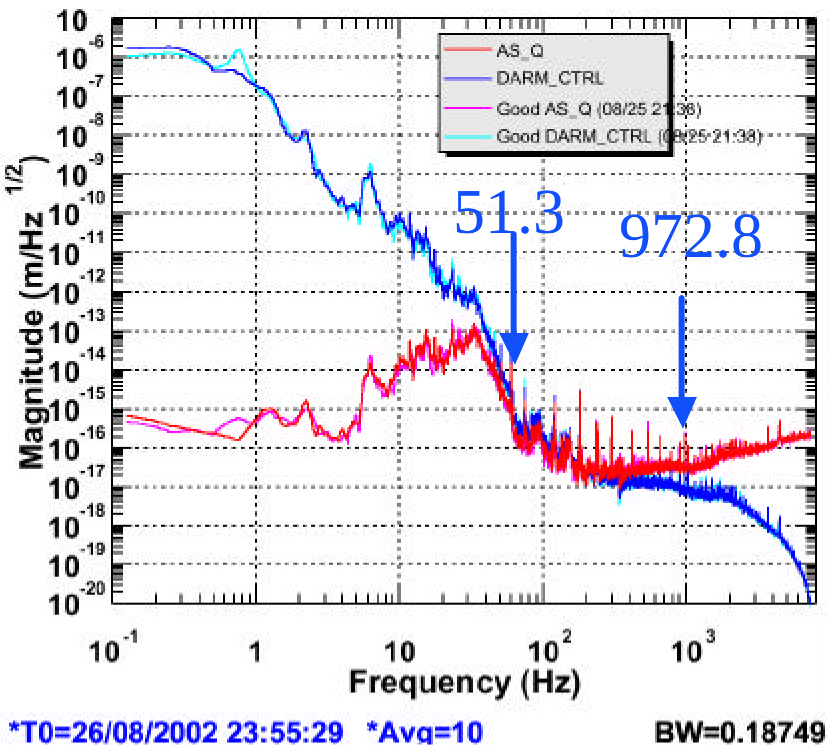
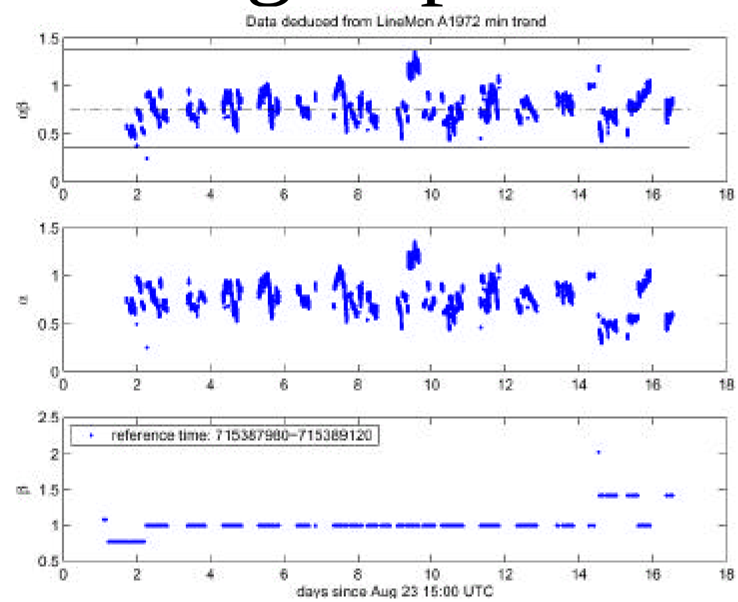
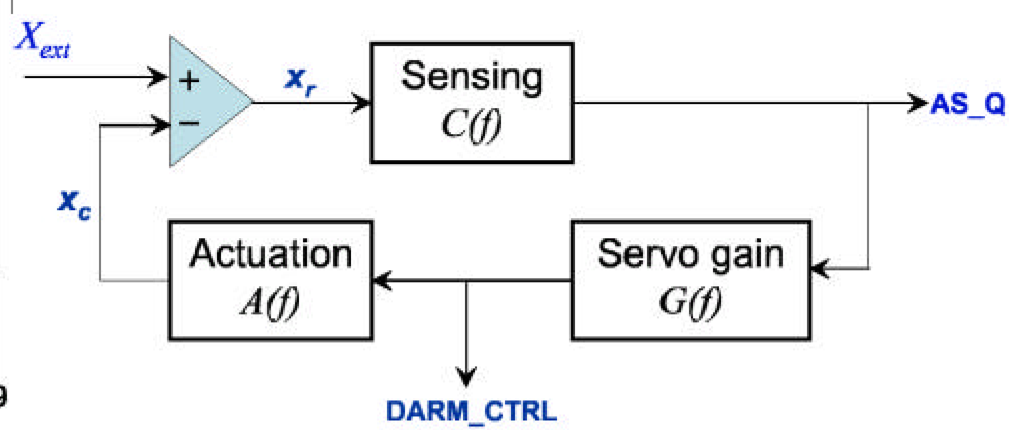
Use of detector calibration info in the burst group

