PART II:
The OMC Whitening Chassis, from Mar 16 to 27 2020

Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What's the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What's the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.1 What’s the issue?

[LHO:55620](#)
[IIET:14567](#)

Monday Afternoon, Time of first observation ready segment after change is

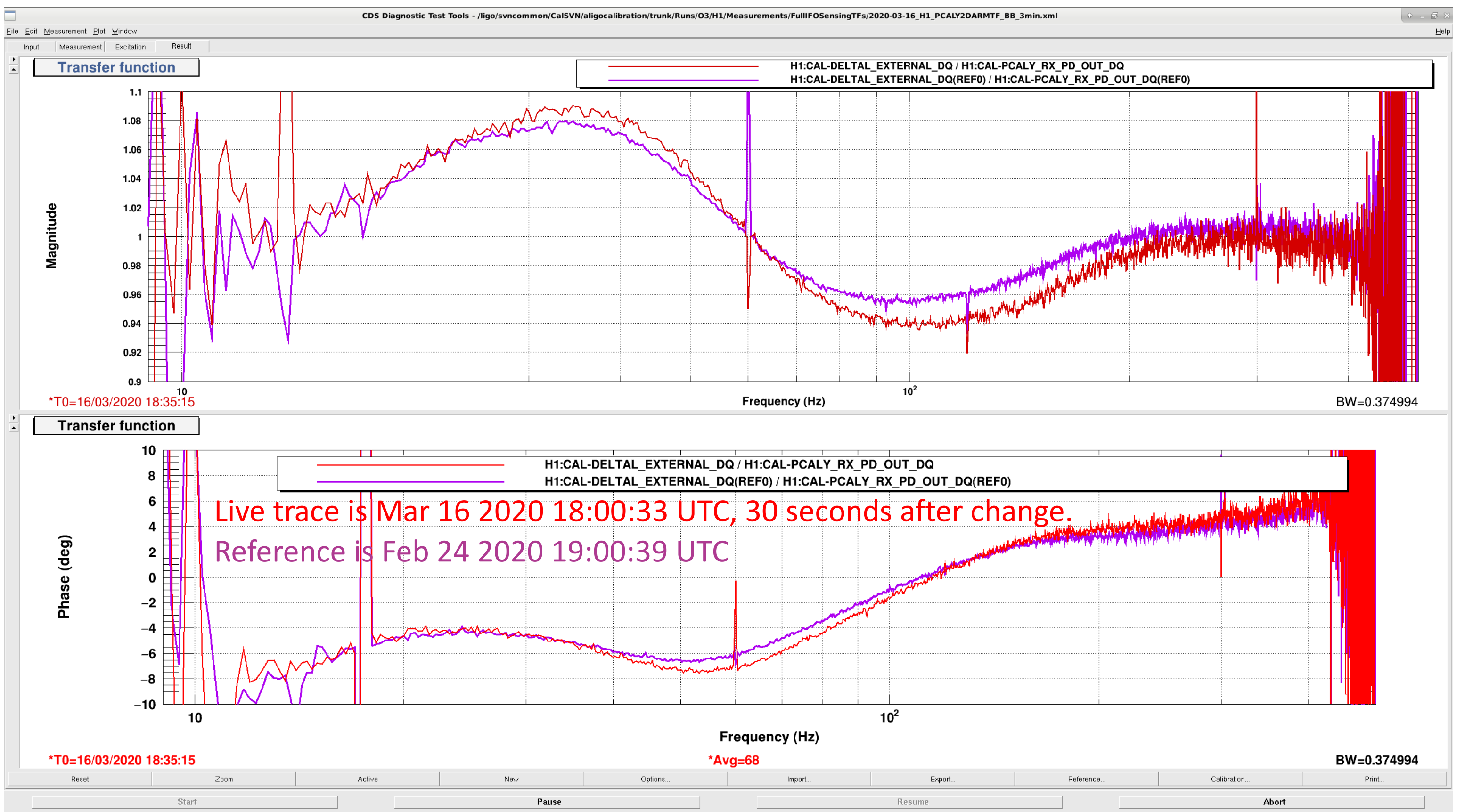
Mar 16 2020 14:18:54 PDT

Mar 16 2020 21:18:54 UTC

GPS 1268428752

Reduced the amount of OMC analog whitening from all 3 stages (2 whitening stages and one low pass) to just 1 stage of whitening. Results in 1-1.5% systematic error change.

Could have modelled sooner and created a new pyDARM parameter file, but ran out of time, so now we must predict the systematic error to be included in the uncertainty budget.



Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

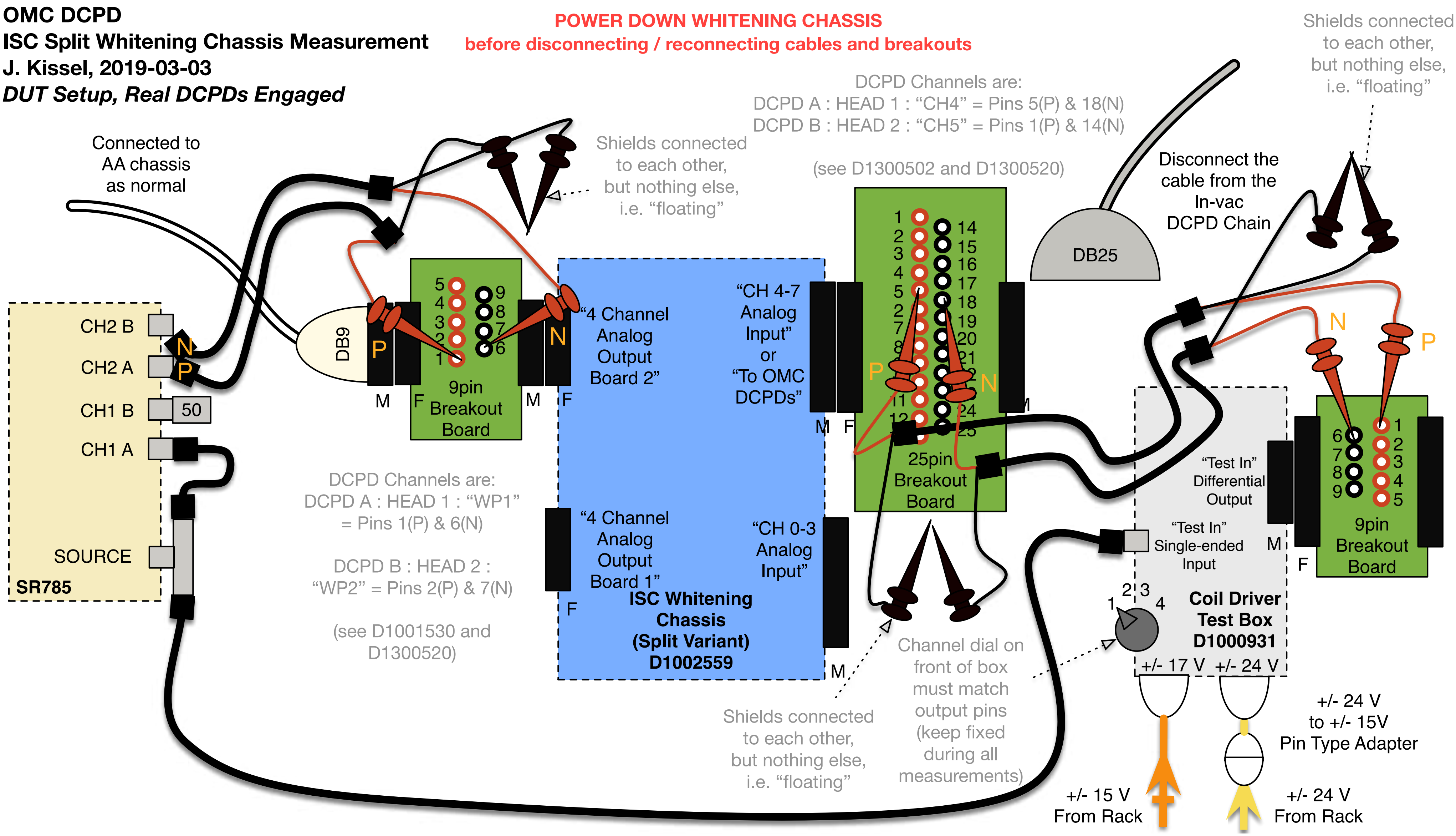
PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What’s the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.2 Game Plan

- Revisit the Stefan / Lilli aLOG, find the fitting script, make sure we can reproduce the results.
  - Will need fit-from-measured zpks's of each stage of both channels. Do we even have the right data to do this?
  - Find that data, if it's not used in Lilli's fitting script already.
  - If need be replot / innovate Lilli's script.
- Find the right Foton Filter File, get all the z's p's and k's for each filter module.
- Make a prediction of
  - All three stages ON and compensated, vs.
  - Only one stage ON and compensated
- Once DCS frames become available, re-process broadband injections before vs. after change.
  - Maybe Maddie has processed them already? check email.
  - Compare against prediction above.
- Build the ability to include that frequency-dependent systematic error in to RRNom.

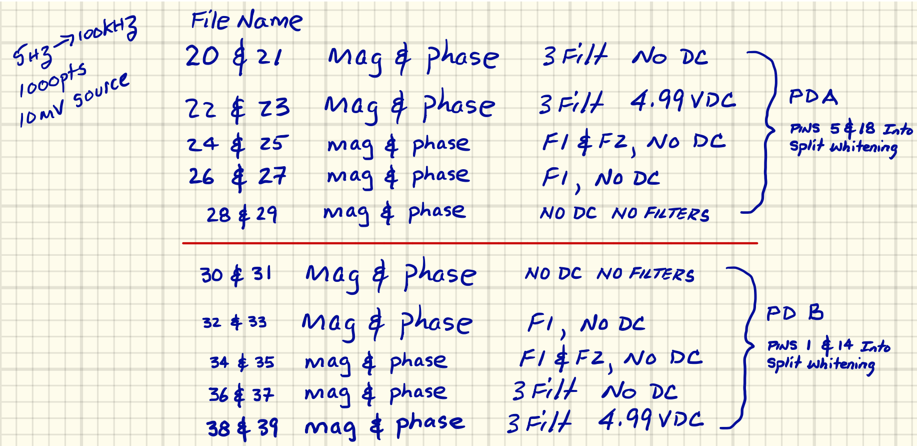
II.2 Review of the Data: Measurement Setup [D1900027]



Open questions (now that we've learned such a valuable lesson in Part I and in LHO:48358 about paying attention to electrical grounds): Should we have floated the shields? Does it matter that we're "connected to the AA chassis as normal?"

II.2 Review of the Data: Lots of renaming…

1. Rich measured the chassis, zipped up the files, and gave them to Stefan as .78d files: LHO:47418
2. Stefan converted the .78d files to useable .asc data, and wrote a script to build the compensation filters: LHO:47257
3. The script and data were renamed and moved by Jeff to CalSVN: LHO:47290
4. Lilli renamed the files (LHO:47358) that based on Rich's configuration map (from LHO:47418)



Here "F1", "F2" (Rich) or "FM1," "FM2" (Lilli) is for "filter module," [as in the corresponding compensation filter] but really means "whitening stage" [as in the corresponding analog filter stage].
All of these newly renamed files were processed with
fit_OMCDCPDWhiteningChassis_20190304.py
and for the record, used a former measurement of the coil driver test box that was used from the 2019-03-07 measurement of S1900070.

Rich's Measurement configuration	DCPD A (Chan 4) SR785 File Name	DCPD A (Chan 4) Lill's File Name	DCPD B (Chan 5) SR785 File Name	DCPD B (Chan 5) Lilli's File Name
All three stages ON	srs0020.78d	PDA_FM1FM2FM3_mag.asc	srs0036.78d	PDB_FM1FM2FM3_mag.asc
no DC voltage offset	srs0021.78d	PDA_FM1FM2FM3_pha.asc	srs0037.78d	PDB_FM1FM2FM3_pha.asc
First two stages ON	srs0024.78d	PDA_FM1FM2_mag.asc	srs0034.78d	PDB_FM1FM2_mag.asc
no DC voltage offset	srs0025.78d	PDA_FM1FM2_pha.asc	srs0035.78d	PDB_FM1FM2_pha.asc
First stage ON	srs0026.78d	PDA_FM1_mag.asc	srs0032.78d	PDB_FM1_mag.asc
no DC voltage offset	srs0027.78d	PDA_FM1_pha.asc	srs0033.78d	PDB_FM1_pha.asc
All stages OFF	srs0028.78d	PDA_noFM_mag.asc	srs0030.78d	PDB_noFM_mag.asc
no DC voltage offset	srs0029.78d	PDA_noFM_pha.asc	srs0031.78d	PDB_noFM_pha.asc

Measurement was taken *after* all ECRs (including fixing sensitivity to DC offset) were implemented
So Lilli using "no DC offset" data is OK, even though DCPDs during NLN have ~50 mA = ~5V running through the chassis

II.2 Review of the Data: Welp, I need to refit the data

1. I need to understand with what raw data I'm working, so I can understand how to work with it.
2. Stefan's metric for success in LHO:47257 was "A / B", but the DCPDs are *added*, so we need something like "A + B"
3. Although the fit looks pretty fantastic in LHO:47358, Lilli just fit each measurement straight up, and didn't compute any ratios of transfer functions to isolate poles and zeros from each stage, so hard to verify which poles are coming from which stage (and if say, FM1's fit changes when it's a part of FM1, FM2, and FM3).
4. When computing the residual contribution to R in LHO:47453, Lilli plotted (A+B)/2, instead of (b_A*A+b_B*B), where b_A and b_B are the balance matrix values,
   - (no need for the "divide by two" part of the "average" since it's an over all gain, and captured in H_C)
5. I think there's a bug in fit_OMCDCPDWhiteningChassis_20190304-compareTF.py, fit_OMCDCPDWhiteningChassis_20190304-compareRes.py, and fit_OMCDCPDWhiteningChassis_20190304-compareRes-writeTXT.py, which all mis-represent the poles and zeros installed in foton; we should just import the design from the real foton file, H1OMC_1239468752.txt
6. IIRrational has improved, and should now need a lot less input, so we can make the code cleaner.

ALSO, Again, Open Question: As discussed in LHO:48358 (learned, unfortunately, a month later), the DUT and coil driver test box *must* be properly referenced to ground, or one may get distortions in the frequency response at high frequency. The data for S1900070 clearly shows a gain of 2.002, confirming that at least the test box measurement was done with an errant "floating" ground. Nothing we can do about it now, but the OMC whitening chassis might behave like the AA chassis, where the high-frequency response gets distorted without measurement shields properly referenced to ground.

Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What’s the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.3 Review of the Circuit: Squad goals

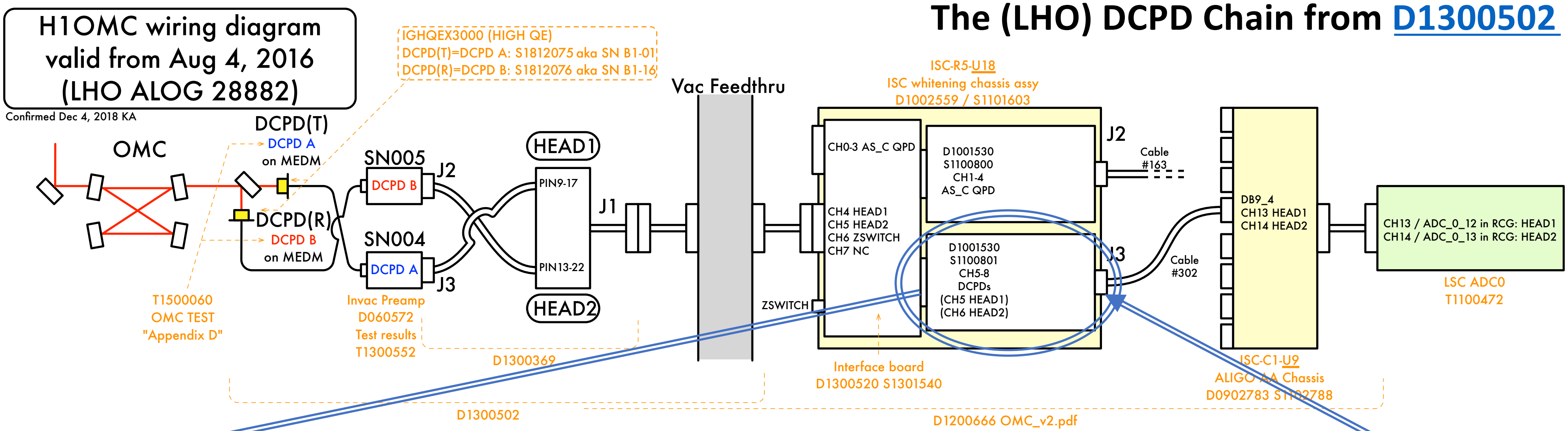
- I'm gunna do the same thing I did for the UIM driver and break down the circuit into manageable pieces (I'll go a little faster, assuming you're used to the methods now having gone through Part I).

- Again, this is (a) to teach you, the next generation, and (b) **so can be sure we know what to expect when fitting, as well as verifying that we're modelling it correctly.**

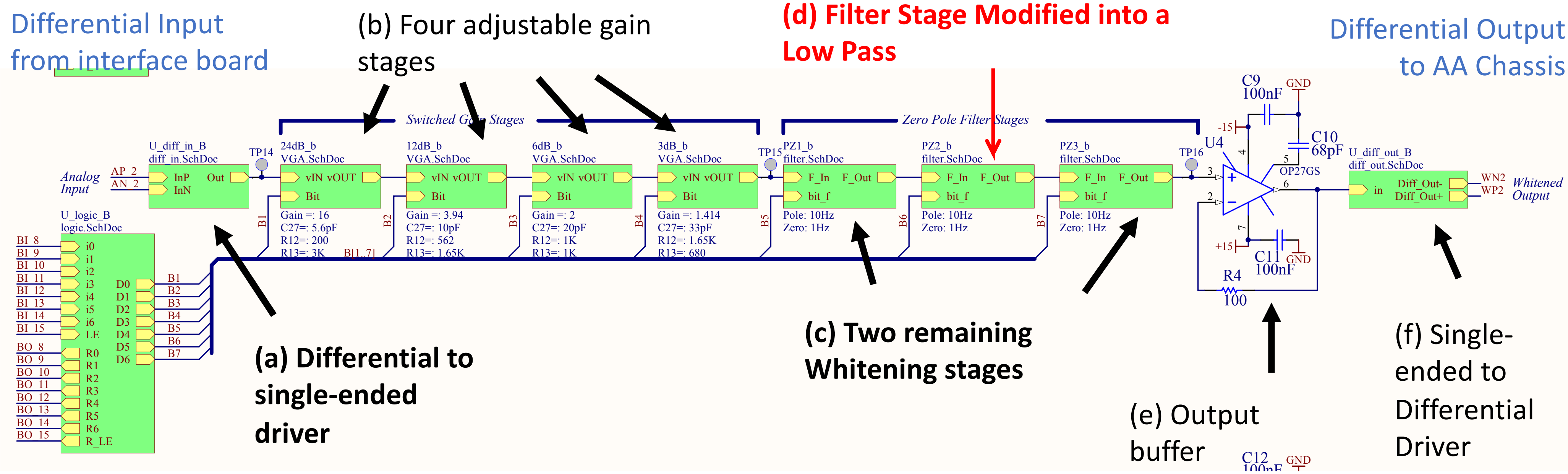
- As before, we'll be focusing on finding frequency response, since the over all gain of the OMC DCPD chain is measured as a whole. You'll find that the super-Nyquist high-frequency poles are especially interesting…

II.3 Review of the Circuit: Forest through the Trees

The (LHO) DCPD Chain from D1300502

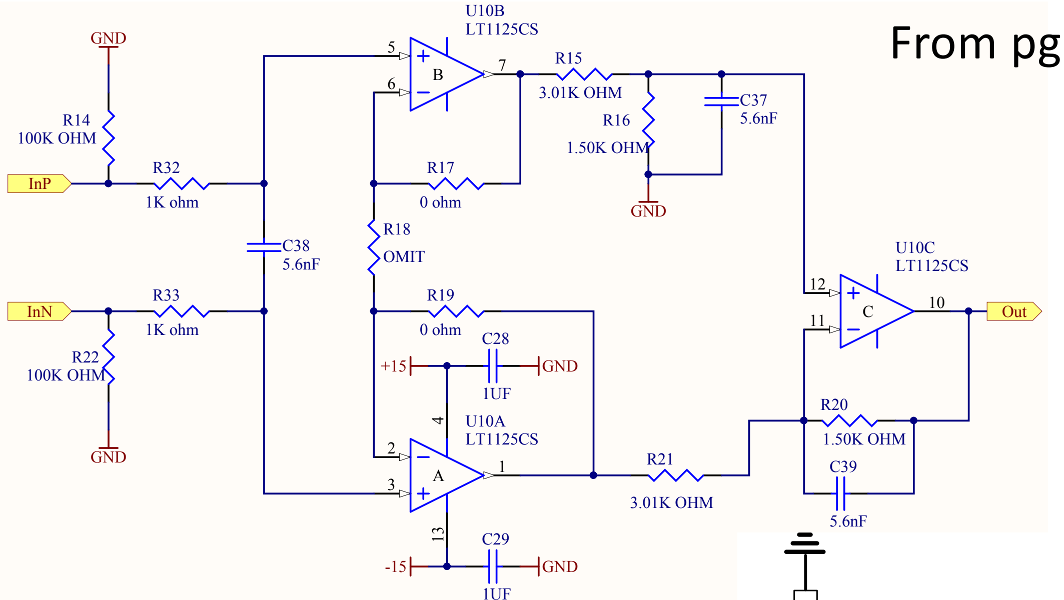


The complex chain internal to the whitening chassis alone from pg 1 D1001530-v7



***Bold things** we'll find to have interesting frequency response*

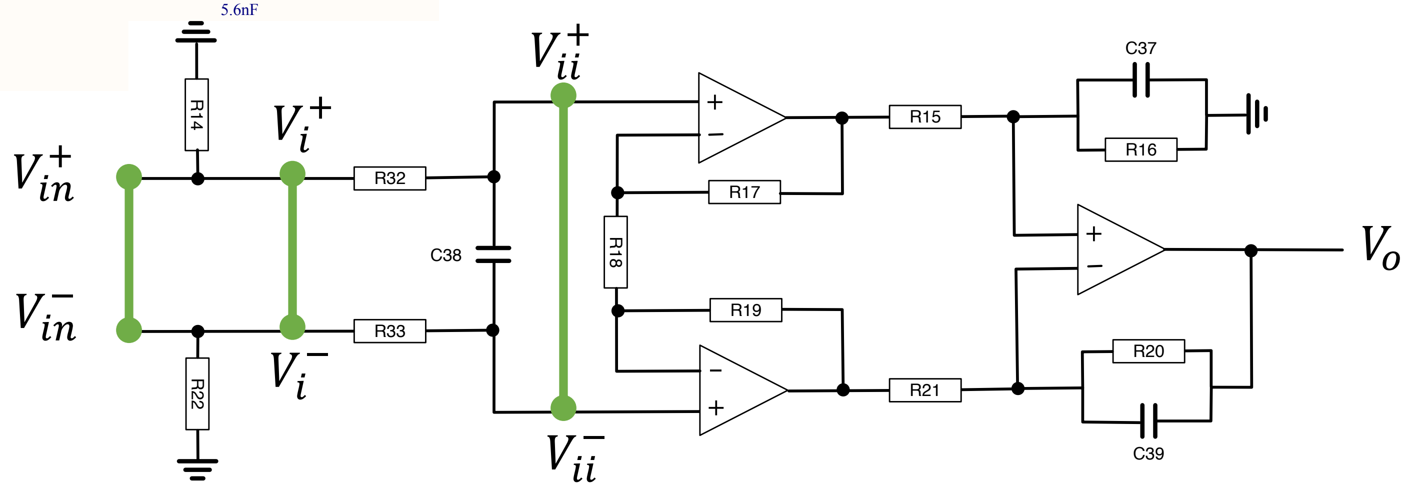
II.3 Review of the Circuit: (a) Diff.-to-S.E. Driver



From pg 2 of D1001530-v7

To analyze, let's break in to three parts

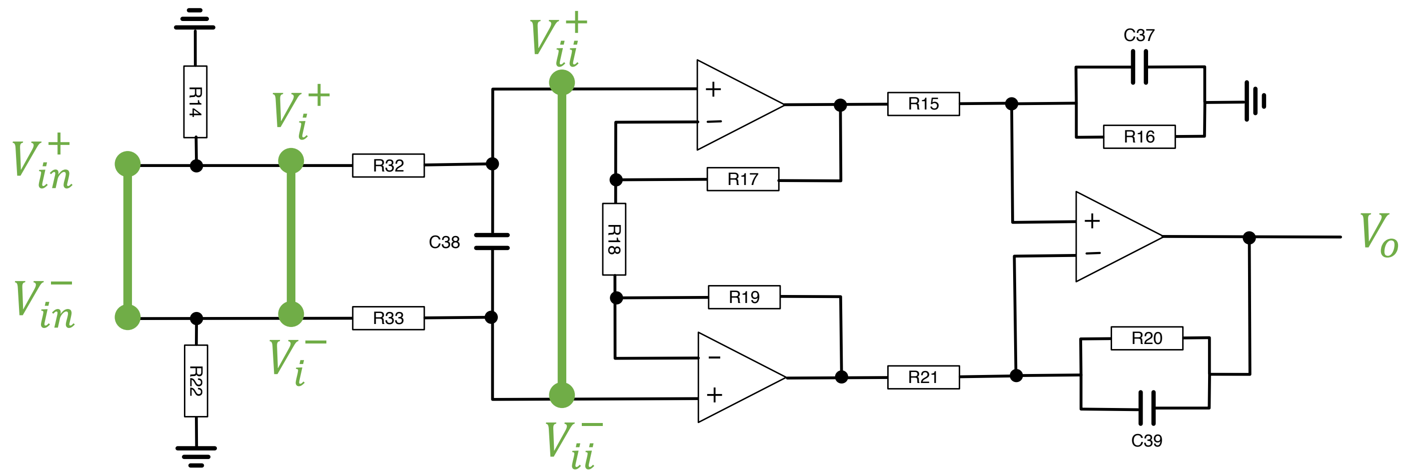
i. The voltage divider
ii. some passive input low pass filter
iii. the *actual* differential-to-single-ended driver



For (a), it's our typical LIGO input voltage divider, aka load termination, for the long cable run from in vacuum. That means we have to understand to what this board is connected (guh). So, we go upstream to the interface board, D1300520-v3 (see no impedance), then further to the head D1300369-v1 (see no impedance), then finally to the in-vac preamp D060572-v1, to see an impedance of 1k (plus whatever small resistance from the cable run). *This* time, unlike the UIM, I think we can ignore parasitic cable capacitance, so this is just a gain "drop" of

$$G_{in} = V_i^+/V_{in}^+ = V_i^-/V_{in}^- = R_{14}/(R_L + R_{14}) \approx 0.99$$

