h $\rightarrow$ AS_Q Filtering in the Time Domain

Patrick Sutton
LIGO-Caltech
for the Penn State PSURG Group

LIGO-G030574-00-Z

Outline

- Goals
- Calibration review
- Mindless first attempts
- Response to problems
- Status/Future

Motivation

- Ability of event trigger generators to detect gravitational wave signals is tested by adding simulated signals to the AS_Q data.
- Efficiency tests need mechanism to convert simulated signals from units of strain to AS_Q counts.
- BlockNormal bursts ETG is a time-domain search tool; we would like to have time-domain h $\rightarrow$ AS_Q filters.
- Other nice properties of time domain filters:
  - » causal
  - » can use online
  - » good for long data streams

Inverse Calibration ($h \rightarrow$ AS_Q)

- Inverse calibration in the frequency domain:

$$X_{AS\_Q}(f) = \frac{\alpha(t)\, C(f)}{1 + \alpha(t)\beta(t)\, G(f)}\, X_h(f)$$

  - » Open-loop gain G, sensing function C are functions of frequency only.
  - » Optical gain α and DARM gain β are functions of time only.
- Goal: Construct digital time-domain filters [b,a] with this frequency response for any $\alpha\beta$.

Calibration vs. Inverse Calibration

- Calibration function AS_Q → h is time-dependent sum of two fixed filters:

$$\frac{1+\alpha\beta\,G}{\alpha\,C} = \frac{1}{\alpha\,C} + \frac{\beta\,G}{C}$$

- Inverse calibration function h → AS_Q:

$$\frac{\alpha\,C}{1+\alpha\beta\,G} \neq \frac{\alpha\,C}{1} + \frac{C}{\beta\,G(f)}$$

Must mass-produce h → AS_Q filters (every $\alpha\beta$ in $0.5 < \alpha\beta < 2.5$).

Simple-Minded Approach

- Let MatLab do the work:

```
>> help invfreqz

INVFREQZ  Discrete filter least squares fit to
   frequency response data.

   [B,A] = INVFREQZ(H,W,NB,NA) gives real numerator and
   denominator coefficients B and A of orders NB and NA
   respectively, where H is the desired complex
   frequency response of the system at frequency points
   W, and W contains the normalized frequency values
   within the interval [0, Pi] (W is in units of
   radians/sample).
```

Knobs

- Knobs:
  - » The orders nb, na for the desired filter.
  - » The relative weighting w(f) of various frequencies in fitting the filter to the target inverse calibration function.
  - » Free to play with transfer function outside of frequency range of interest.
- Questions:
  - » What filter orders give good fits?
  - » What frequency weights give good fits?
  - » How does the quality of the fit vary with $\alpha\beta$?

Filter Quality

- Use fractional RMS error:

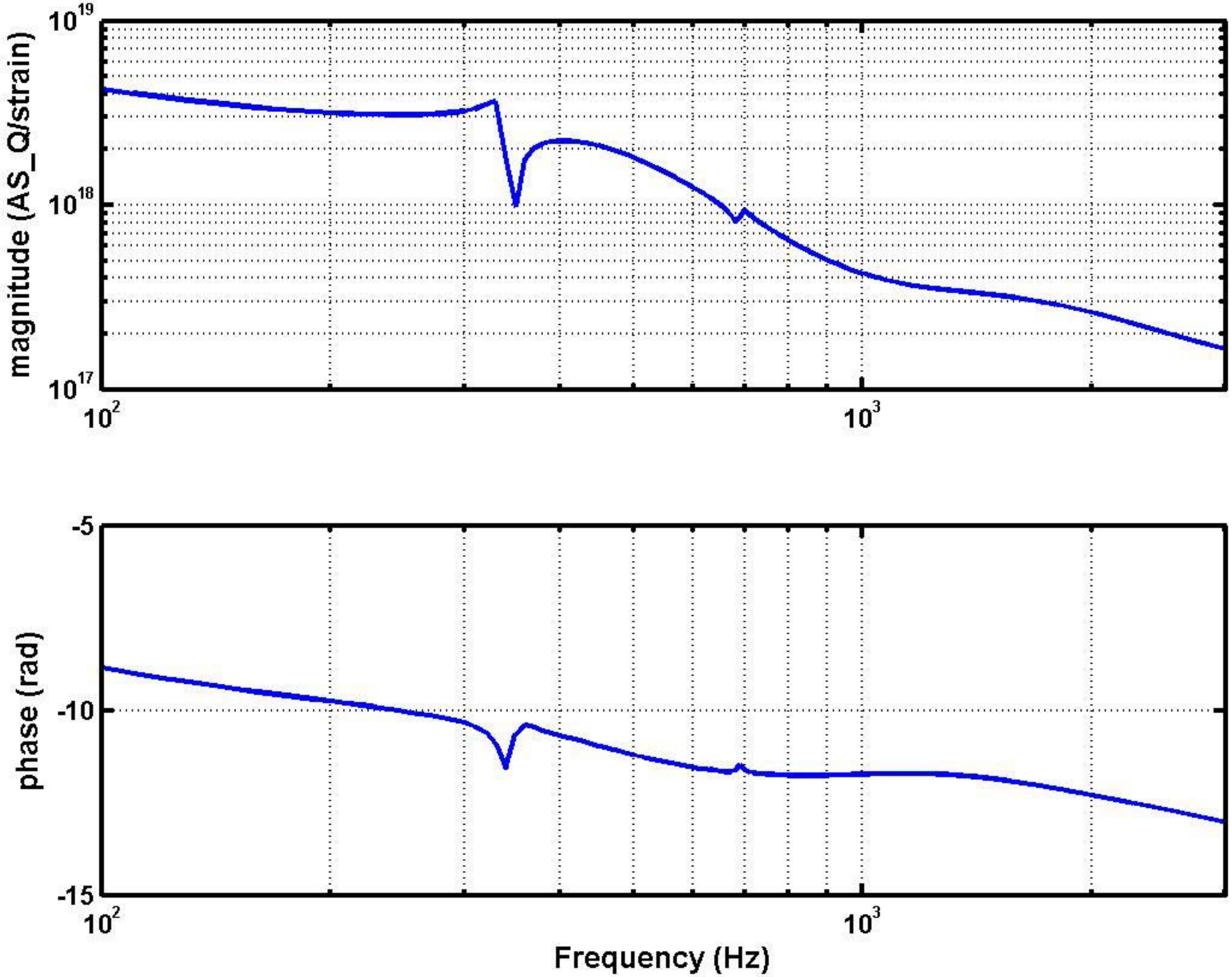
$$\text{Error} \equiv \sqrt{\frac{1}{N}\sum^{N}\left|\frac{Cb_{\text{filt}} - Cb}{Cb}\right|^2}$$

Compare desired to actual frequency response of filter at N frequencies. Ignore frequencies outside of the bands of interest (~[128,2048]Hz minus violin-mode harmonics).

- Desire error ≤ few %.