$$AS\_Q = X_{ext}\frac{C(f)}{1+H(f)} \rightarrow X_{ext}\frac{\alpha C(f)}{1+\alpha\beta H(f)}.$$







Do we need calibration information for the burst search?

- The burst analysis pipeline uses 3 (and growing) Event Trigger Generators (ETG's, aka DSOs)
  tfclusters, slope, power
- They all search for excess power, in T-F plane or in a filtered time series.
- We don't need calibrated info for this;
  we're not doing matched filtering of any kind.
- In fact, the first thing we do is HPF and whiten the data (in datacond).
- We rely on coincidence between 2 or more detectors for detection confidence.
- Excess power detected in 2 or more IFOs in time coincidence must be consistent in terms of waveforms, frequency band, peak amplitude (strain).

Use of Calibration information

- We need calibration information for (at least) TWO things:
  - Evaluating efficiency for burst waveforms as a function of their peak or rms strain amplitude.
  - Requiring consistency of the waveforms and amplitudes between 2 or more detectors.
- Only the first of these is currently implemented for the S1 analysis; but post-coincidence consistency checks are high priority for the S2 analysis!

Calibrated power for S1 burst simulations (P. Sutton)

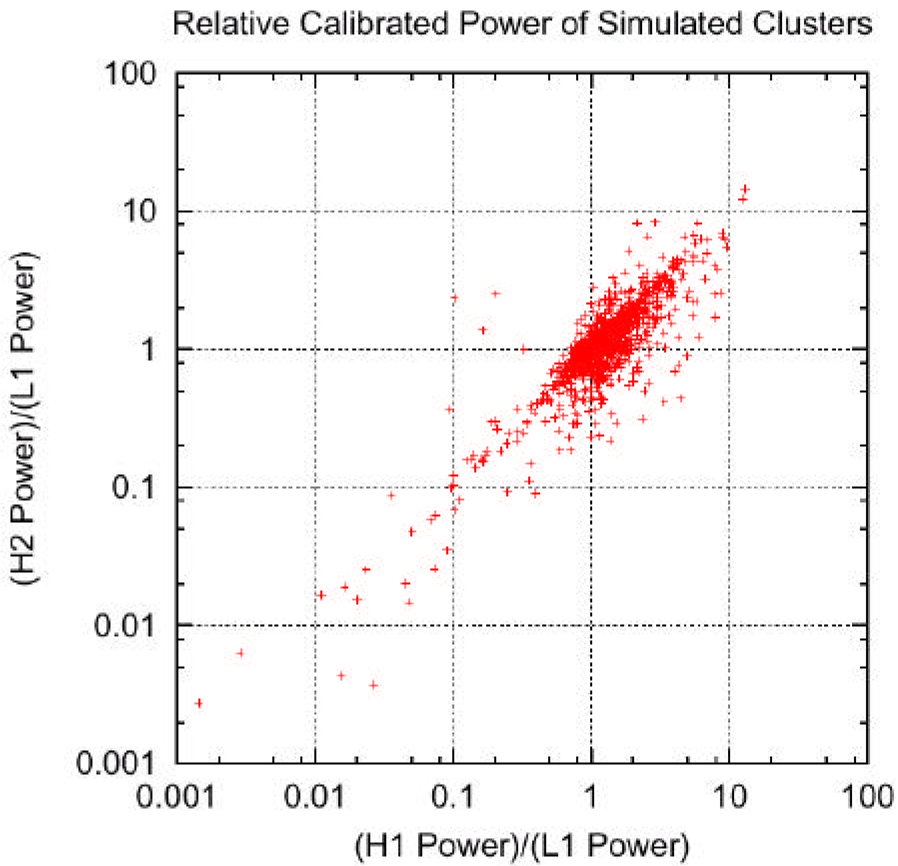


Figure 2: Calibrated power of clusters seen in H1 and H2 relative to L1.

