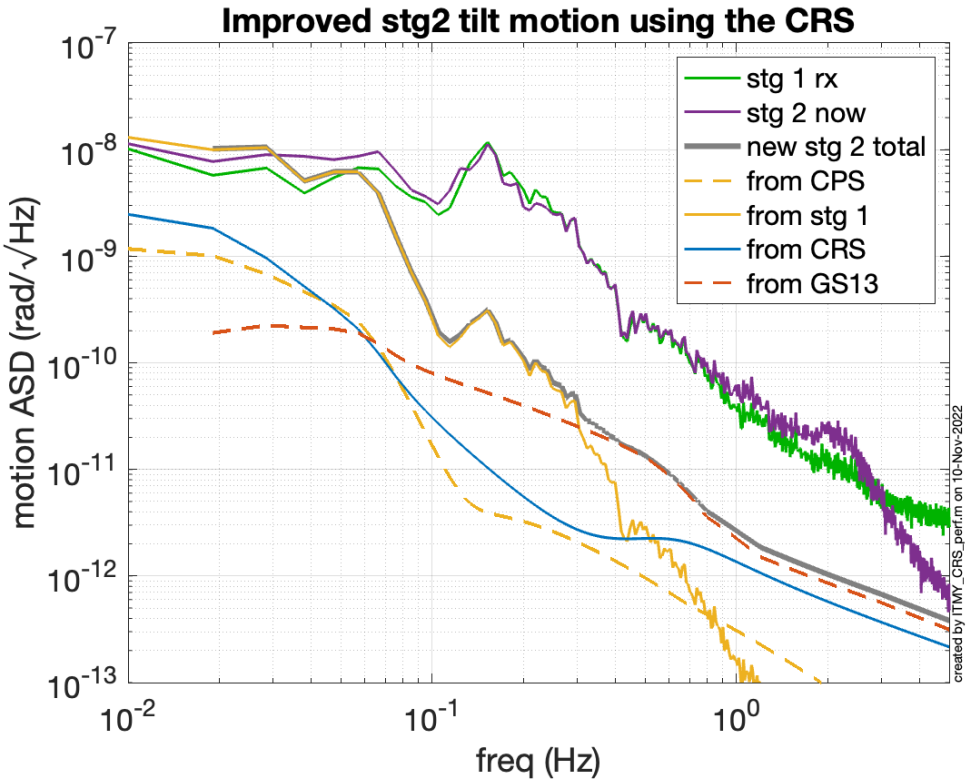

Figure 1: Final Modeled performance of stage 2 tilt. The purple line is the current tilt of stage 2 of LHO:ITMY on a relatively quiet day. The grey line is the modeled tilt performance which could be achieved if a CRS were added. Even on quiet days the performance could be dramatically improved from 0.1 to 3 Hz.

2 Noise Inputs

The calculations are all included in the seismic SVN at .../Common/Documents/T2200393. The motion inputs are drawn from the rotation study T2100273 'Design for new rx/ry blend filters for Stage 2 of the BSC-ISI'. The tilt motion we pick for this calculation is the tilt of ITMY at LHO. This ISI is one of the best performing platforms at low frequency. Figures 2 and 3shows the tilt of stage 1 (the input motion for stage 2) compared to the ITMY motion at LLO at 2 different times. We pick the lowest input motion because this represents the least improvement possible given a fixed sensor noise. For larger input motions (large microseism, earthquakes, wind) a more aggressive tilt blend would be appropriate and more isolation performance is likely.