II.3 Review of the Circuit: (a) Diff.-to-S.E. Driver



For (ii), the passive low-pass filter, we can do the usual trick of splitting the different circuit in half, and dividing the impedance of components that span the legs in two. In this case just a capacitor, so

$$\left.\frac{(V_{ii}^+ - V_{ii}^-)}{(V_i^+ - V_i^-)}\right|_{C_{38}} = \left.\frac{V_{ii}^+}{V_i^+}\right|_{2C_{38}} = \frac{1/i\omega 2C_{38}}{(R_{32} + 1/i\omega 2C_{38})} = \frac{1}{(1 + i\omega(2R_{32}C_{38}))}$$

$$f_p^{LP} = 1/\left(2\pi(2R_{32}C_{38})\right) = 28.42\ \text{kHz}$$

For (iii), the actual differential driver, you can see that I've re-drawn it to better match the traditional analysis for of an instrumentation amplifier, make it easier to agree that the transfer function takes the form

$$G_{ia} = \frac{V_o}{(V_{ii}^+ - V_{ii}^-)} = \left(1 + \frac{2R_{17}}{R_{18}}\right)\frac{Z_{RC}}{R_{15}} \qquad \left.G_{ia}\right|_{DC} = \frac{R_{16}}{R_{15}} = 0.498$$

because, as expected, the circuit is symmetric (e.g. R17=R19, R15=R21, etc.). Since R17 = 0 Ω, and R18 has been omitted, i.e. R18 = ∞ Ω, the first term reduces to 1.0. With

$$Z_{RC} = R_{16}/(1 + i\omega R_{16}C_{37})$$

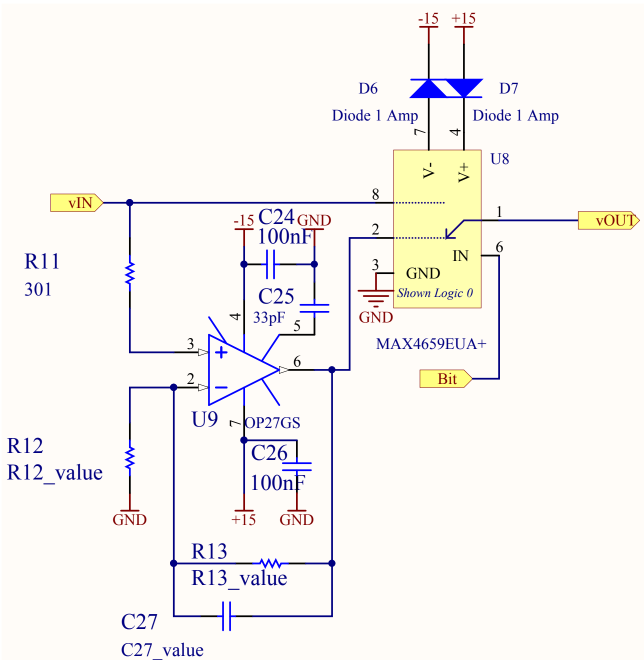
then

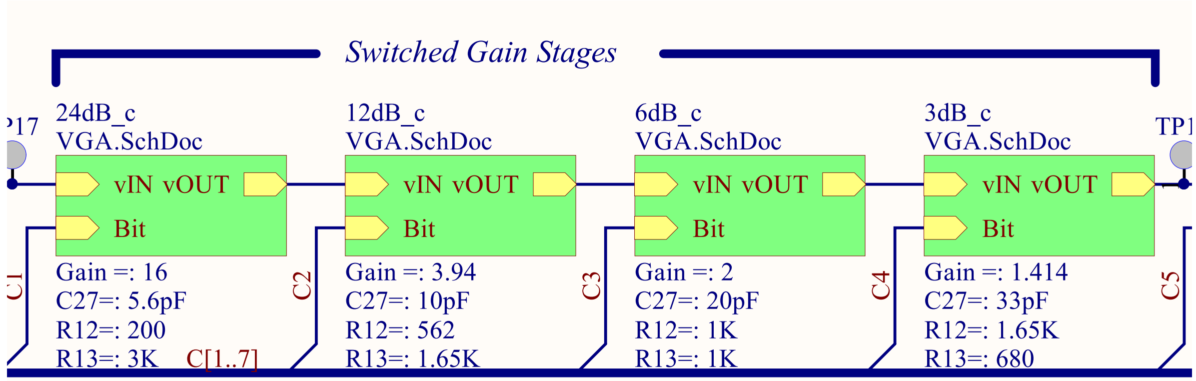
$$G_{ia} = \frac{R_{16}}{R_{15}} \frac{1}{(1 + i\omega R_{16}C_{37})}$$

$$\boldsymbol{f_p^{ia} = 1/(2\pi R_{16}C_{37}) = 18.95\ \text{kHz}}$$

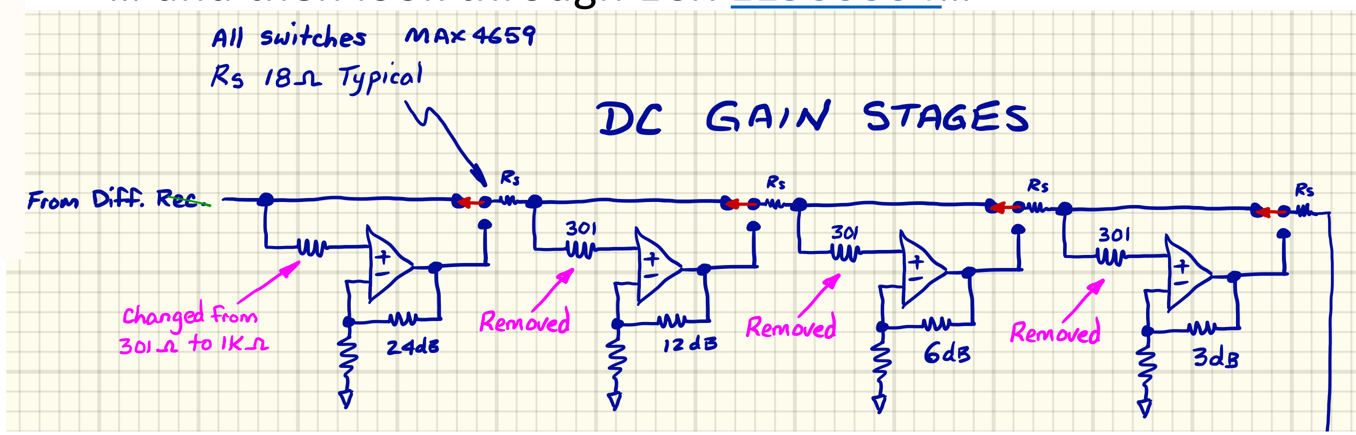
II.3 Review of the Circuit: (b) Adjustable Gain Stages

From pg 3 of D1001530-v7… … and back to pg 1 of D1001530-v7 to get values…





… and then look through ECR E1900064…



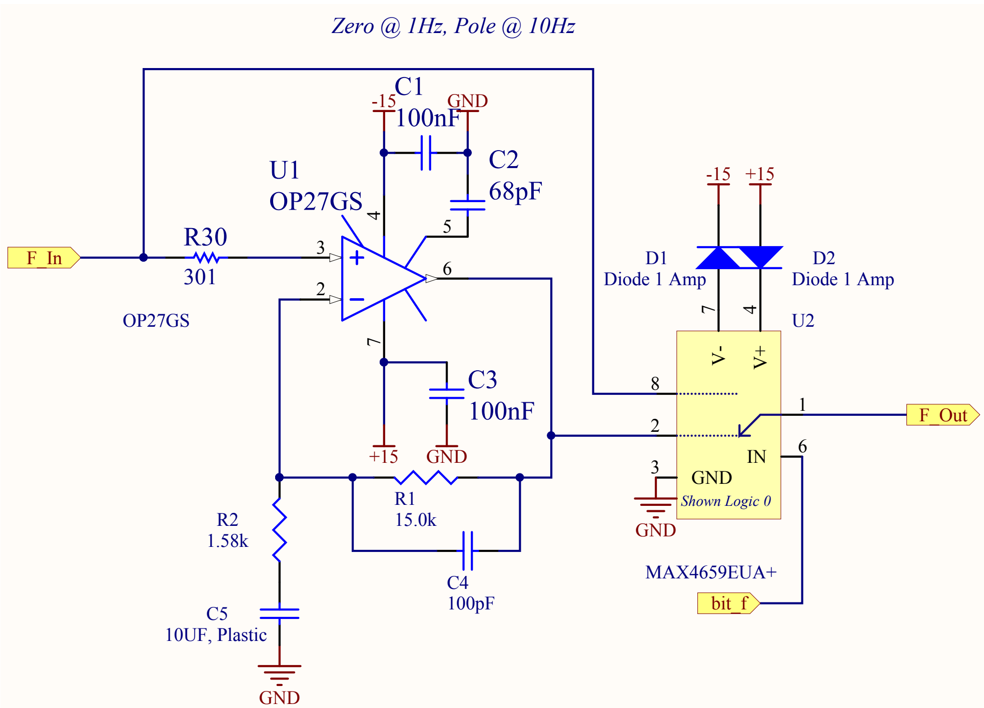
To remember that ***all* of the switchable gain stages have been disabled / bypassed in O3**:

- the removal of each R11 in the last three stages makes an infinite resistance to ground, so there's no longer any input, and further (though no where so explicitly stated) the switches are bypassed with a jumper so the output of the opamp is no longer connected.
- R11 in the first stage remains, but is increased to 1k. Because it's connected to the ZRC network of the "Diff. Rec." aka "Differential to Single Ended Driver" aka "Instrument Amplifier," and the real part of ZRC is dominated by R20 = 1.5 kΩ, which means the only remaining, fixed gain is

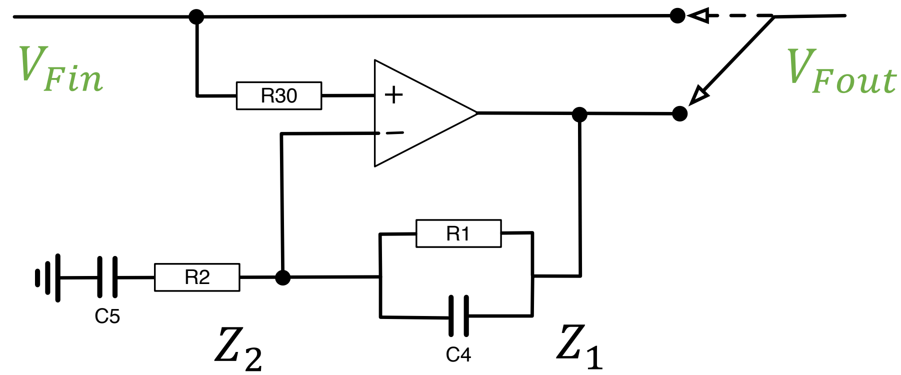
$$G = R_{11}/(R_{20} + R_{11}) = 1500/(1000 + 1500) = 0.4$$

II.3 Review of the Circuit: (c) Whitening Stages

From pg 4 of D1001530-v7…



These can be redrawn as a non-inverting amplifier…



$$Z_1 = \frac{R_1(1 + 1/i\omega C_4)}{(R_1 + 1/i\omega C_4)} = \frac{R_1}{(1 + i\omega R_1 C_4)}$$

$$Z_2 = R_2 + 1/i\omega C_5 = \frac{(1 + i\omega R_2 C_5)}{i\omega C_5}$$

$$\frac{V_{Fout}}{V_{Fin}} = 1 + \frac{Z_1}{Z_2} = 1 + \frac{\frac{R_1}{(1 + i\omega R_1 C_4)}}{\frac{(1 + i\omega R_2 C_5)}{i\omega C_5}} = \frac{1 + i\omega(R_1C_5 + R_1C_4 + R_2C_5) - \omega^2(R_1R_2)(C_4C_5)}{(1 + i\omega R_2C_5)(1 + i\omega R_1C_4)}$$

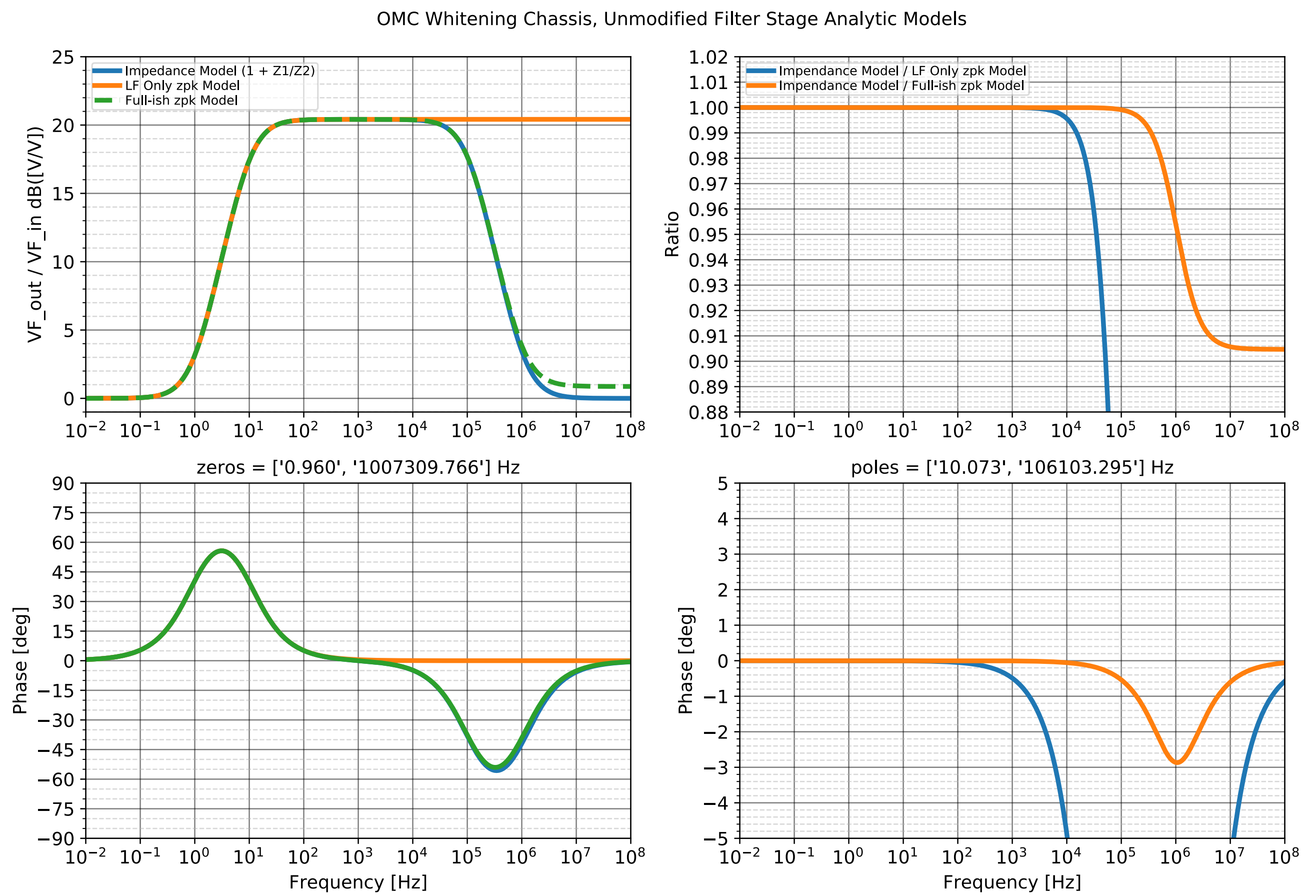
$$\boldsymbol{f_z^F = 1/\big(2\pi(R_1C_5 + R_1C_4 + R_2C_5)\big) = 0.960\ \text{Hz}}$$

$$f_z^F = 1/[(2\pi)^2(R_1R_2C_4C_5)] = 1.069\ \text{MHz}$$

$$\boldsymbol{f_p^F = 1/\big(2\pi(2R_2C_5)\big) = 10.073\ \text{Hz}}$$

$$\boldsymbol{f_p^F = 1/\big(2\pi(2R_1C_4)\big) = 106.103\ \text{kHz}}$$

II.3 Review of the Circuit: (c) Whitening Stages



The z:p pair we really want at 0.96:10.07 Hz i.e. the “1:10” whitening as expected.