Patrick Sutton has begun to look at an "amplitude" cut for S1, using simulated injections.

Evaluating efficiency for burst waveforms

- We inject short (< 1 sec duration) waveforms into the data streams of each IFO.
- Because the waveforms are simple, we choose to do this in datacond, before the data ever makes its way into the search algorithm (ETG) in the wrapperAPI/mpiAPI.
  - » As far upstream in the pipeline as possible
  - » The ETG doesn't even know what it's getting
  - » most (all) of the other groups apply the calib info in LAL code;
    if we do it differently, must ensure that we're doing the same thing
- Philip Charlton has implemented a datacond action, respfilt(), which reproduces what is done in LAL code
  - » Checked against independent Matlab code
- So, we generate a burst (GA, SG, ZM, ..) in datacond, as h(t)
  - » Pass through respfilt() to convert to AS_Q counts
  - » Add to the raw data, whiten and HPF as usual
  - » Send it on the the wrapperAPI for event trigger generation

Datacond action respfilt() from Philip Charlton

•***y = respfilt(x, response, sense, alphas, gammas [, direction])***
Contruct a transfer function from calibration data and apply it to a time-series.

•*Input parameters:*

•***x*** - a real TimeSeries.

•***response*** - a FrequencySequence<complex<float> > representing a response function.

•***sense*** - a FrequencySequence<complex<float> > representing a sensing function.

•***alphas*** and ***gammas*** - TimeSeries<complex<float> >s representing calibration measurements taken over a period of time.

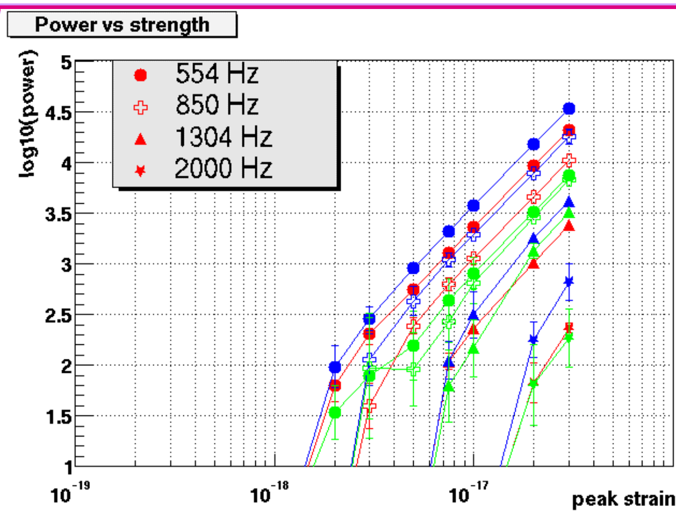
•***direction*** (optional) - a Scalar<int> flag indicating direction in which to perform the transformation. A value of 0 indicates that the input is transformed using the constructed transfer function, while a value of 1 indicates that the inverse of the transfer function is used. The default is 0.

•*Result:*

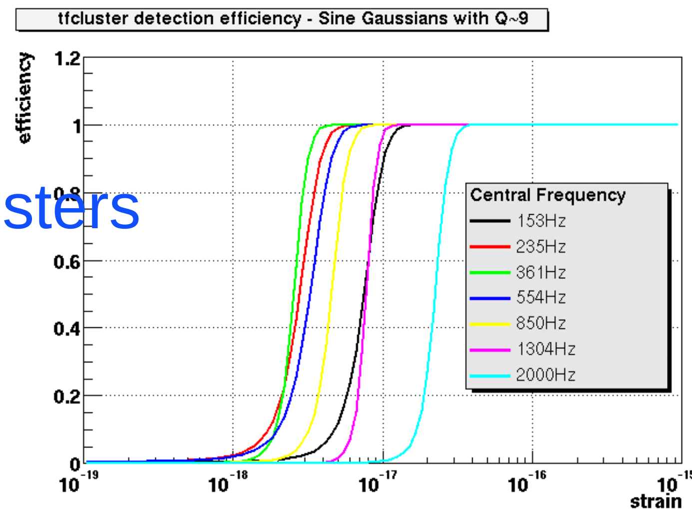
•***y*** - a real TimeSeries with the same precision, size and meta-data as ***x***, containing data obtained by applying the transfer function to ***x***.