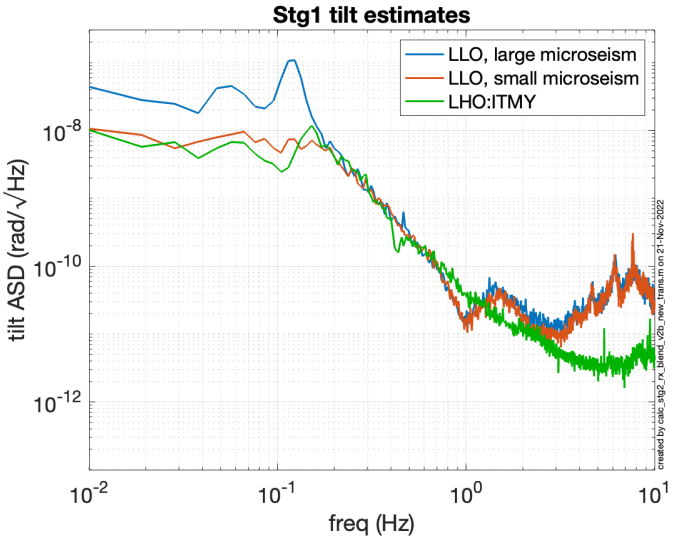

Figure 2: Stage 1 RX tilt motion of ITMY at LHO and LLO. The LHO motion (green) will be used for the rest of this calculation. This is only an estimate below about 0.1 Hz. Adapted from T2100273.

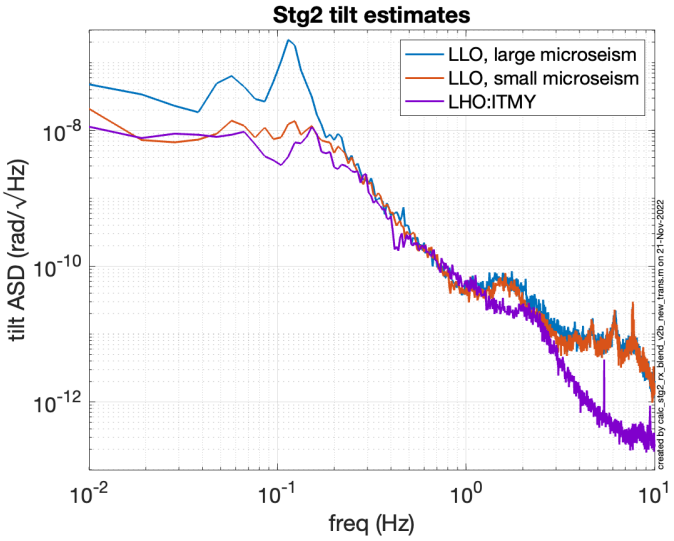

Figure 3: Stage 2 RX tilt motion of ITMY at LHO and LLO. The LHO motion (purple) will be used for the rest of this calculation. This is only an estimate below about 0.1 Hz.Adapted from T2100273.

The noise of the GS-13, the fine CPS on stage 2, and the geometric correction for the rx degree of

freedom are included in SEI_sensor_noise.m. The noise of the 'ideal' CRS is drawn from SEI log 1969 The comparison of the 3 sensor noises and the motion of stage 1 and stage 2 is shown in figure 4. For this estimate, stage 2 rx motion is only damped, as was the configuration during O3. is The CRS should be able to measure the damped motion from 10 mHz to 4 Hz.