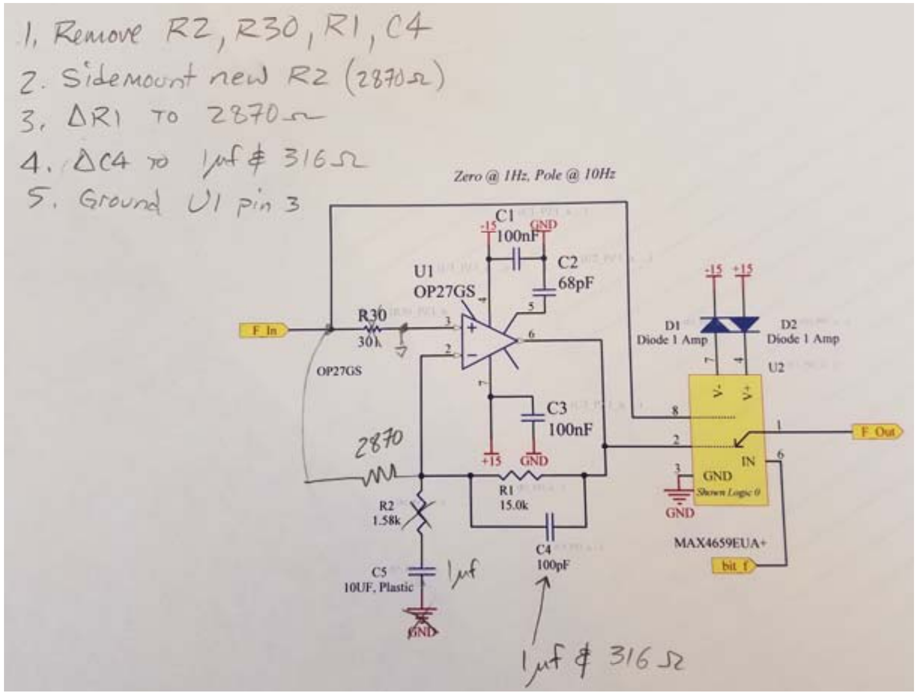
But... there is the z:p pair at 1.06:0.103 MHz that causes 0.5 deg phase loss at 1 kHz if we exclude it.

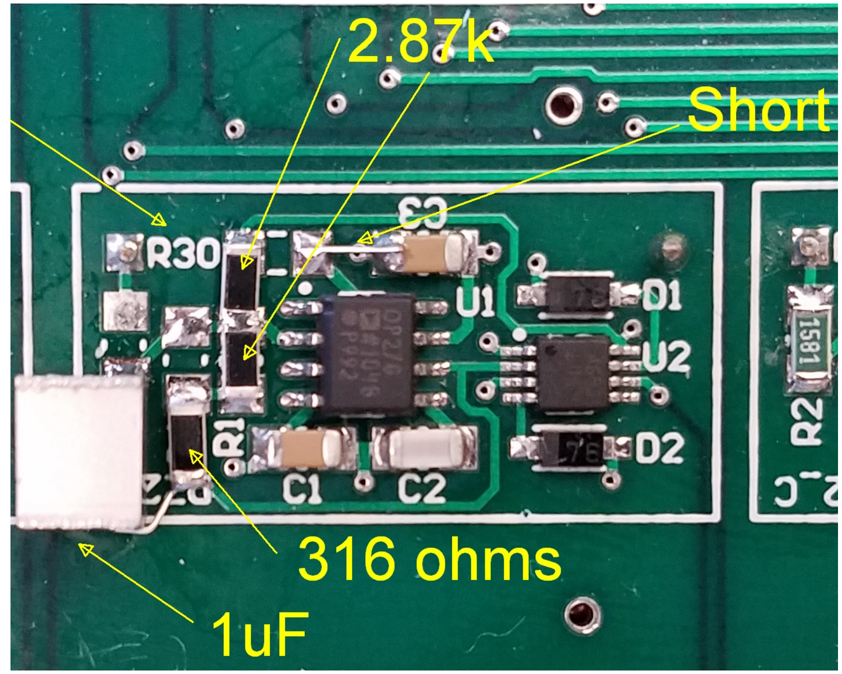
Two-take-aways:

1. **This 1.06 : 0.103 MHz response is a part of each filter stage**, so it comes and goes, depending on whether you’re using the stage (you need a “high-frequency pole” for each stage you use)
2. **Very important lesson for measurement taking and fitting:** know where the zeros and poles are ahead of time, so you don’t end up fitting for them with data that doesn’t sufficiently cover their frequency region (fitting for zeros or poles outside of data frequency region = BAD, prone to errors)

II.3 Review of the Circuit: (d) The New Low Pass

Another documentation doozy… images from [E1600252](E1600252)

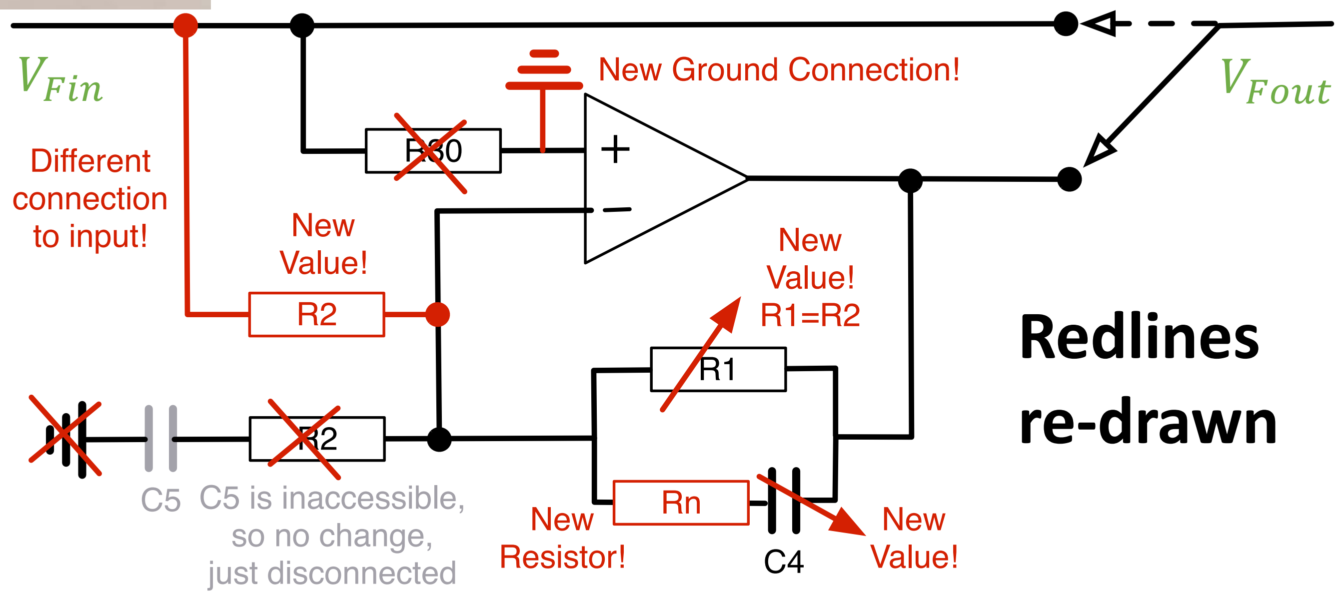




Seriously: While I chide Rich Abbott for his documentation, this change is indicative of his true engineering brilliance, and artistic mastery of circuits.

He's a busy man, just getting the job *done*.

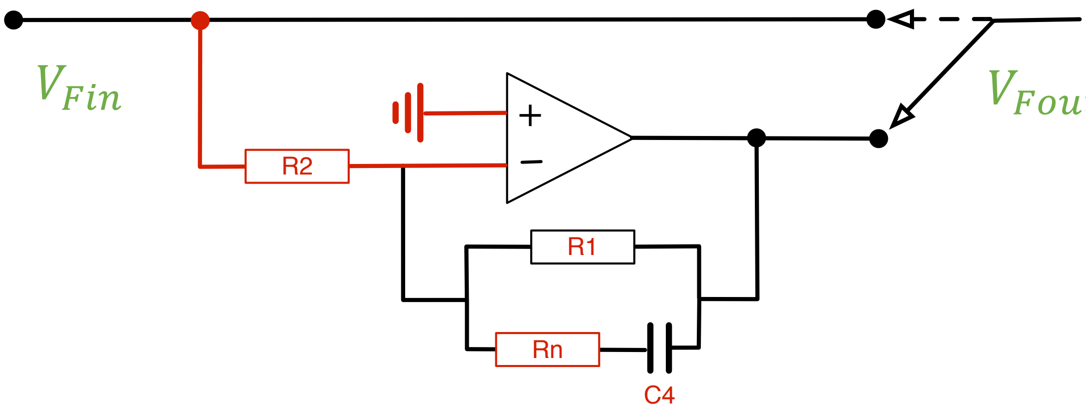
It's *literally* only us who care at this level.



Redlines re-drawn

II.3 Review of the Circuit: (d) The New Low Pass

Redlines re-drawn again, shown with functionally equivalent design intent as an inverting op-amp.



Now "easy" to analyze:

$$Z_1 = \frac{R_1(R_n + 1/i\omega C_4)}{\left((R_1 + R_n) + 1/i\omega C_4\right)} = R_1 \frac{(1 + i\omega R_n C_4)}{(1 + i\omega(R_1 + R_n)C_4)}$$

$$\frac{V_{Fout}}{V_{Fin}} = -\frac{Z_1}{Z_2} = -\frac{R_1}{R_2}\frac{(1 + i\omega R_n C_4)}{(1 + i\omega(R_1 + R_n)C_4)}$$

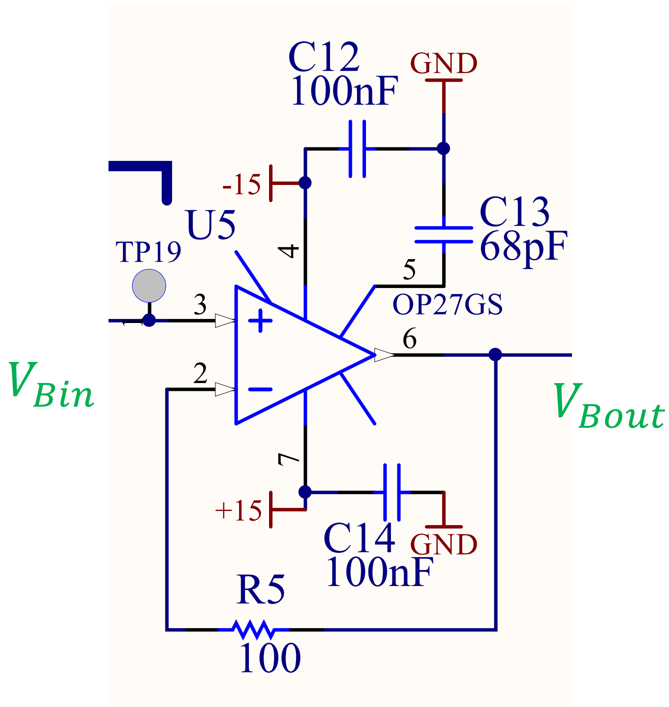
$$G_{ia}\Big|_{DC} = -\frac{R_1}{R_2} = -1.0$$

$$\boldsymbol{f_z^F = 1/(2\pi R_n C_4) = 503.655\ \text{Hz}}$$

$$\boldsymbol{f_p^F = 1/(2\pi(R_1 + R_n)C_5) = 49.954\ \text{Hz}}$$

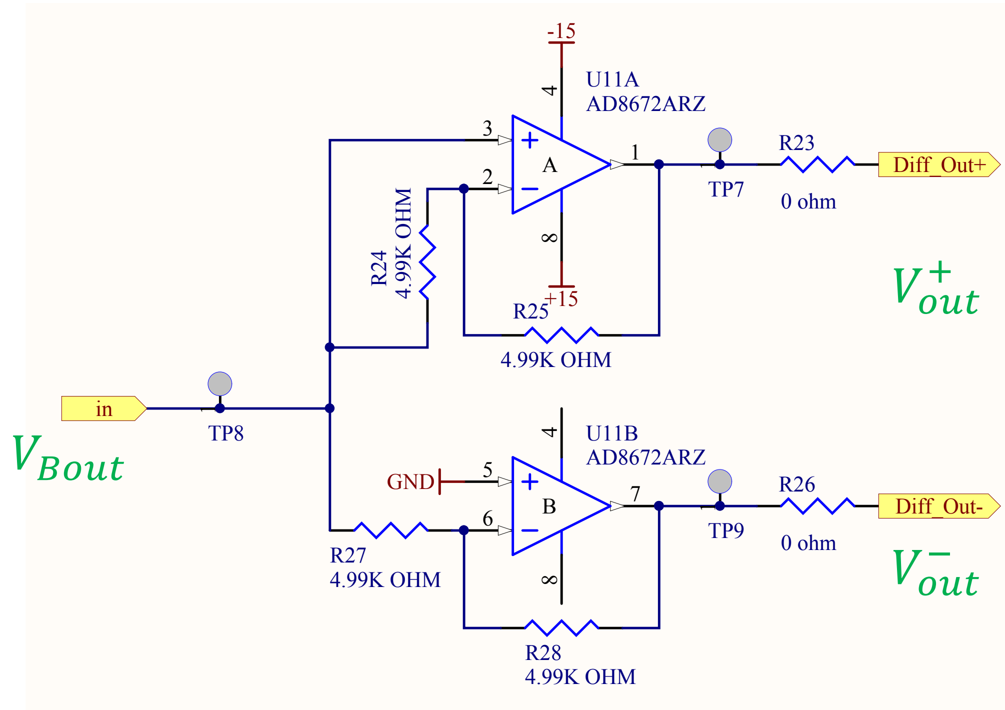
OK, cool. No sneaky high frequency poles here…