• This action uses calibration information from a frame file to construct a transfer function, which is applied to the input time-series.

Sine-Gaussians - efficiencies



tfclusters



Simulations with calibrated SG's

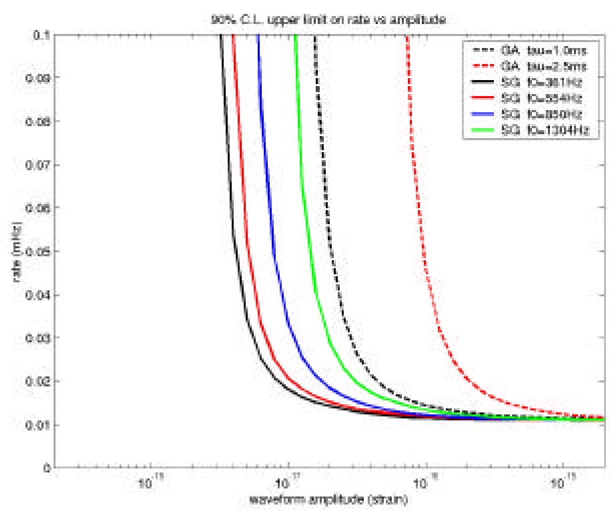
? ETG power vs peak strain

? apply threshold

? efficiency vs peak strain

? event rate = detected rate / efficiency

? event rate vs peak strain



Is our primary result for S1

Efficiency systematics

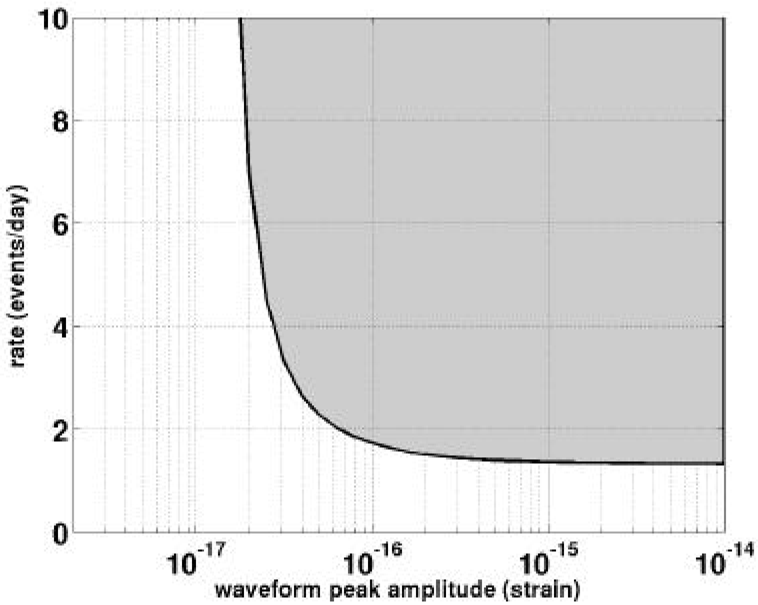
- Uncertainty in the detector response function is one of, or *the*, biggest uncertainties in our analysis
  - “DC” calibration in nm/ct
  - Frequency dependence ($C(f)$, $H(f)$)
  - Time dependence not monitored by the calibration lines
- If we have some estimate of the uncertainties in these, we can run simulations to propagate the uncertainty to our final result (laborious, but straightforward)
- We rely on the calibration group for these estimates!

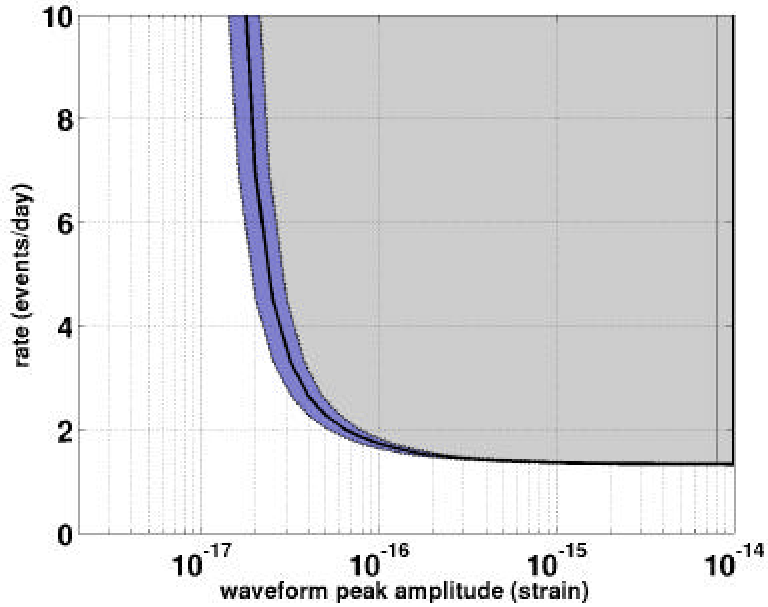
S2-LLO 4k (L1) fully recycled ifo, details.