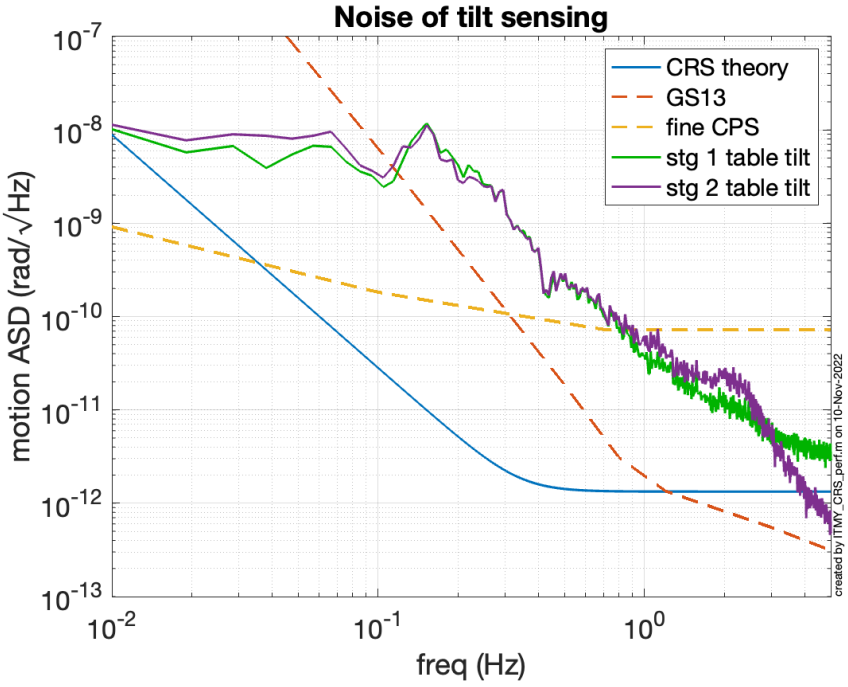

Figure 4: Noise of the stage 2 tilt sensors compared with low noise motion of the ISI. The blue CRS noise is the theory limit. The stage 1 motions are from T2100273, and are just estimates below about 0.1 Hz.

3 Sensor Blending

In figure 4 we see that the blend frequencies should be around 40 mHz and 1 Hz. We make these two blends separately, and then combine them. This could be further optimized and also adapted for larger input tilt conditions

The low blend (blend 1) is shown in figure 5. The goal is to give decent performance at the microseism and make sure the noise of the CRS stays well below the 10 nrad/rtHz level below 30 mHz.

The design of the blend for the CRS and the GS-13 is slightly different. Above the blend we try to keep the noise very close to the GS-13 noise floor above a few Hz because the platform goes to the GS-13 noise floor by about 5 Hz. Below the blend frequency we are trying to get the GS-13 noise below the CRS by the microseism. In the performance plot we see that the GS-13 noise limits from about xxx to yyy Hz, but at the microseism it is below the CRS noise. Even with a CRS we STILL would benefit from improved GS-13s.

The final performance is modeled by combining both blends and all the sensor noise. We assume very large loop gain. In the T2100273 calculation, we saw that the loop gain did not limit the performace below 5 Hz. That is why these plots stop at 5 Hz.