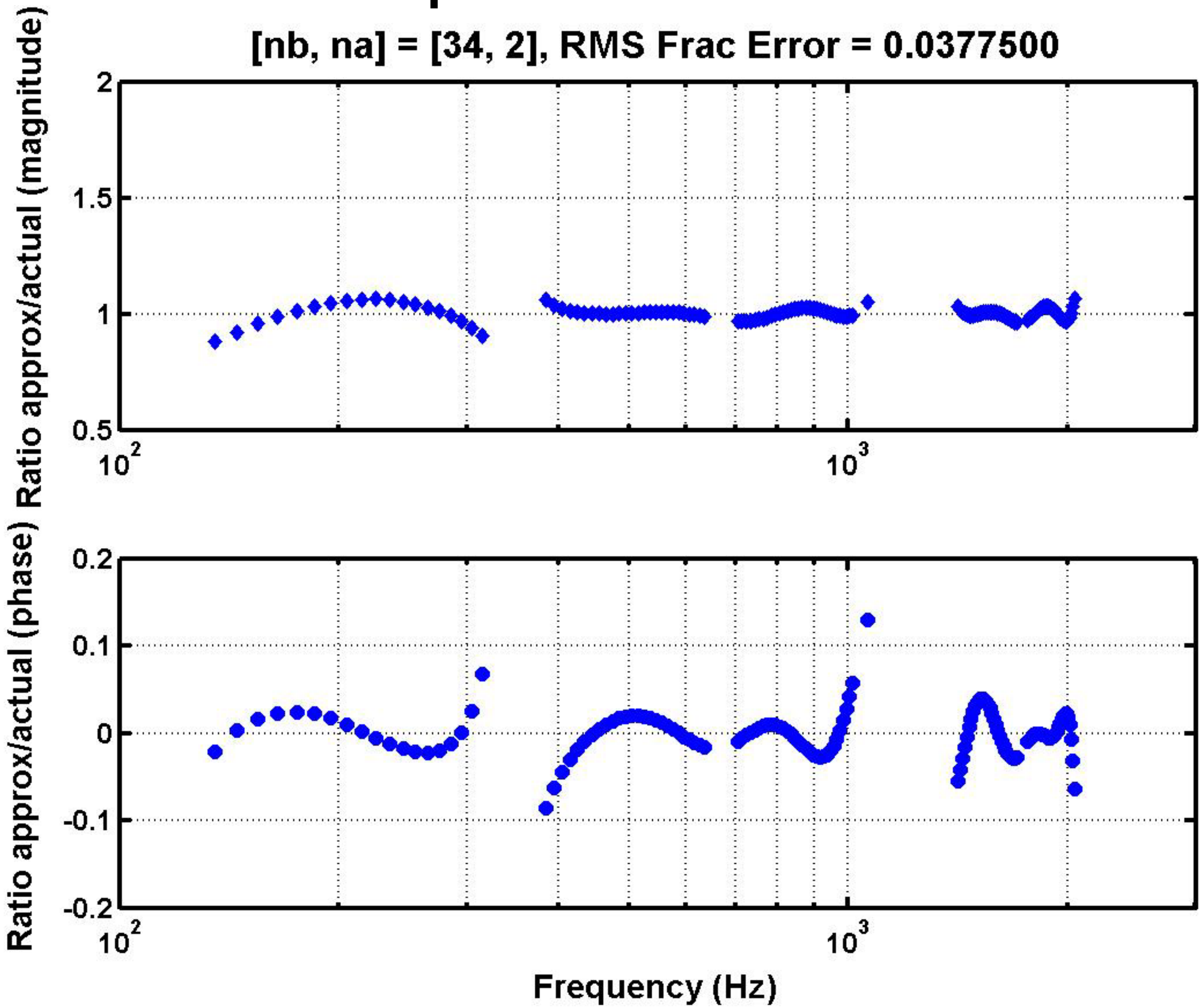
Example for H1:

Transfer function to match:



Example for H1:

How well we do:

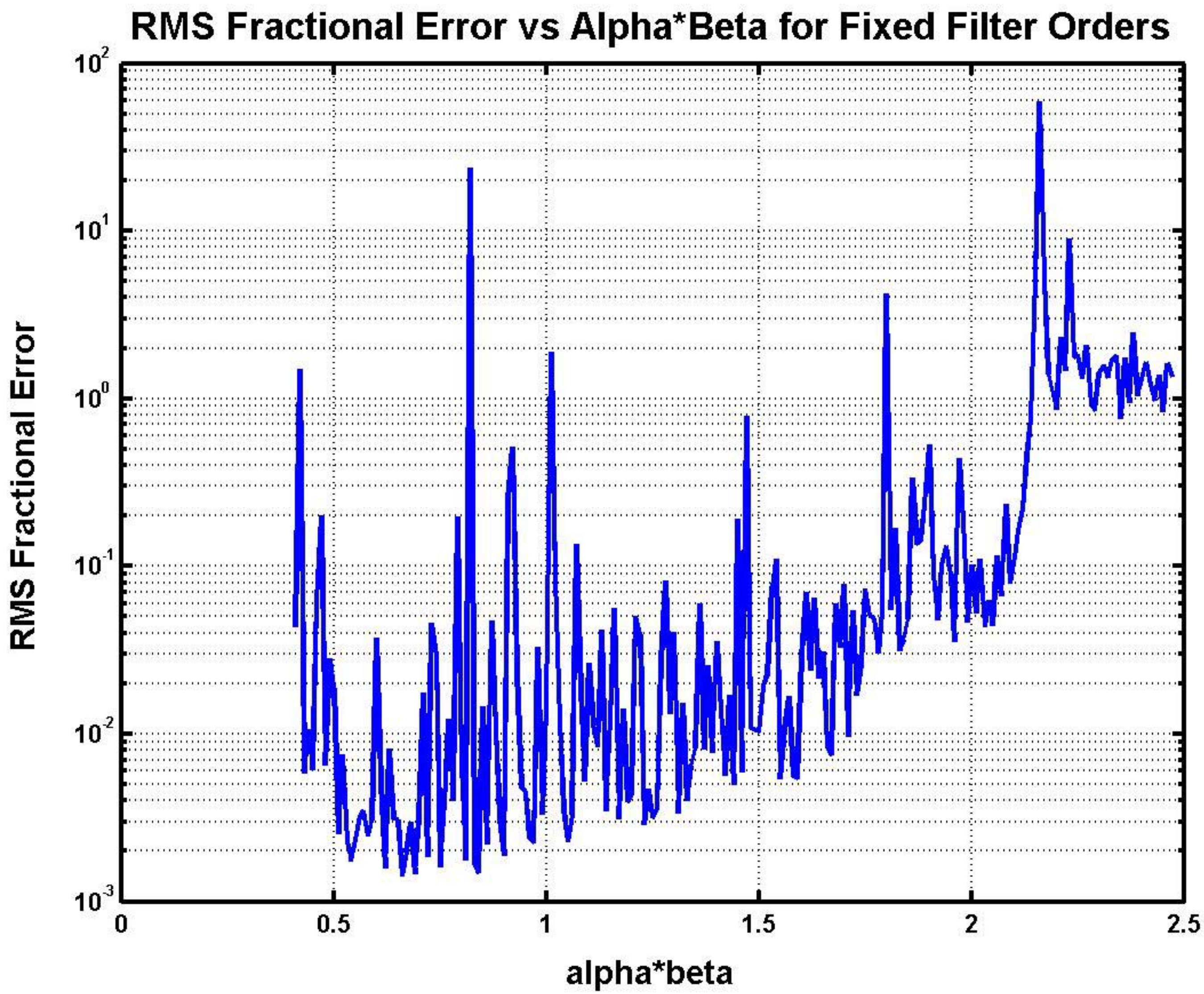


Problem solved?

- Can guarantee **`invfreqz`** filters are stable.
- *Not* guaranteed to be accurate. Vary $\alpha\beta$, see what happens…

Disaster!