Current ETM calibrations:

L1:LSC-ETMX_OUT:	(0.39 +/- 0.02) nm/count
L1:LSC-ETMY_OUT:	(0.37 +/- 0.03) nm/count

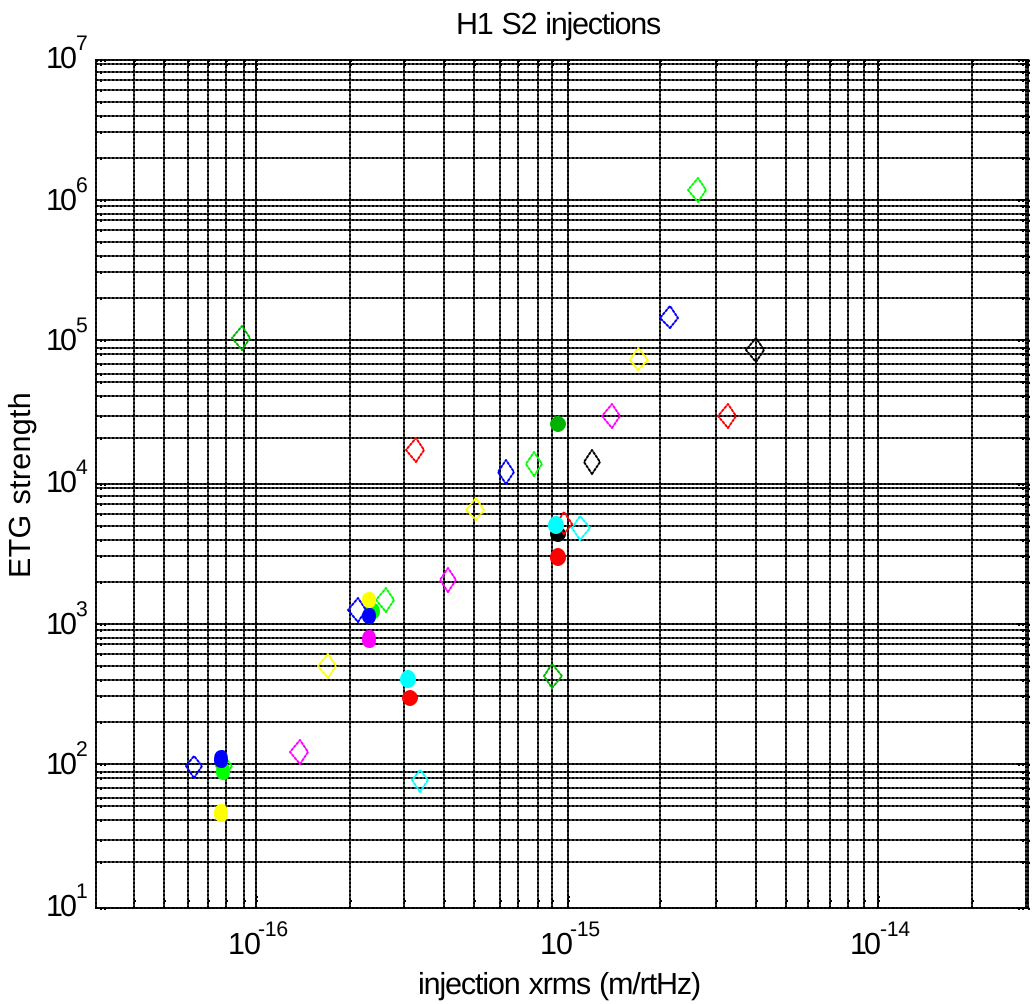
Calibration uncertainties feed directly into final result





Event rate vs peak strain ➡ with ~ 10% calib uncertainty

Comparison between HW and SW injections as a test of calibration



- Comparison currently only available for latest round of intra-run injections into H1.
- Solid points = HW
- Open diamonds = SW
- Find 45^{o} line connecting points and diamonds of same color (f_0)?
- That's qualitative evidence that HW and SW injections with same (nominal) xrms are found by tfclusters with same strength.
- Much more work, statistics, etc, required to establish this quantitatively! INSPIRALS.