II.3 Review of the Circuit: (e) and (f) buff and SE-D amp



(e) is a standard gain buffer,

$$G_B = \frac{V_{Bout}}{V_{Bin}} = 1 + \frac{R_5}{\infty} = 1.0$$



(f) is response-free single-ended to differential amplifier

$$G_{oa} = \frac{(V_{out}^{+} - V_{out}^{-})}{V_{Bout}}$$

$$= \frac{V_{out}^{+}}{V_{Bout}} - \frac{V_{out}^{-}}{V_{Bout}}$$

$$= \frac{R_{25}}{R_{24}} - \left(-\frac{R_{28}}{R_{27}}\right)$$

$$\boldsymbol{G_{oa} = 2.0}$$

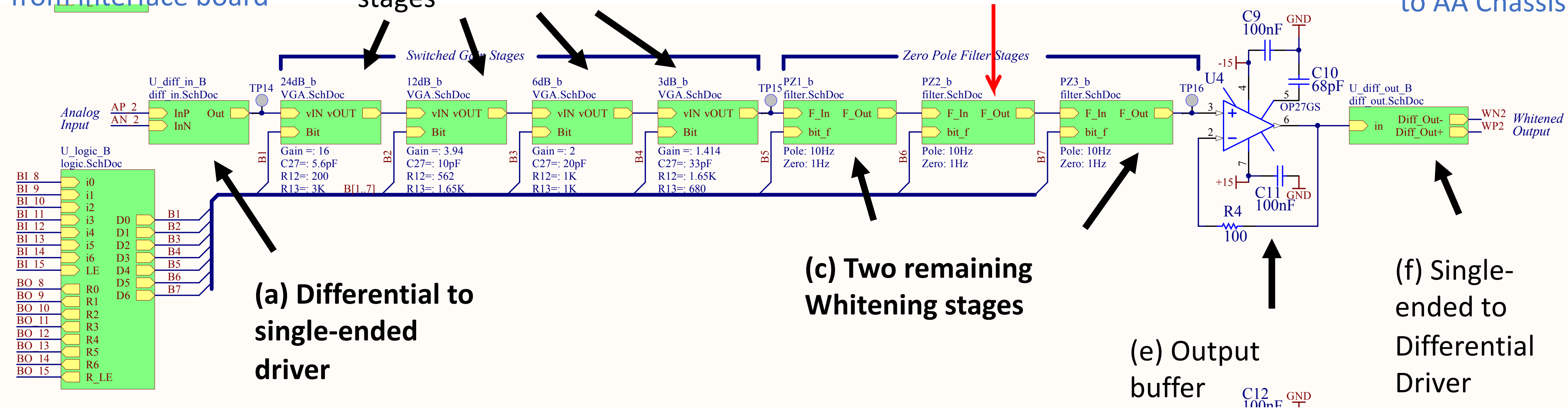
II.3 Review of the Circuit: Conclusions

Differential Input from interface board

(b) Four adjustable gain stages

(d) Filter Stage Modified into a Low Pass

Differential Output to AA Chassis



(a) $G_{in} \approx 0.99$

$$G_{ia}\Big|_{DC} = 0.498$$

$$f_p^{ia} = \mathbf{18.95\ kHz}$$

(b) $G = 0.4$

(e) $G_B = 1.0$

(f) $G_{oa} = 2.0$

(c) Two whitening stages available, Per stage you get:

$$f_z^F : f_p^F = \mathbf{0.960 : 10.073\ Hz}$$

$$f_z^F : f_p^F \approx 1.069 : 0.106\ \text{MHz}$$

(d) One low-pass available:

$$f_z^F : f_p^F = \mathbf{503.655 : 49.954\ Hz}$$

The overall gain (differential in, differential out) is therefore

$$G_{overall} = G_{in}\, G_{ia}\Big|_{DC}\, G\ G_B\, G_{oa} \approx 0.4$$

Outline

Two Parts, each quite long. *sigh*

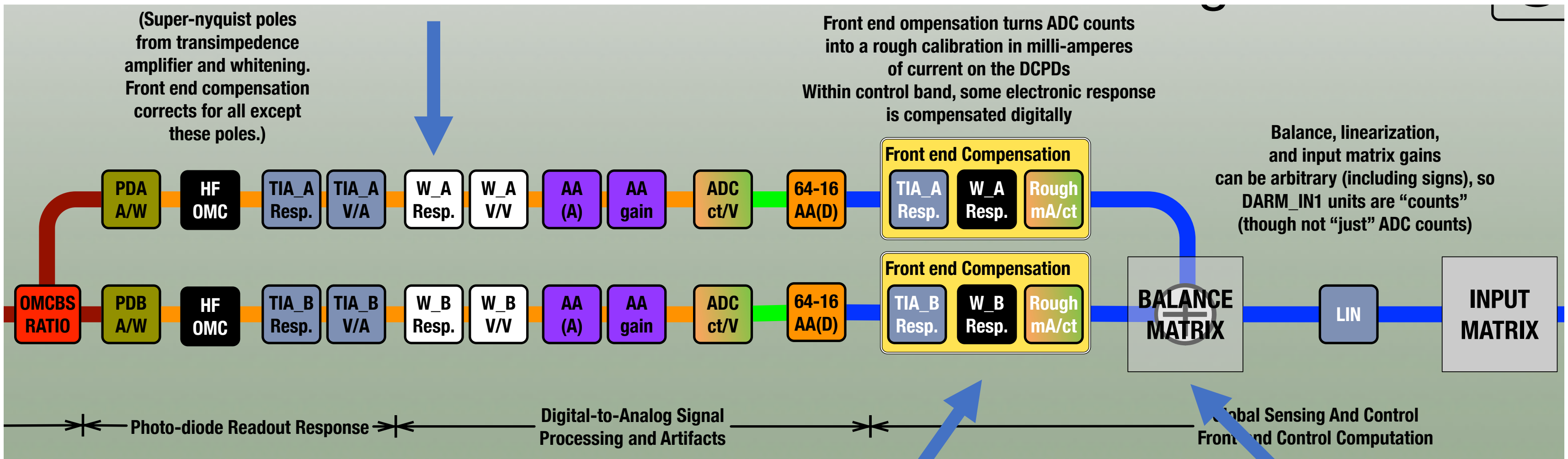
PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What’s the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.4 Compensation Scheme: Context

Using a zoom in on the relevant corner of the Subway Map G1501518, each channel's zeros and poles of OMC whitening chassis response is represented as the white "W_A Resp." and "W_B Resp." boxes



For the low frequency zeros and poles, i.e.

$$f_z^F : f_p^F = 0.960 : 10.073 \text{ Hz}$$

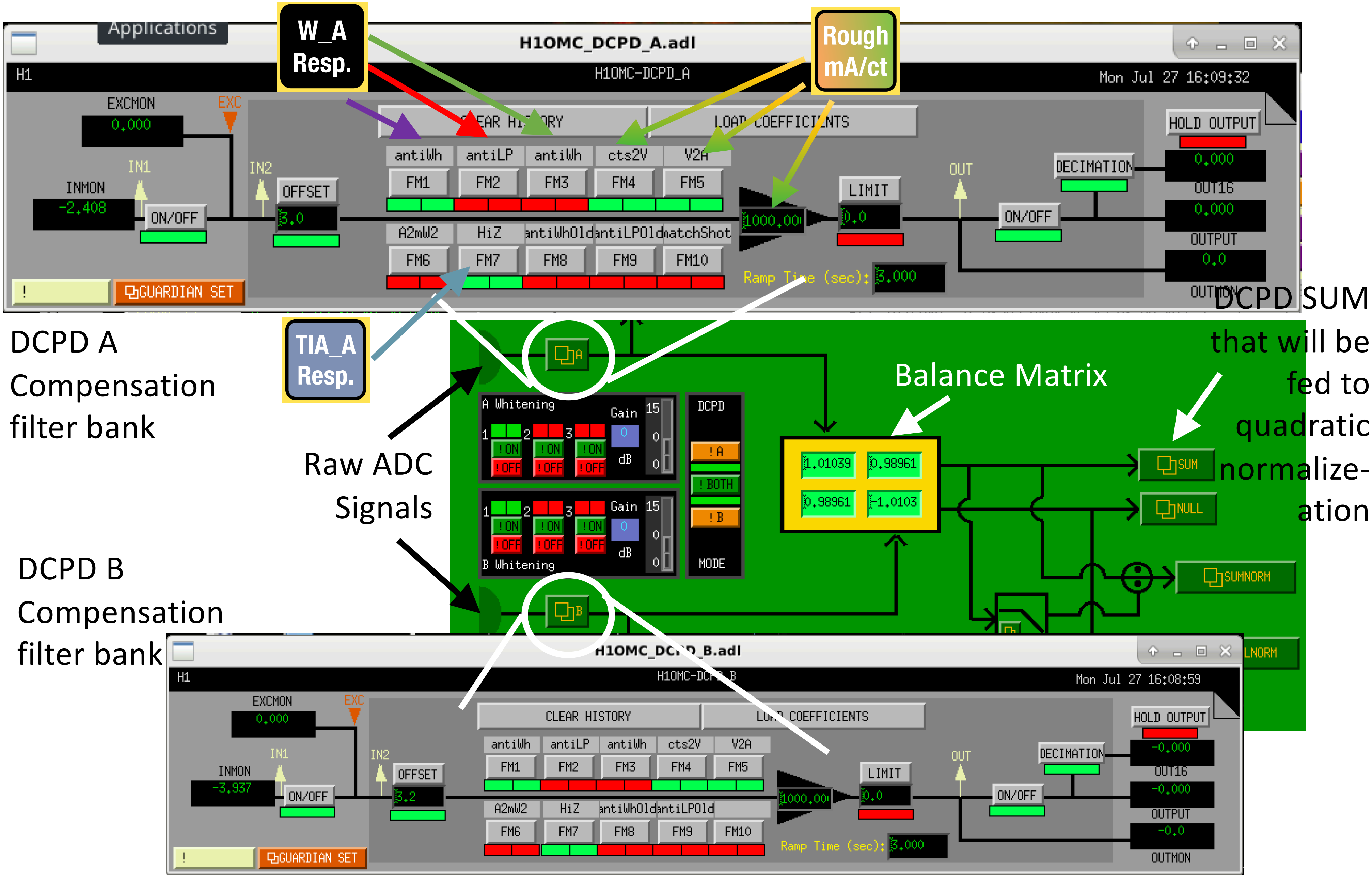
$$f_z^F : f_p^F = 503.655 : 49.954 \text{ Hz}$$

$$f_z^F : f_p^F = 0.960 : 10.073 \text{ Hz}$$

we "compensate" (multiply) the raw ADC signal with (by) digital filters "in loop," with digital IIR filters in the front-end shown as black boxes with the same name

The Balance Matrix digitally adds the two paths together, with coefficients that are *close* to 1.0 (not 0.5!) These take care of the *differential* gain between the channels

II.4 Compensation Scheme: Context



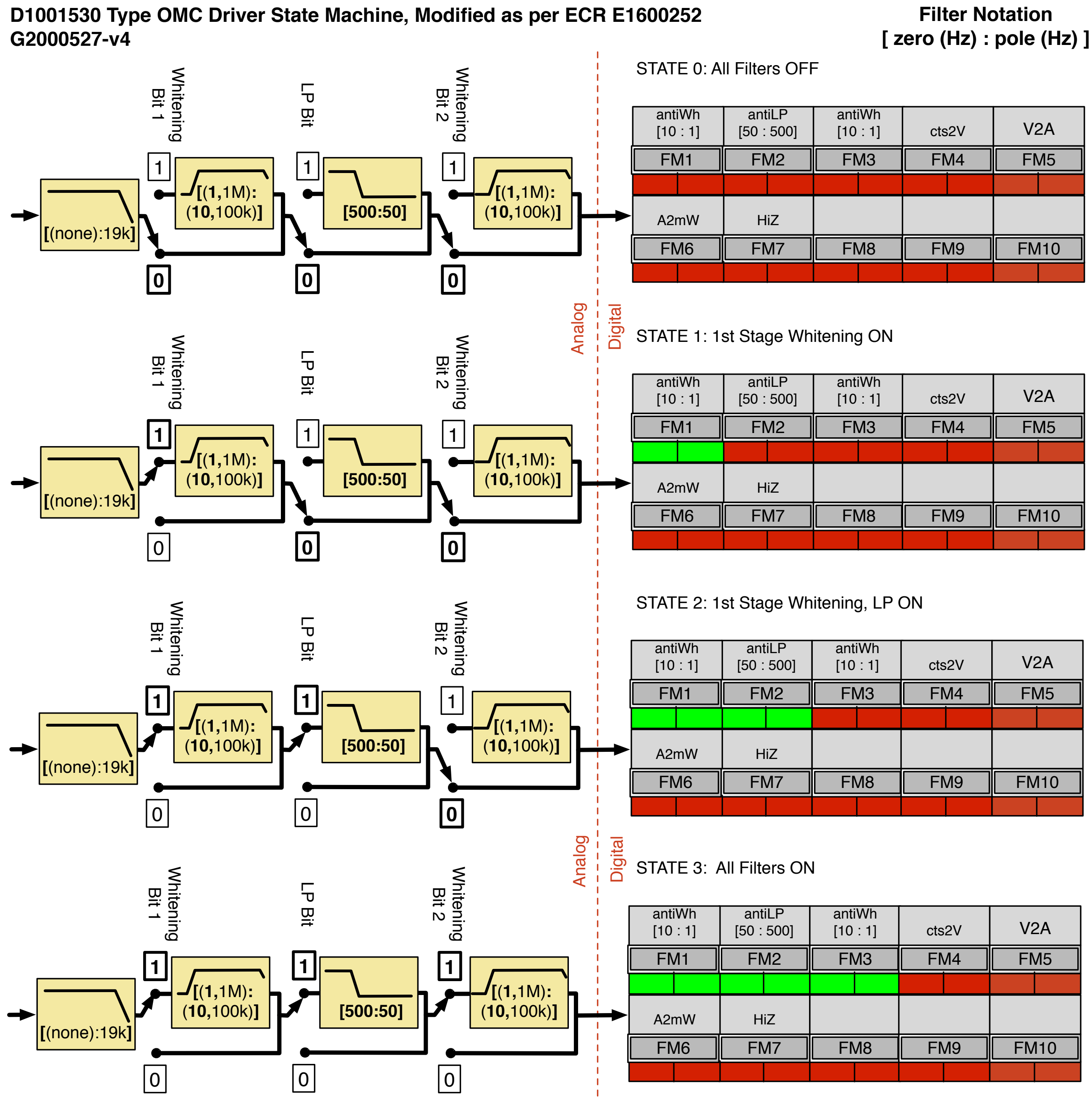
II. 4 Compensation Scheme: The switchable response

Here's a new state machine diagram to show what's is done to compensate the switchable filter response.

Notice, that the fixed 19 kHz pole and the 1 MHz:100 kHz zero:pole portion of the switchable response is not compensated.