Vary $\alpha\beta$, see what happens…

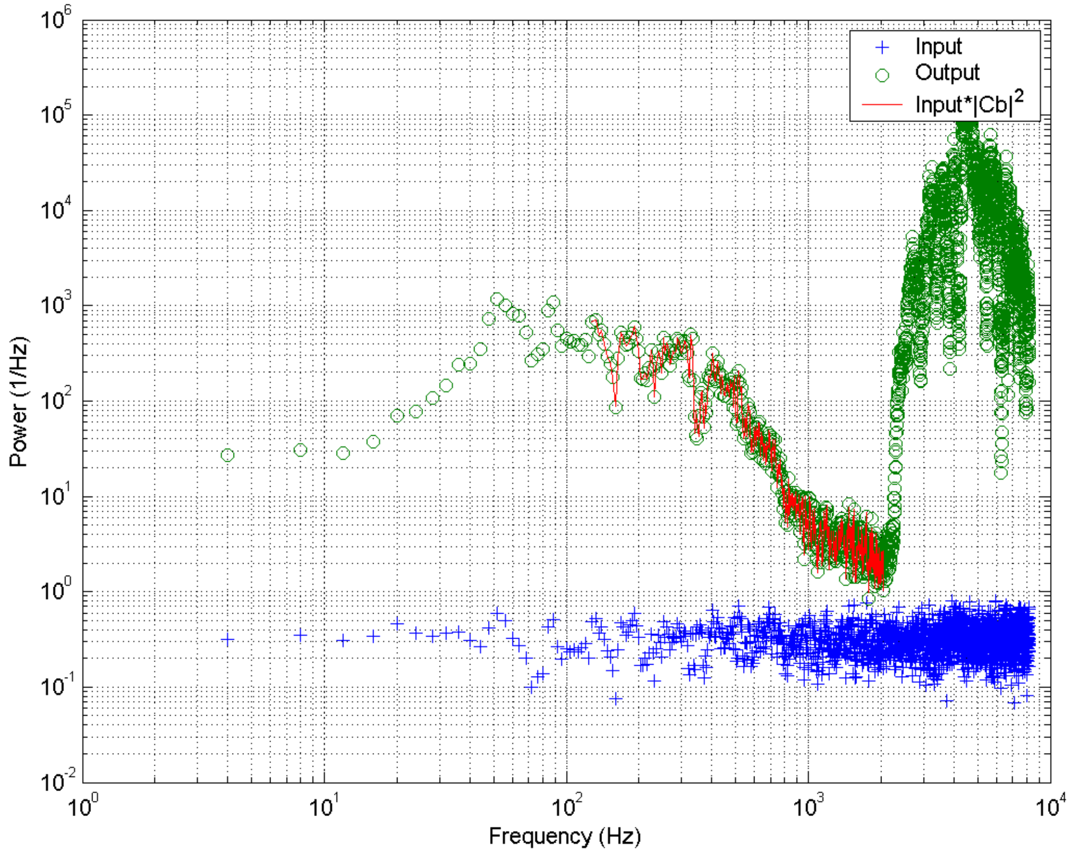


Implications

- **`invfreqz`** is not robust: changes of a few percent in $\alpha\beta$ can lead to filters with errors that differ by orders of magnitude.
- Problem is generic across filter orders nb, na.
- Reponse:
  - » don't generate filters on-the-fly.
  - » generate a fixed bank of filters for each IFO for closely-spaced $\alpha\beta$. Test each for quality before adding to the set.

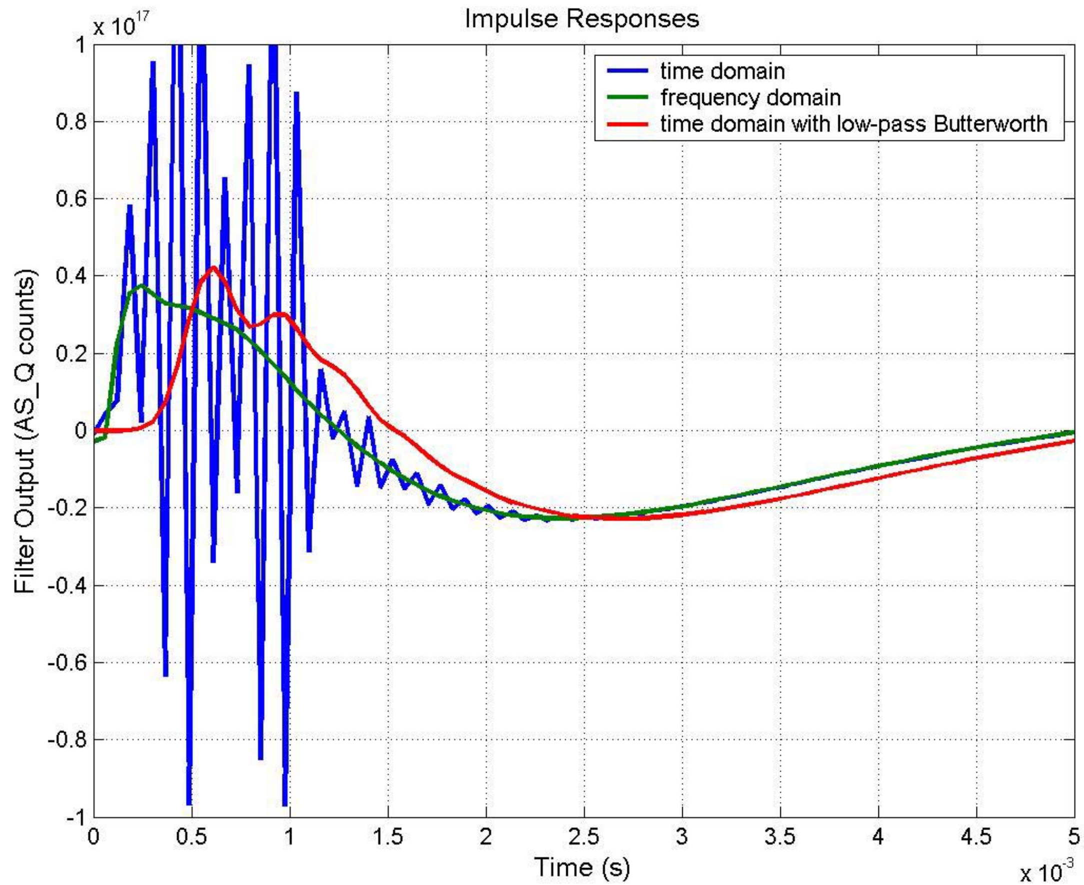
Next problem:

- Calibration info and/or frequency band of interest usually doesn't go up to Nyquist frequency; can't ignore this region.
- In the frequency domain:



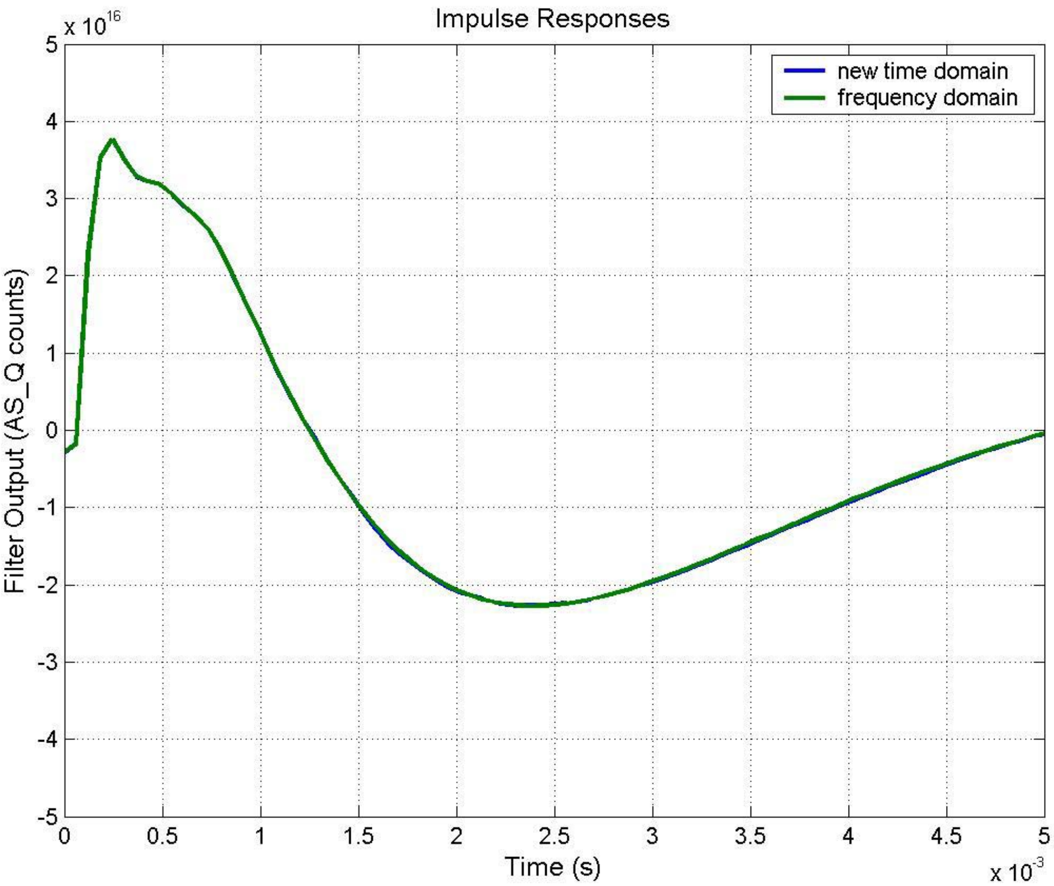
Next problem:

- In the time domain (impulse response):



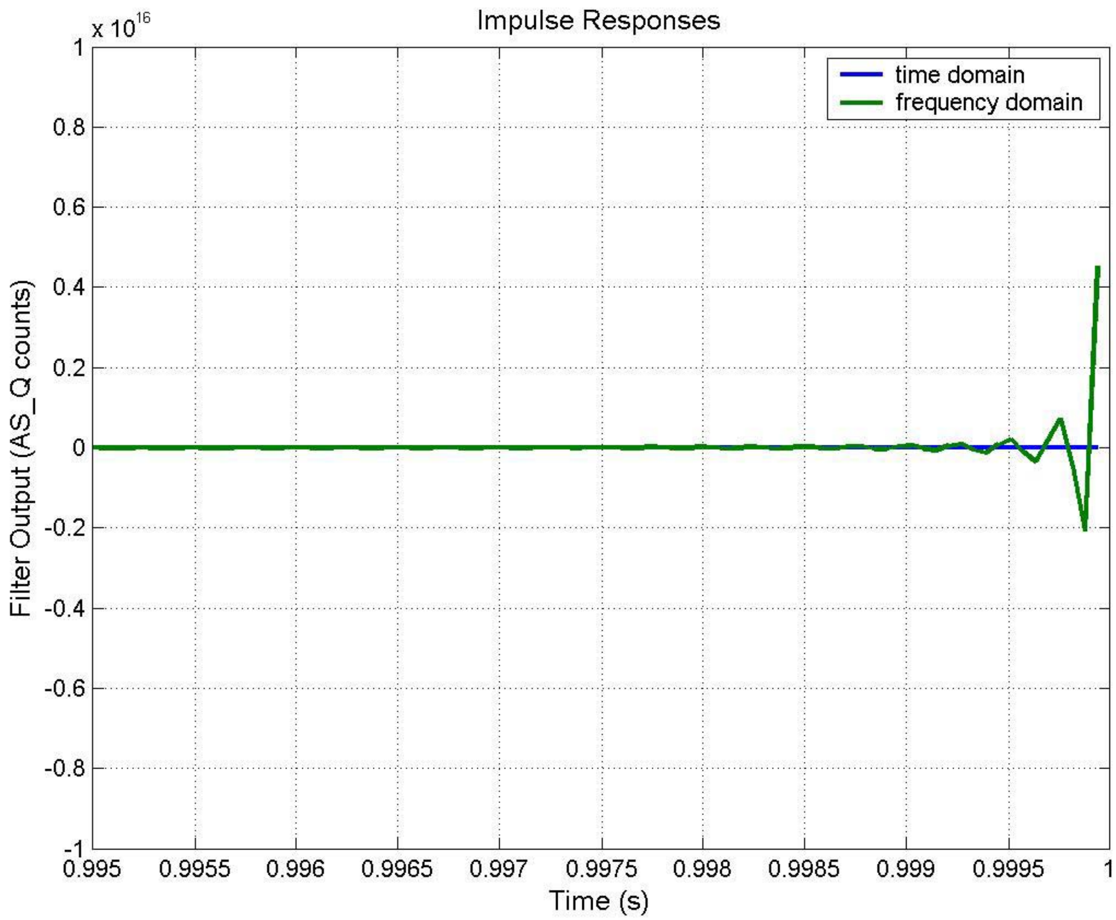
Solution:

- Extrapolate inverse calibration function smoothly to zero outside frequency range of interest:



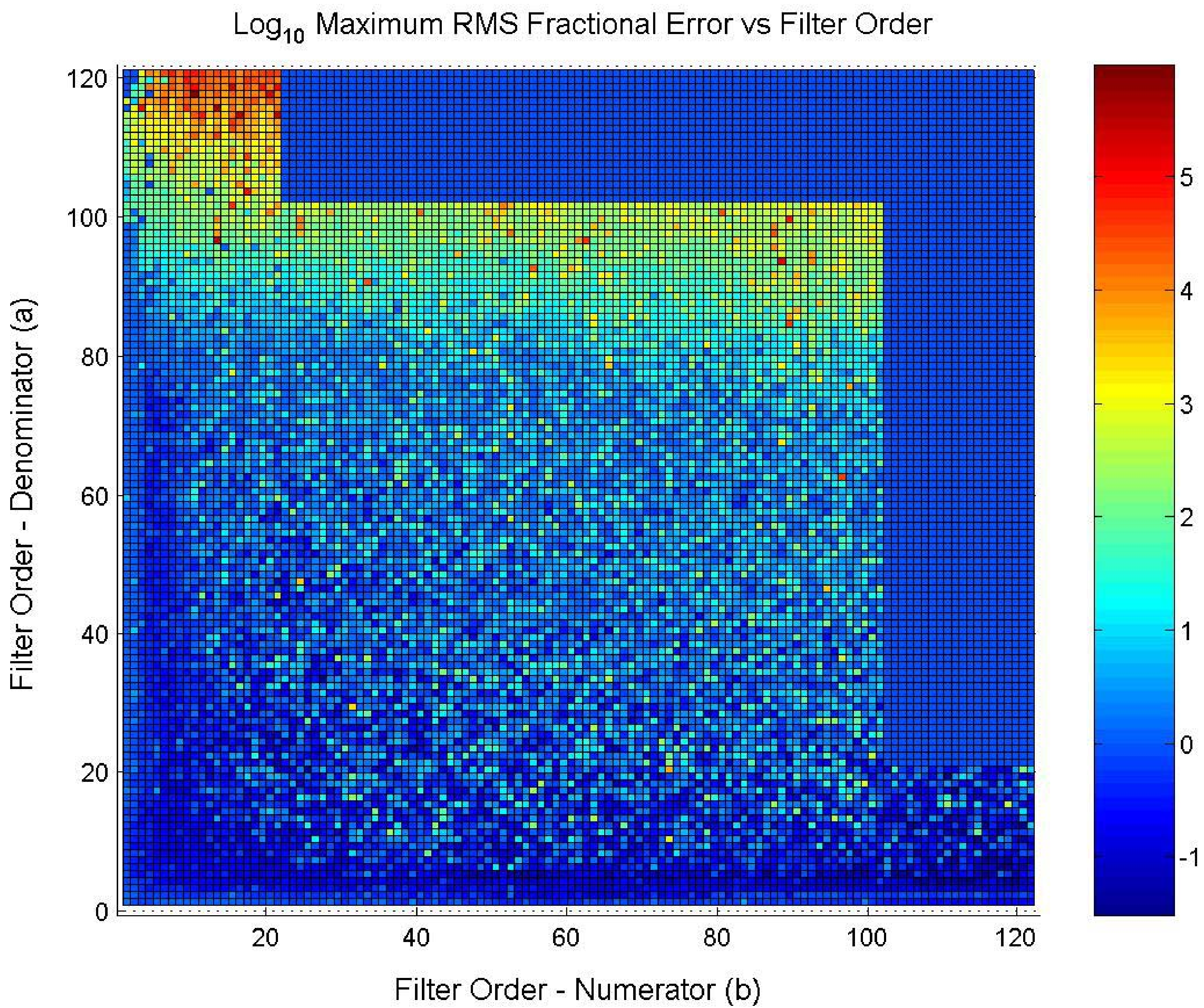
Nice bonus of time domain:

- Filter is causal:



Summary/Outlook

- Calibration: **good**.
- Time domain calibration: **hard**, but possible if careful.
- Next steps: Generate S2 H1/H2/L1 filter banks for closely spaced values of $\alpha\beta$.
- Other possibility: get smart and use knowledge of IFOs. See what Adhikari's MatLab IFO model can do for us.
- Suggestions are welcome!



Log10 Maximum RMS Fractional Error vs Filter Order
Filter Order - Denominator (a)
Filter Order - Numerator (b)
0
20
40
60
80
100
120
-1
0
1
2
3
4
5