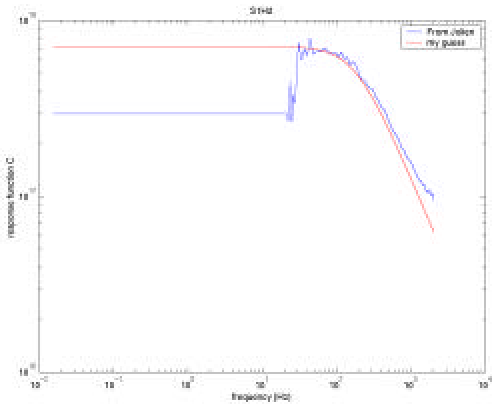
A couple of issues that have complicated the S1 analysis

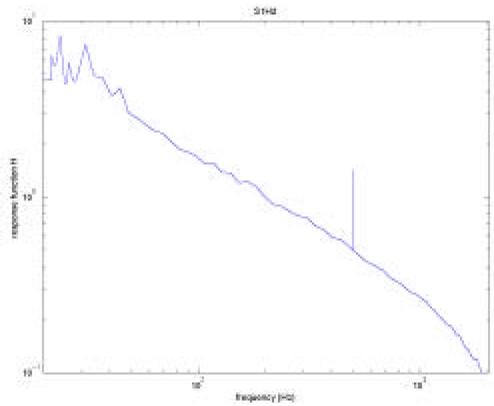
- Different kind of calibrated info from LHO and LLO
- Calibration info not available for full “good” S1 triple-coincidence data

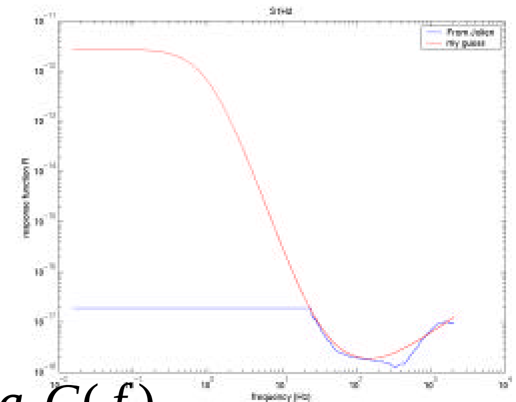
Different calib info from LHO (raw data) and LLO (model)