These are the filters that Stefan updated quickly, and documented in LHO:47257.

(There are other filters in here to compensate for the switchable trans-impedance and other gain-only compensation, but not relevant to this discussion.)



II.4 Compensation Scheme

For all the high-frequency response values,

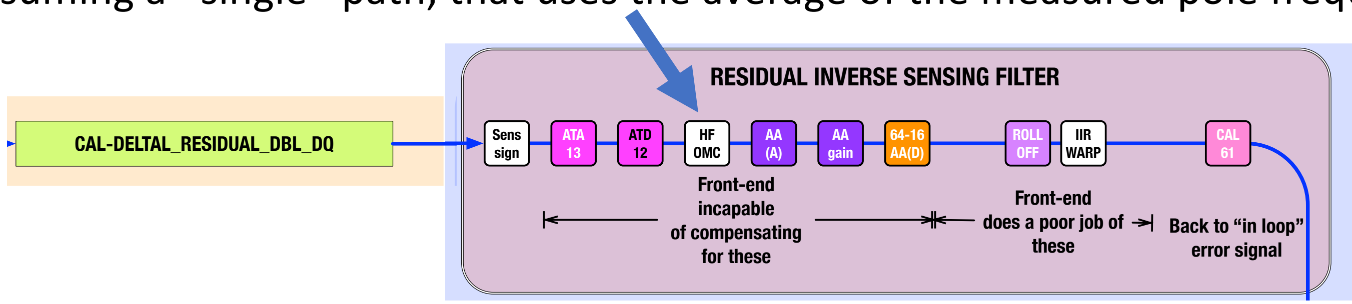
$$f_p^F = 0.106 \text{ MHz}$$
$$f_p^{ia} = 18.95 \text{ kHz}$$
$$f_p^F = 0.106 \text{ MHz}$$

(Notice the 1 MHz zero is dropped from the discussion, since it is assumed to have negligible impact in the response below 1 kHz)

the front-end code isn't running fast enough

- clock cycle is 61 usec,
- i.e. a rate of 16.384 kHz,
- thus a Nyquist frequency of 8.192 kHz,
- but with IIR filter warping, practically that means zeros and poles can't be accurately represented above ~5 kHz

Therefore we must rely on the acausal GDS (or DCS) pipelines to compensate these poles.
We do so assuming a *single* path, that uses the average of the measured pole frequencies from each channel.



pyDARM creates this filter, informed by the model parameter files …

Now you know what we'll be comparing our *new* fit against, and you've got a good handle on how things fit in to the bigger compensation picture. So let's get to fitting, and estimating the systematic error.

Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

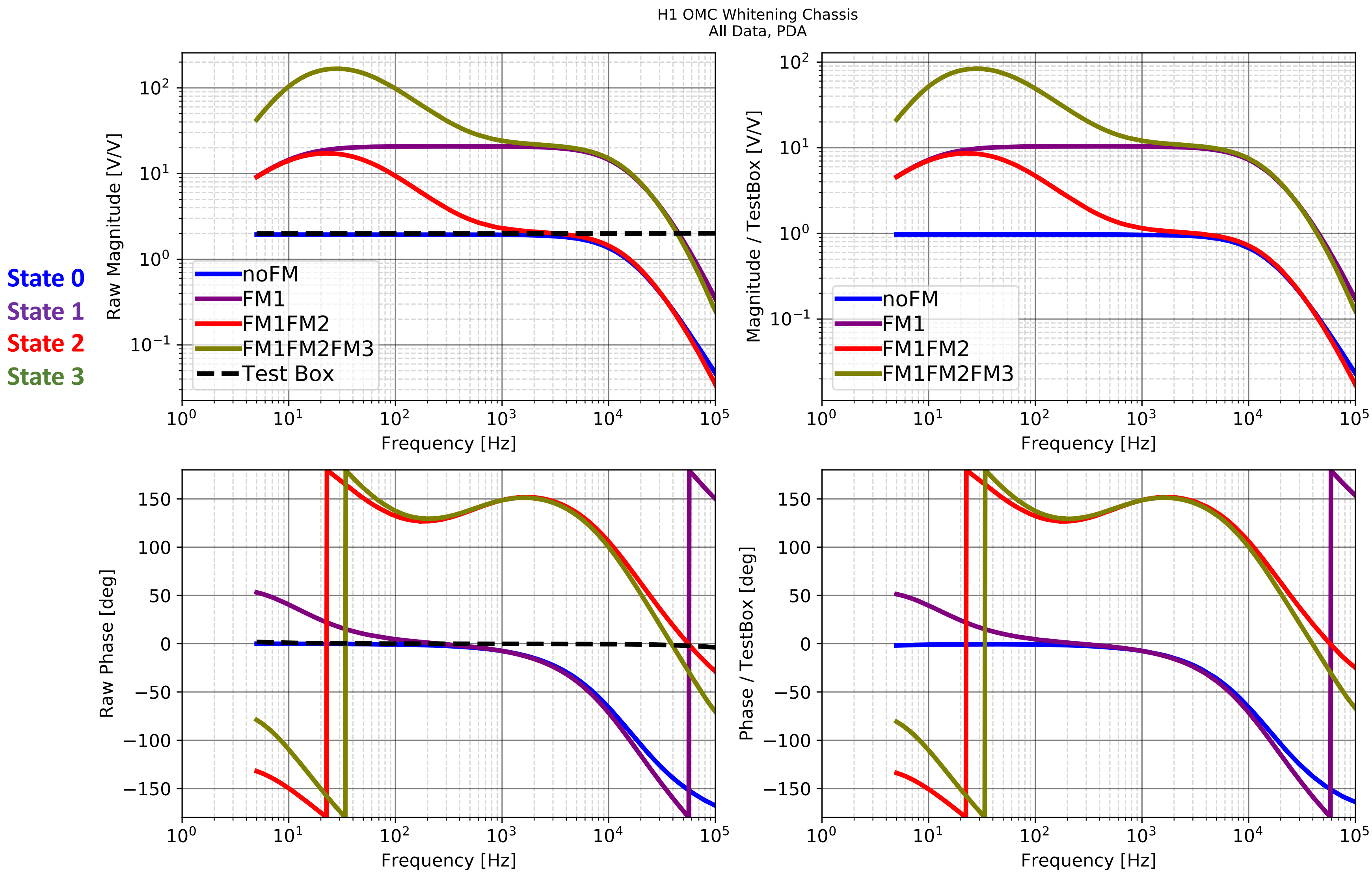
1. What's the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.5 Fitting and Predicting Systematic Error

- Order of operations for fitting and predicting the error:
  A. For each of the two DCPD Channels:
     i. Import measurement (which are of cascaded application of filtering), divide by test box
     ii. Take ratios of measurements to obtain individual stage response
     iii. Fit individual stages, obtain expected zeros and poles
     iv. Stack fit zeros:poles to re-create cascading application of filtering, compare against measurement to create residuals
     v. Reconstruct residuals of compensation we actually used (both in front-end for LF and GDS for HF)
  B. Then, for each filter state
     i. Sum each paths residuals (use "balance matrix" for compensation version, and apply HF poles)
     ii. Create a residual (both with existing compensation, and with fit)
     iii. Take the ratio of State 1 (one Whitening Stage – i.e. configuration after Mar 10), and State 3 (both whitening stages and the low pass – i.e. what we had for all prior times in O3)
  C. Finally, estimate response function error
     i. Use ratio of State 1 to State 3 as "η_{OMCWC}" for C
     ii. Compute reference R from 20200103 model, then a modified R using C* η_{OMCWC}
     iii. FINAL ANSWER ESTIMATE Take ratio of R_η_{OMCWC} / R_reference

II.5 Fitting and Predicting Systematic Error

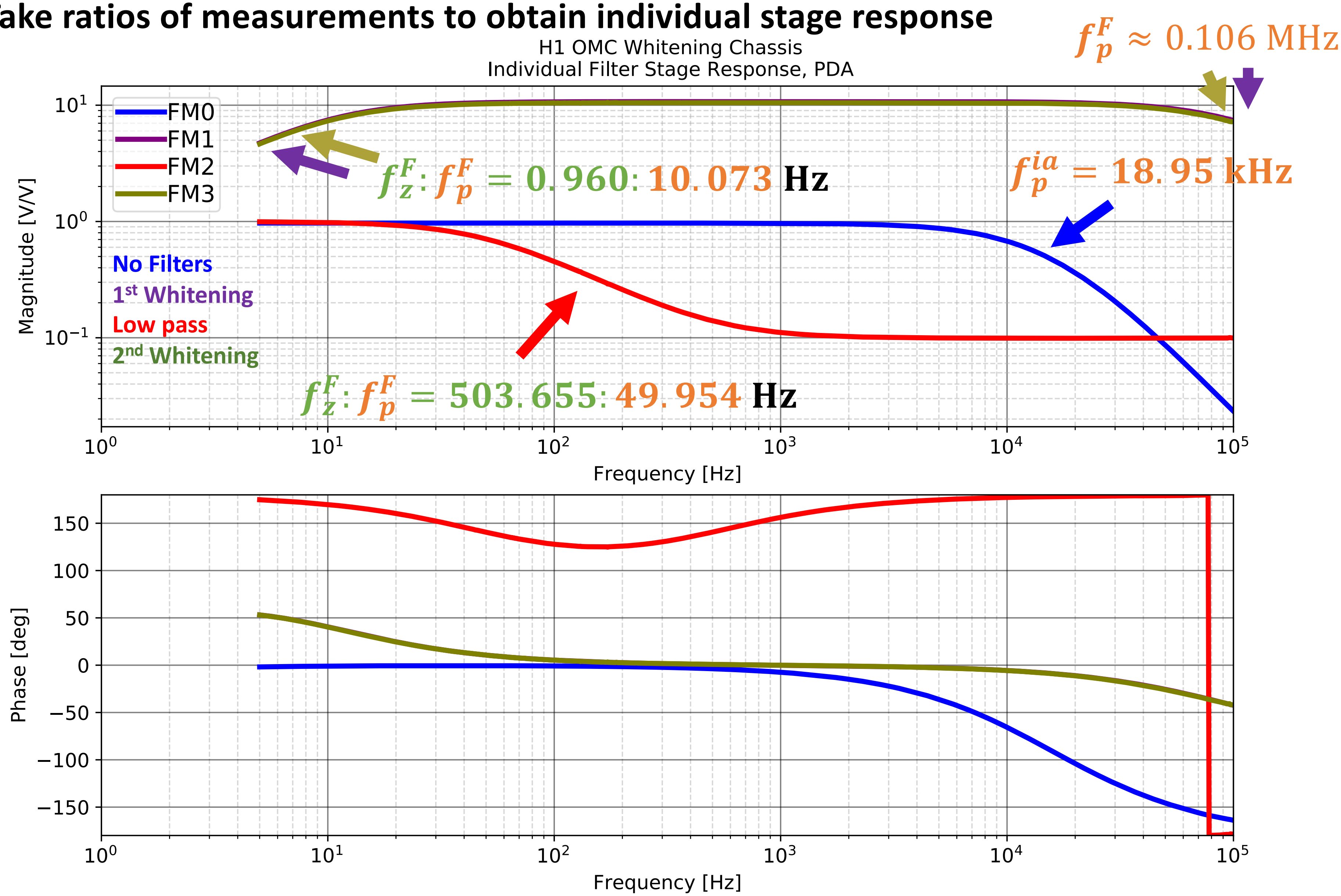
A.i. Import measurement (which are of cascaded application of filtering), divide by test box



This (the right panels) is the data that Lilli fit in LHO:47358. In fact, we really only took the results from the green curve (i.e. State 3). **One can see how a fitting program might get confused**, having to fit for the poles and zeros of all *four* states simultaneously… and **note the frequency vector only spans 5 Hz to 100 kHz**.

II.5 Fitting and Predicting Systematic Error

A.ii. Take ratios of measurements to obtain individual stage response

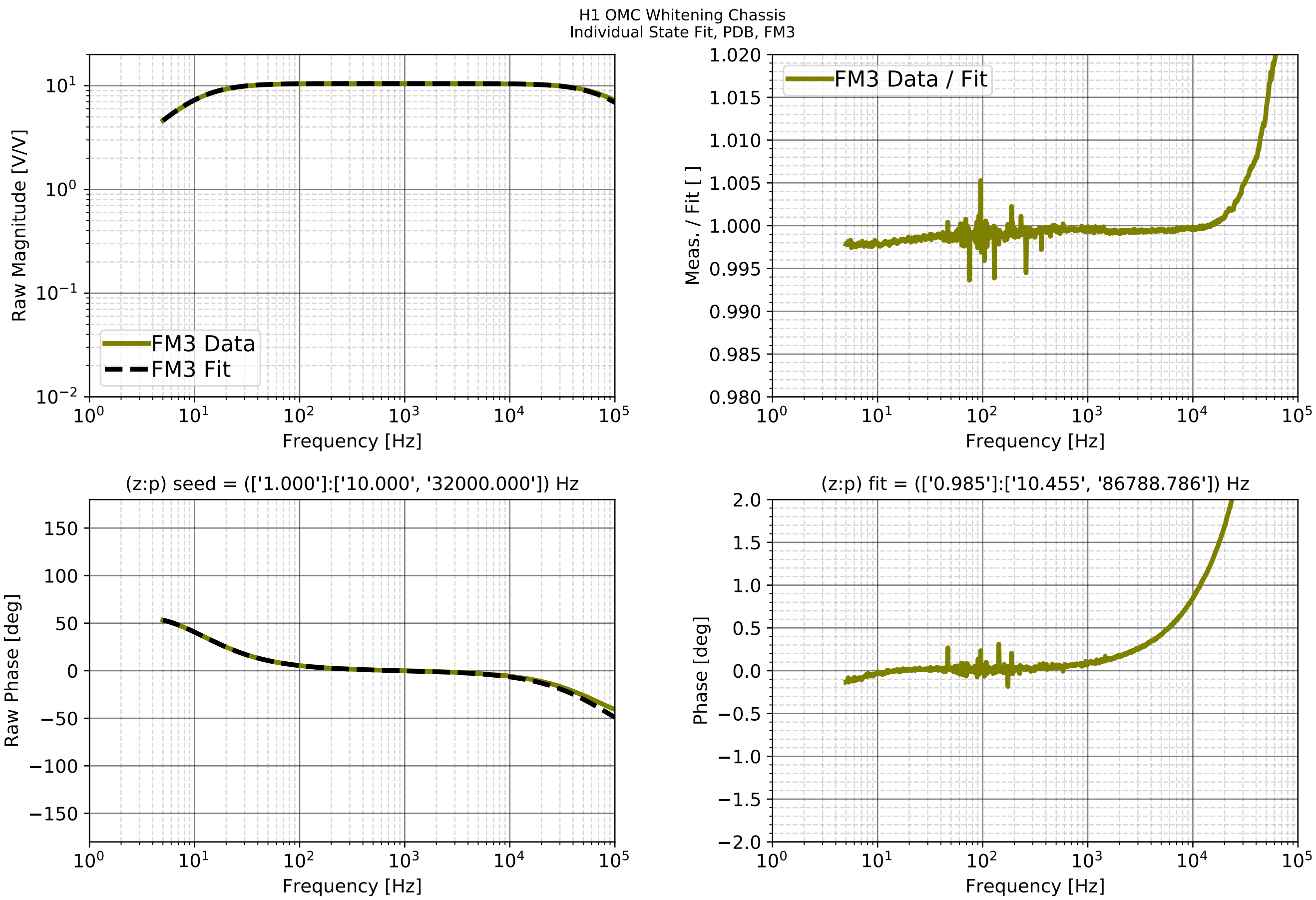


Ah, that's reassuringly simple. BUT – we see much more clearly that the data only goes down to 5 Hz and up to 100 kHz, so fitting for the 1 Hz zero and 0.1 MHz pole will be error prone.