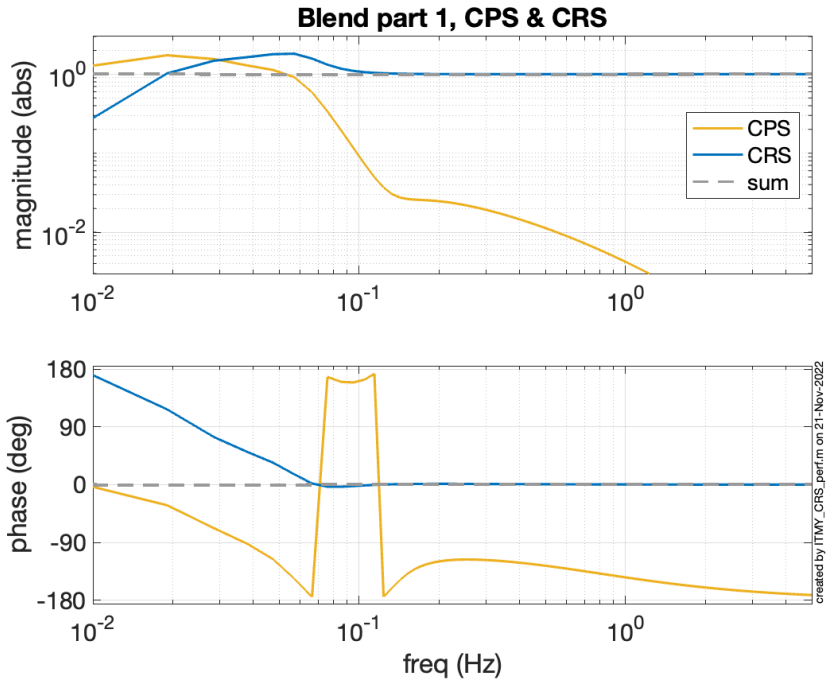

Figure 5: Low frequency blend to combine the CPS and the CRS

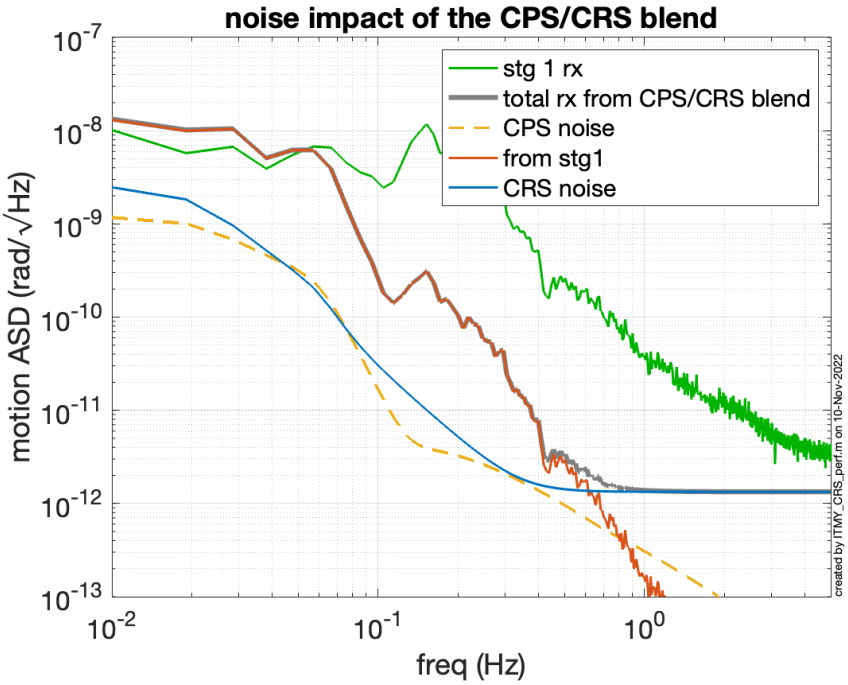

Figure 6: Performance of the CPS/CRS blend

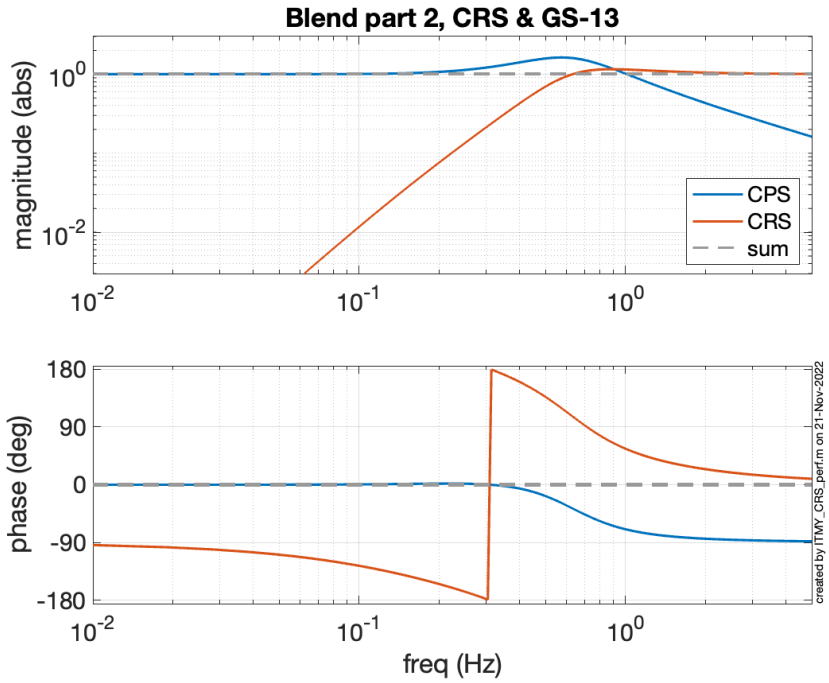

Figure 7: High frequency blend to combine the CRS and the GS-13

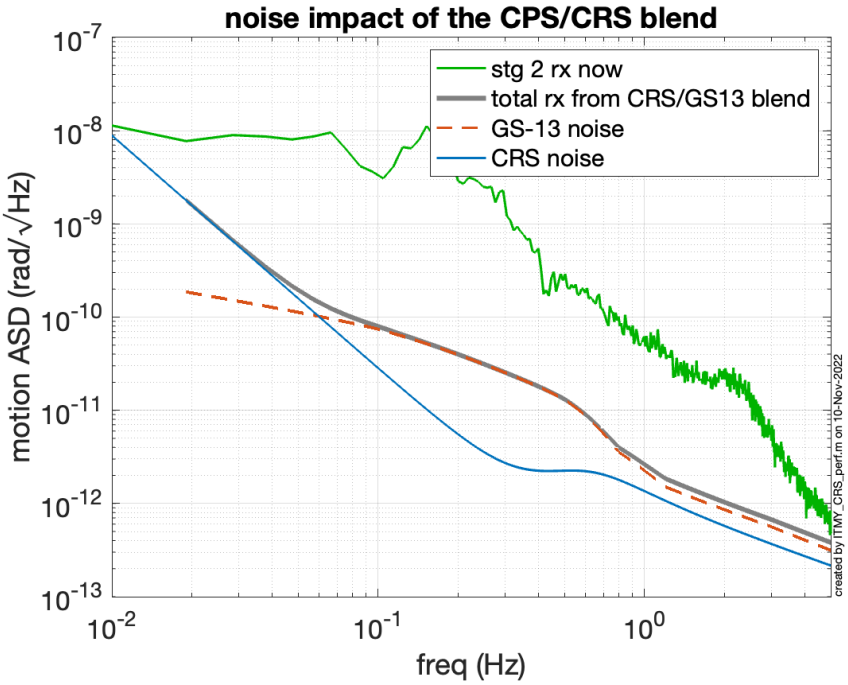

Figure 8: Performance of the CRS/GS-13 blend

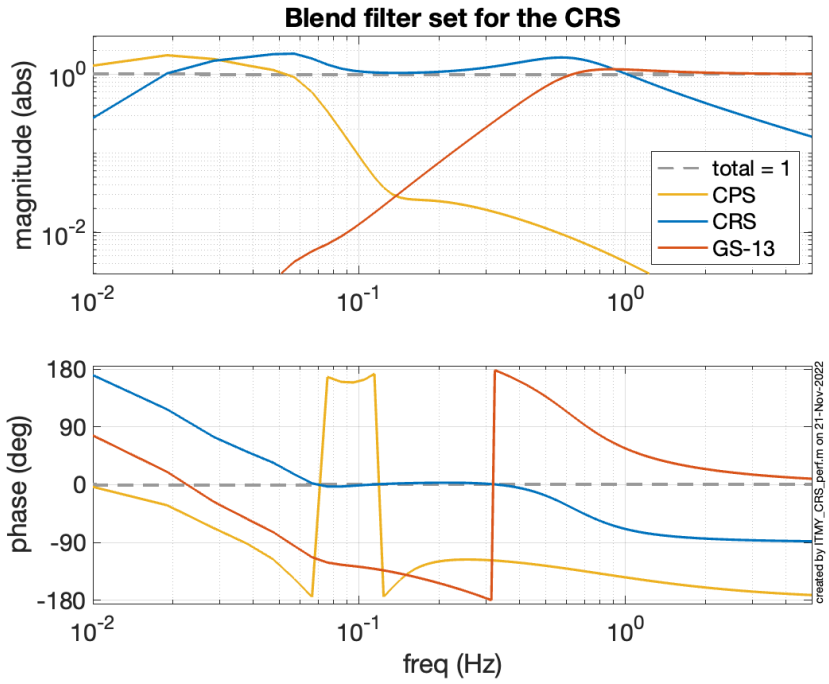

Figure 9: Final Blend filters

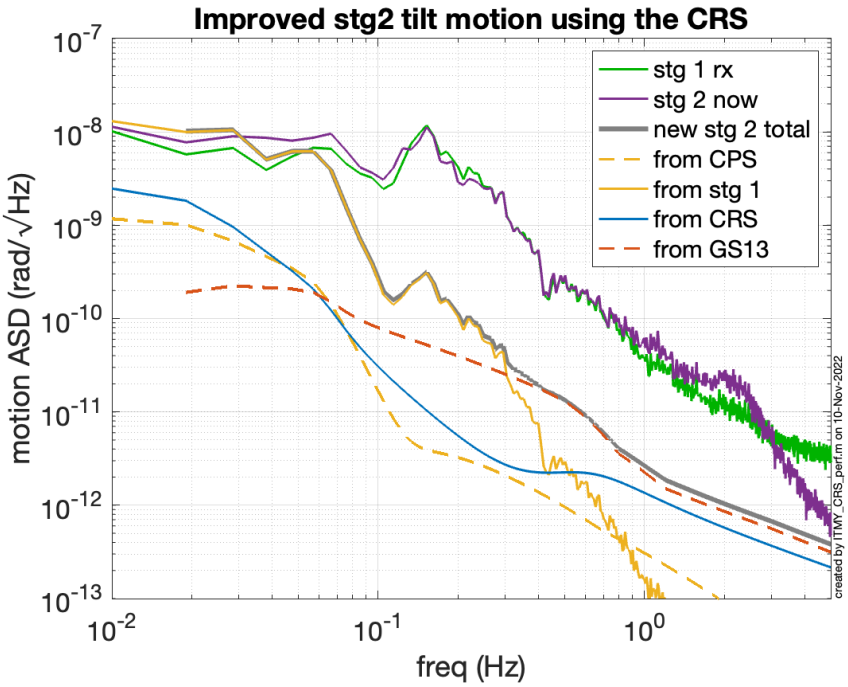

Figure 10: Final Modeled performance. This plot is the same as figure 1 . Isolation begins around 70 mHz. The low frequency cut-off is designed to limit the noise injection of the CRS into tilt at 10 mHz and below. For noisy days (wind, microseism, earthquakes) this blend should