C(f) ***H(f)*** ***R(f) = 1/T(f)***

H2

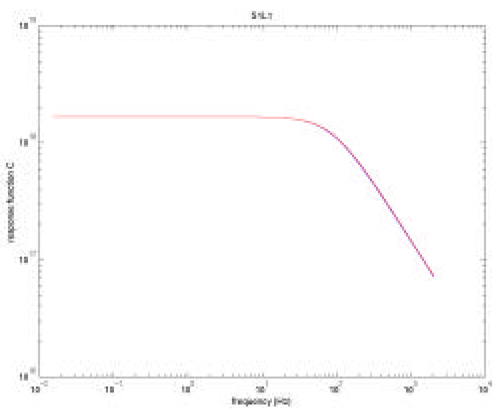


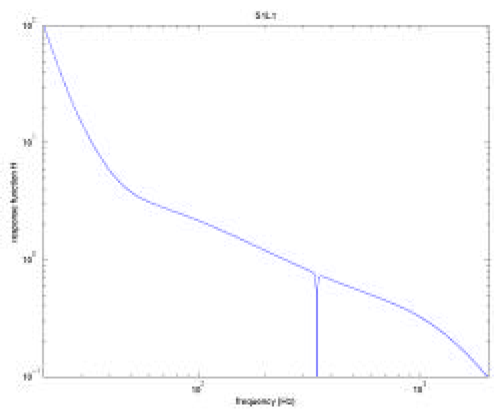


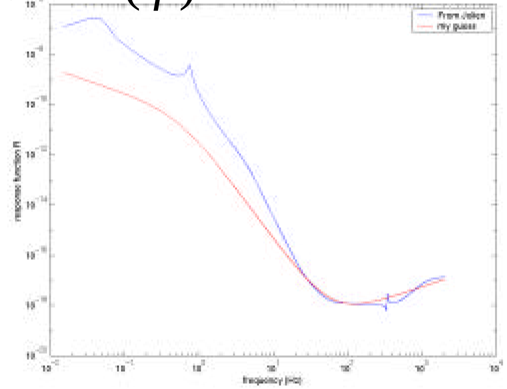


$$ASQ(f) = h(f)T(f) = h(f)/R(f) = h(f)\frac{a\;C(f)}{1+ab\;H(f)}$$

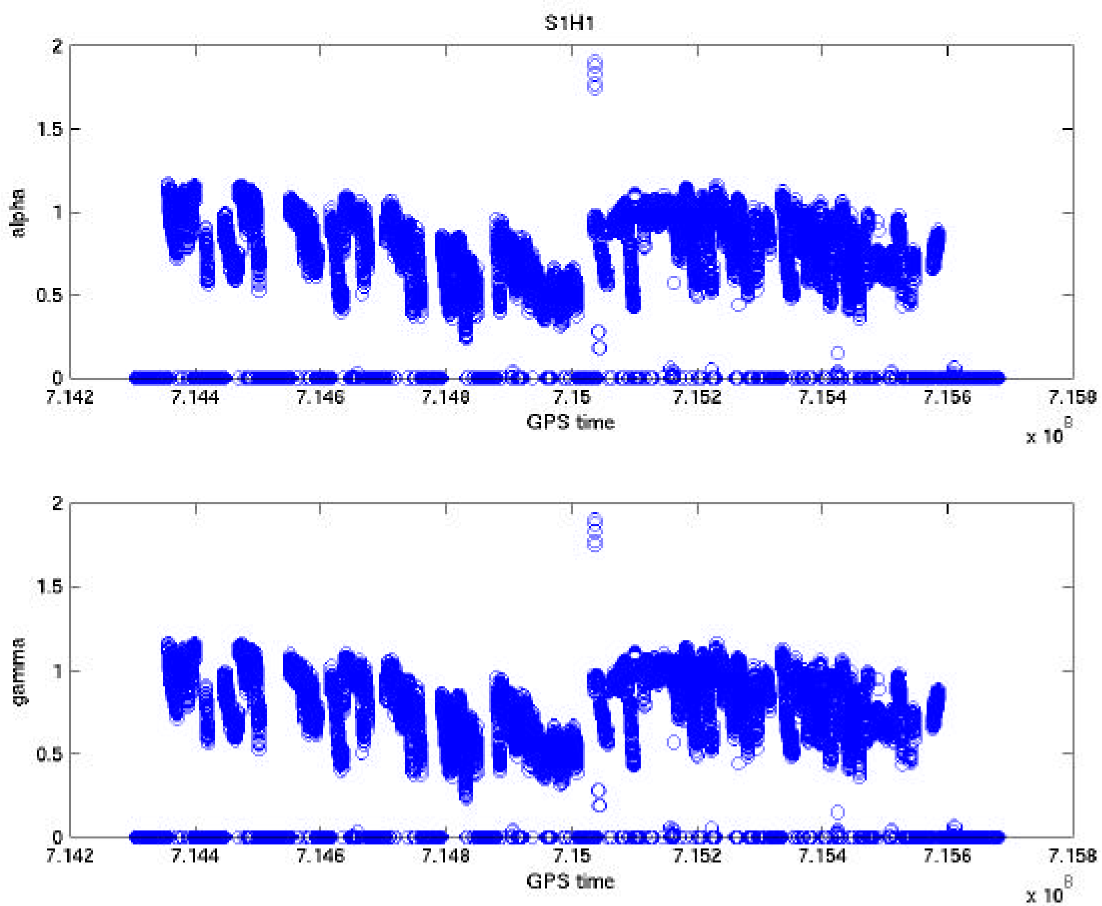
L1







Calib info availability



There are numerous data intervals throughout S1, even in the triple-coincidence, where α, γ are zero, or anomalously large or small, even through the data (psd) looks fine.

Presumably, this is due to the unavailability of calibration lines…

Do we veto such data stretches?

Patrick Sutton estimates that this reduces the triple coincidence by ~30%!!