So again, we need better data, but we'll proceed with what we have…

II.5 Fitting and Predicting Systematic Error

A.iii. Fit individual stages, obtain expected zeros and poles



2nd Whitening Stage, PDB

Expected:

$f_z^F : f_p^F = 0.960 : 10.073$ Hz

$f_p^F \approx 0.106$ MHz

Previous Results:

$f_z^F : f_p^F = 1.000 : 10.467$ Hz

$f_p^F = 0.0328$ MHz

$G = 0.9973$

Seed (PDB):

$f_z^F : f_p^F = 1.0 : 10.0$ Hz

$f_p^F = 0.032$ MHz

New Results (PDB):

$f_z^F : f_p^F = 0.985 : 10.455$ Hz

$f_p^F = 0.868$ MHz

Fits are now much more simple – but the fit for the two whitening stages are limited because they're trying to fit for zeros or poles outside the data's frequency region.

At the low end, **given incomplete data for the 1 Hz, the fitter yields a residual with ~0.25% error in magnitude.**

I think this is why Stefan felt the need to add a little gain modifiers to each compensation filter as reported and installed in LHO:47257.

II.5 Fitting and Predicting Systematic Error

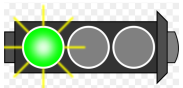
A.iii. Fit individual stages, obtain expected zeros and poles

PDA	New Result Zeros	Old Result Zeros	New Result Poles	Old Result Poles	Old Result Gain
Fixed Response	0.2057		0.0489, 13612.7, 17873.2	11346	
1st Whitening Stage	0.9894	0.969	10.4215, 98277.1	10.440, 32875.3	G = 0.9734
Low Pass	501.119	497.5	49.7277	49.63	G = -0.9995
2nd Whitening Stage	0.9677	0.9865	10.3377, 86258.7	10.372, 32875.3	G = 1.0020

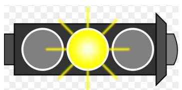
PDB	New Result Zeros	Old Result Zeros	New Result Poles	Old Result Poles	Old Result Gain
Fixed Response	0.2067		0.0505, 13707.1, 18046.0	11521	
1st Whitening Stage	0.979	0.966	10.1639, 98413.7	10.160, 32863	G = 0.975458
Low Pass	501.790	497.7	49.8120	49.72	G = -1.0004
2nd Whitening Stage	0.9845	1.000	10.4551, 86788.8	10.467, 32683	G = 0.9973

II.5 Fitting and Predicting Systematic Error

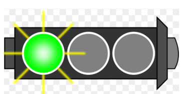
A.iii. Fit individual stages, obtain expected zeros and poles



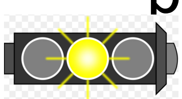
- Most low frequency zeros and poles are very similar,



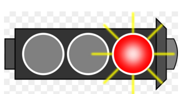
- BUT new fit takes out DC gain from State 0 with nearly cancelling zero:pole pair at ~0.20:0.05 Hz; old fit did it with a gain on each of the State 1, 2, and 3 fits.
  - Which is better? Dunno. Seems like new fit might be conceptually better, but here's where we'd need data down to (past) 1 Hz.
  - Preferably, get data down to 0.5, 0.2, 0.1… for whatever patience allows, and fit again.



- New fit does a *much* better job at fitting for the 0.1 MHz pole



  - (though the State 3 fit still reports ~20 kHz too low)
  - Would be nice to get data out to 200 kHz, or to see if "ratio of State 3 / State 2" data we used agrees with a new measurement of "only State 3 on" but probably not worth chasing



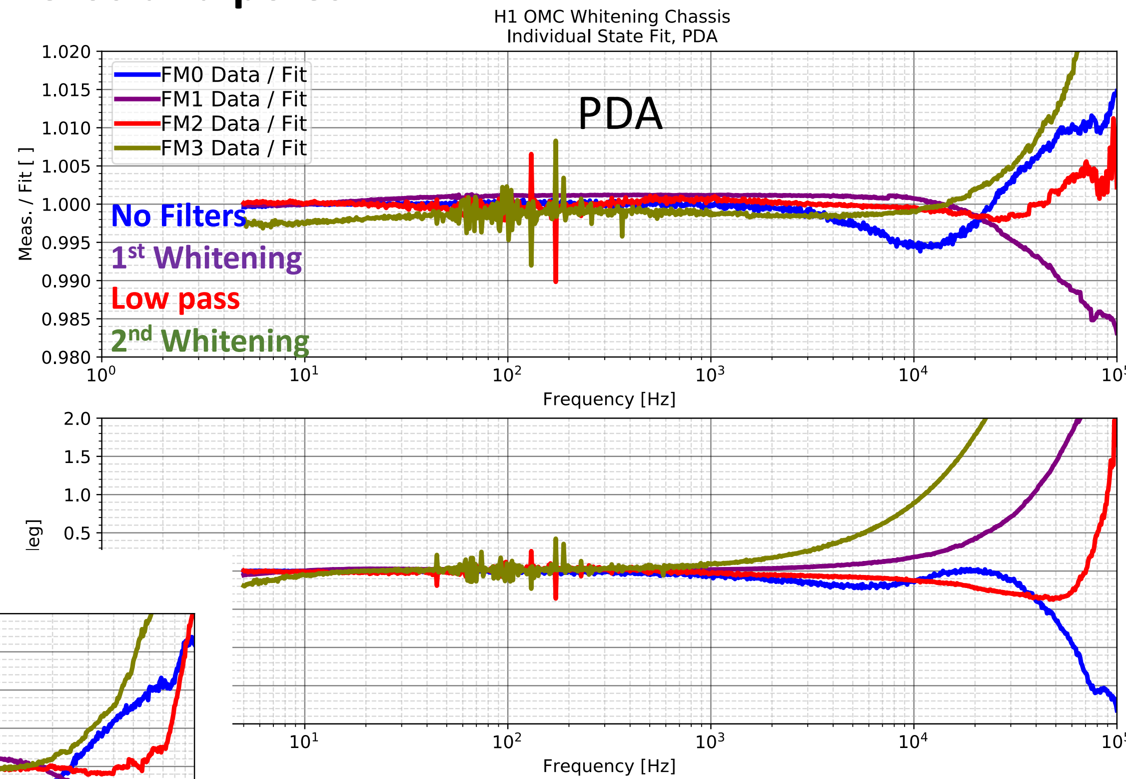
- Both old and new fit for State 0 have a high frequency pole at 11 or 13 kHz, respectively, where there shouldn't be
  - What's going on here? Should it be included? Dunno…

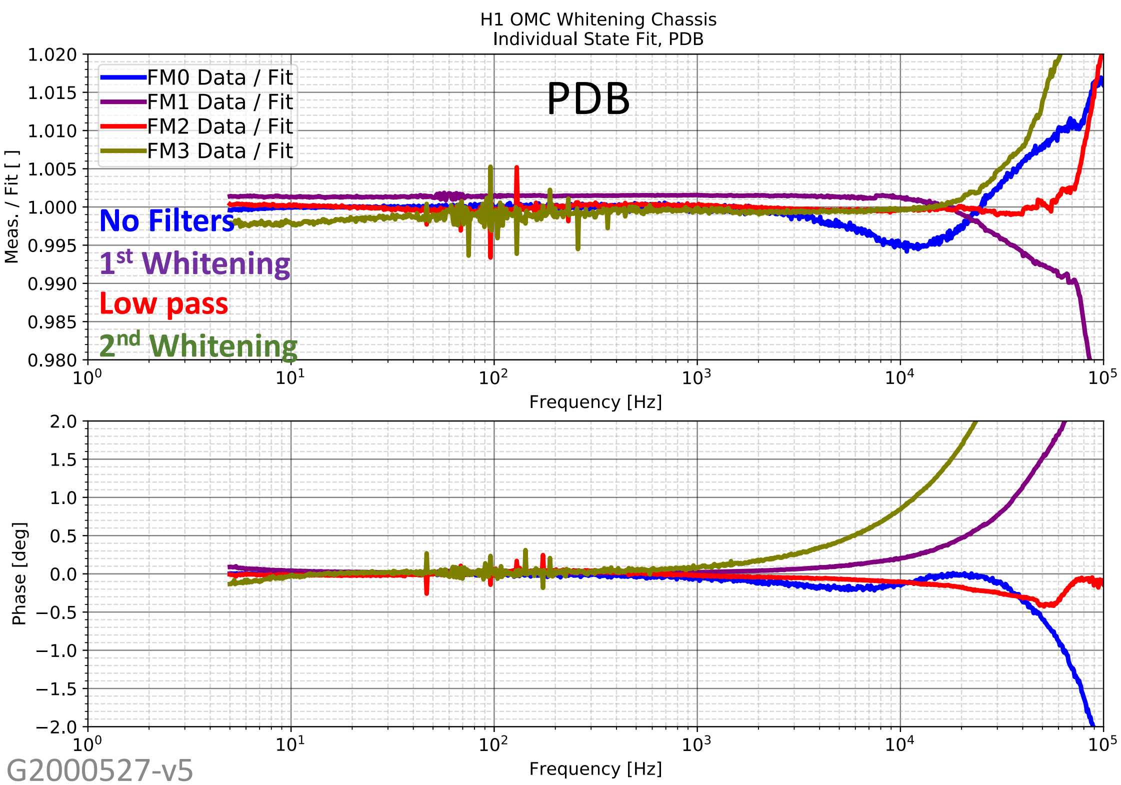
II.5 Fitting and Predicting Systematic Error

A.iii. Fit individual stages, obtain expected zeros and poles

Here's the residuals between each stage and the fit for that stage.

Notice that the fits are pretty dang awesome – but the 1:10 Hz whitening stages have low-frequency gain issues. **Again: this is the result of incomplete data.**





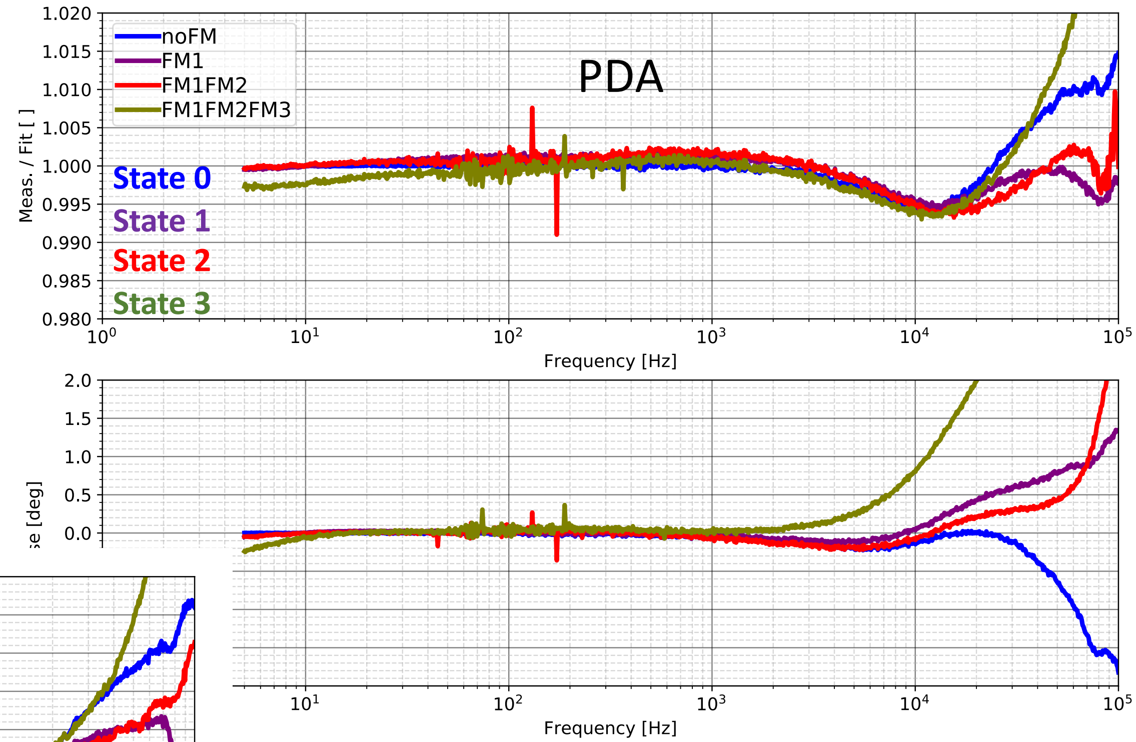
But alas. We move on with the plan, 'cause maybe "perfect is the enemy of good enough."