probably be pushed down. The microseismic tilt is reduced by about a factor of 30, limited by the CPS/CRB blend. Above 400 mHz the performance is limited by the GS-13 noise, along with loop gain starting above a few hertz.

4 The blend code

All the figures and the blending code is in ITMY_CRS_perf.m. The code for the filters is included below.

```
%% low frequency blend
blnd1.f0 = 0.025;
blnd1.LP_pole = zpk(-2*pi*[],-2*pi*[blnd1.f0, 30 *blnd1.f0], ...
    2*pi*blnd1.f0 * 2*pi*30*blnd1.f0 );
% the pole around 1 Hz is to keep below the GS13, later
blnd1.LP_ellip = myellip_z(2.5*blnd1.f0, 2, 1, 12, 4)*10^(1/20);
blnd1.LP_proto = blnd1.LP_pole * blnd1.LP_ellip;

blnd1.HP_proto = zpk(-2*pi*[0 0 0],...
    -2*pi*[0.2*blnd1.f0, 0.5*blnd1.f0, blnd1.f0],1);

[blnd1.LP_zpk, blnd1.HP_zpk]=maketruecomplements_tol(...
    blnd1.LP_proto, blnd1.HP_proto,true,.02);

%% high frequency blend

blnd2.f0 = 0.8;
blnd2.LP_proto = zpk(-2*pi*[],-2*pi*[blnd2.f0],2*pi*blnd2.f0 );
blnd2.HP_proto = zpk(-2*pi*[0 0 0],...
    -2*pi*[0.3*blnd2.f0, 0.5*blnd2.f0, blnd2.f0],1);

[blnd2.LP_zpk, blnd2.HP_zpk]=maketruecomplements_tol(...
    blnd2.LP_proto, blnd2.HP_proto,true,.02);
```

5 CRS noise curve

```
function noise=CRSTheoryNoiseModel(f)
% theory curve. Michael Ross, Sept 2022
    Q=1e3;
    f0=17e-3;
    r=15e-2;
    kb=1.380e-23;
    T=293;
    I=0.0941;

    Tf=1./abs(f.^2./(f.^2-1i*(f0^2/Q)-f0^2));

    readout=Tf.*2e-13/r;
    thermal = Tf.*sqrt(4*kb*T*f0^2./(I*2*pi*f.*Q.*((f.^2-f0^2).^2+f0^4/Q^2)));

    noise=sqrt(readout.^2+thermal.^2);
end
```