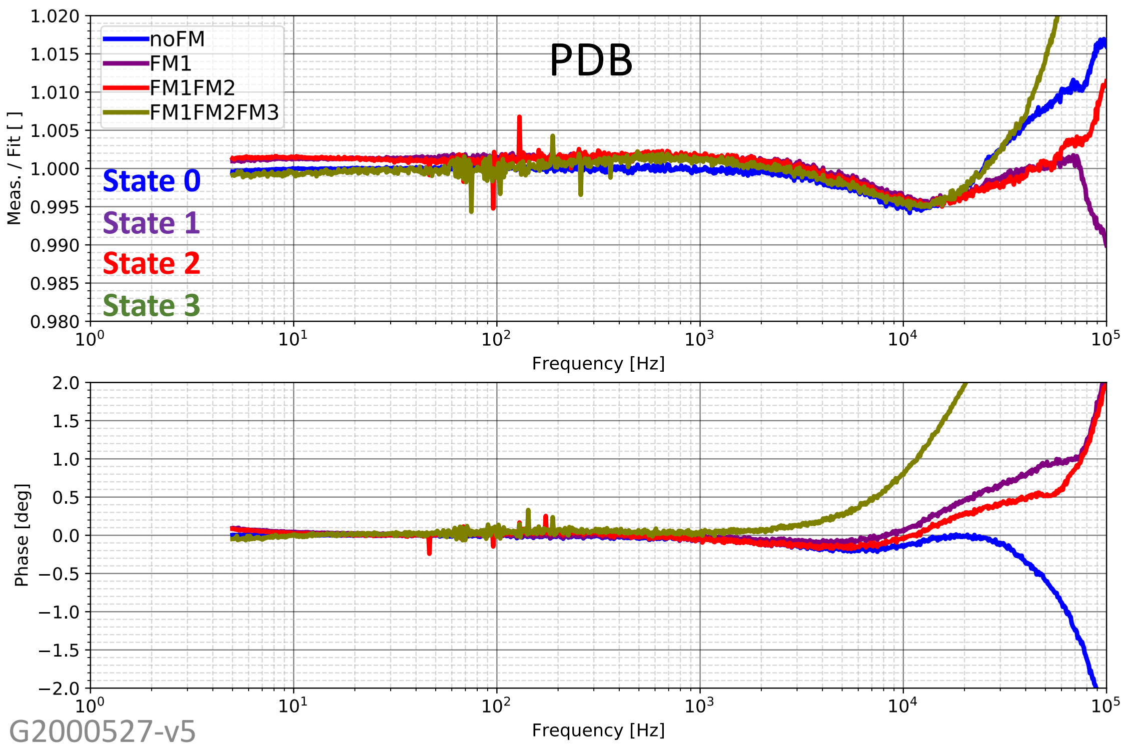
II.5 Fitting and Predicting Systematic Error

A.iv. Stack fit zeros:poles to re-create cascading application of filtering, compare against measurement to create residuals

Take the poles and zeros from the fit each stage, collect them, successively adding them to a new model filter of each state, and divide that cascaded model against the original data

We see how the lack of low frequency data for the whitening stages impacts this residual – especially for state 3, which has the error from both the 1st and 2nd whitening stage





… but the magnitude residual is still only at most 0.0025 away from unity at 10 Hz, and closer than that for most frequencies until 5 kHz.

So now we have a good model of the measurement.
Is it better than the old model that Stefan and Lilli created and installed?

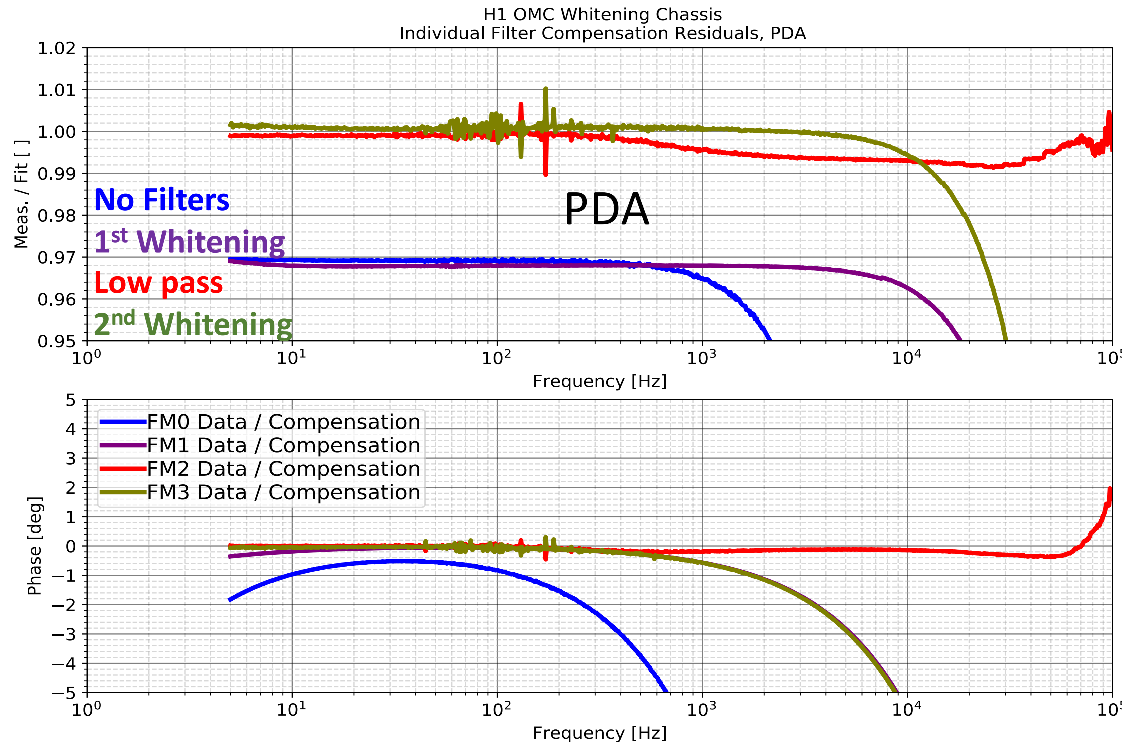
II.5 Fitting and Predicting Systematic Error

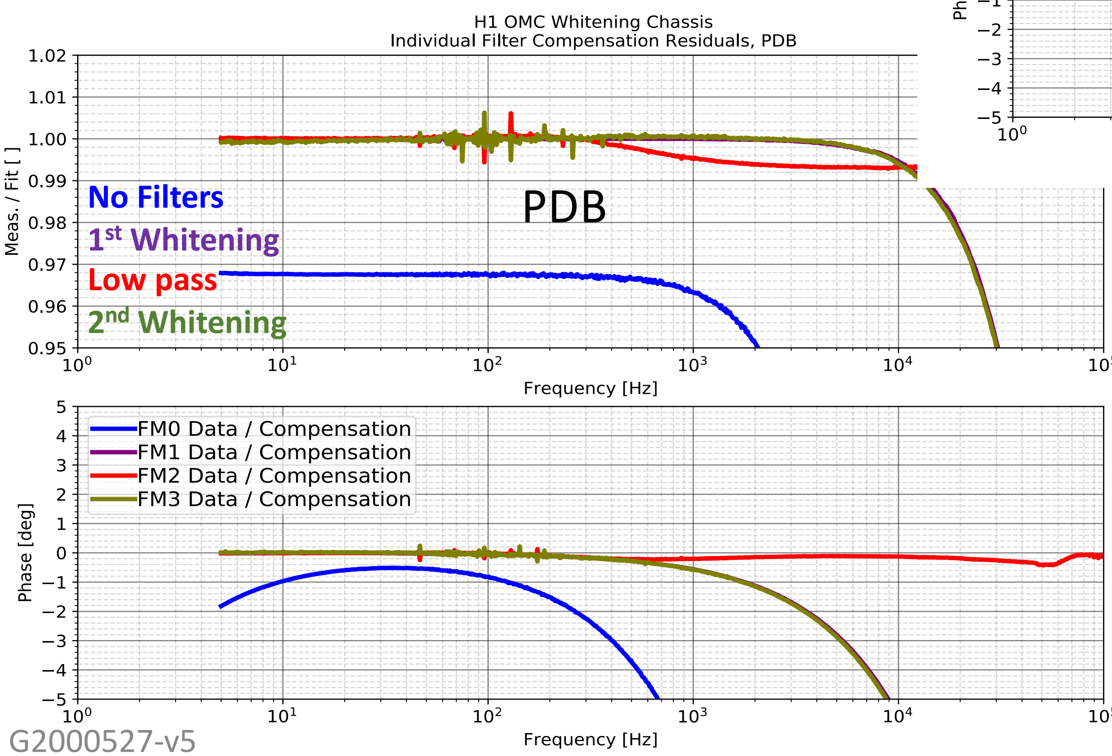
A.v. Reconstruct residuals of compensation we actually used (both in front-end for LF and GDS for HF)

This pulls the LF zeros and poles from Stefan's work.

It does *not* yet include HF poles

The residual is each individual stage's measured response / compensation FM response





NOTE THE YLIM SCALE CHANGE

- **PDA has a 3% gain error in it's 1st whitening stage compensation, see rabbit hole**
- "No filters" response is clearly missing the 19 kHz pole, fine – we compensate that later. Fine.
- 2nd stage compensation is < 0.1% away from unity magnitude. Interesting.

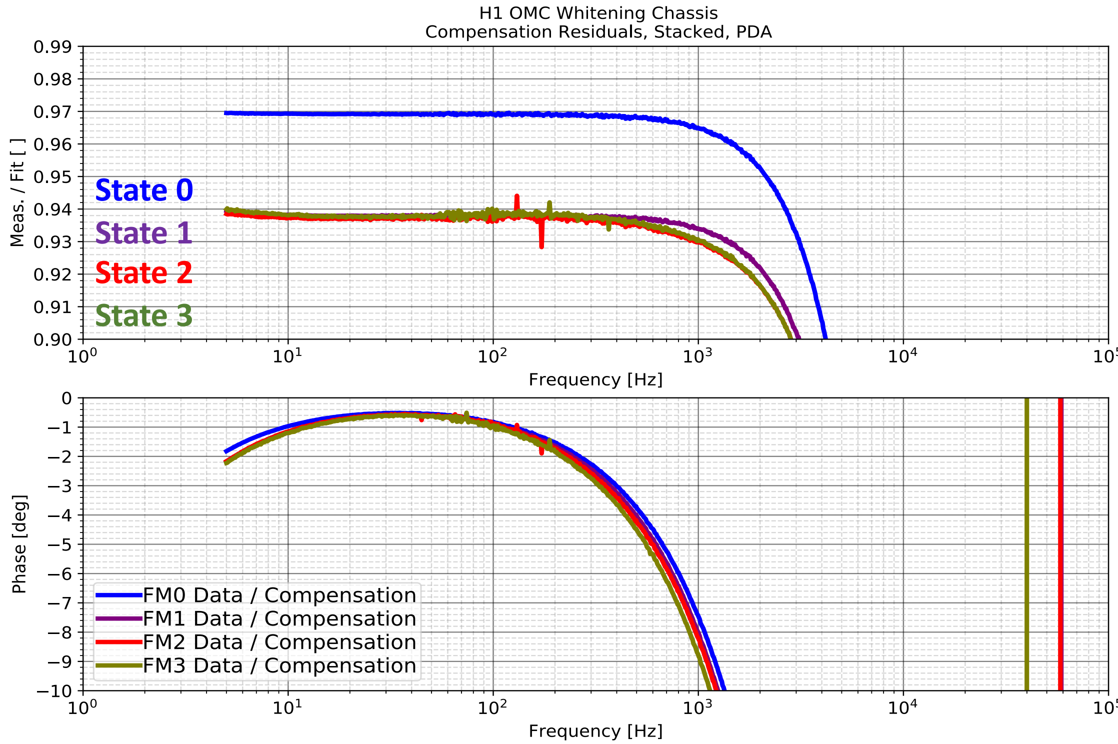
II.5 Fitting and Predicting Systematic Error

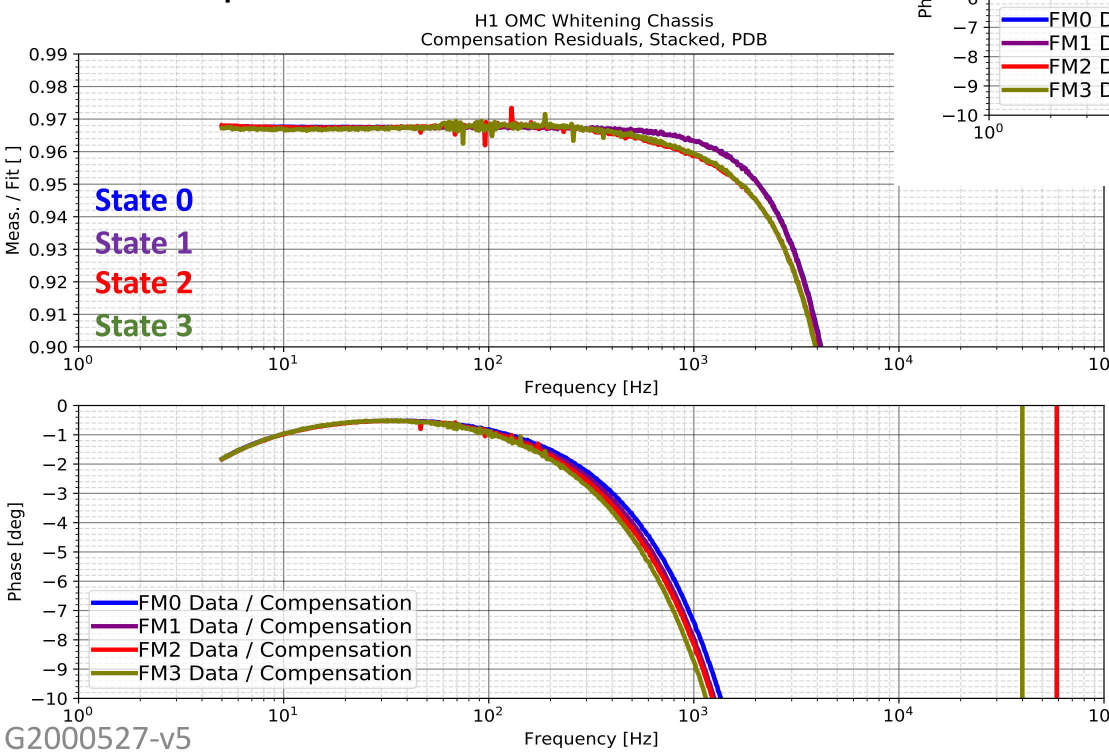
A.v. Reconstruct residuals of compensation we actually used (both in front-end for LF and GDS for HF)

Here's the "stacked" version…

One can see the "DC Gain" error is stacking up – **especially, strangely the 1st whitening stage of PDA**

but the magnitude response is still "flat" up to ~1kHz





Phase is rolling off quickly, but that's the missing compensation for 19 kHz pole.

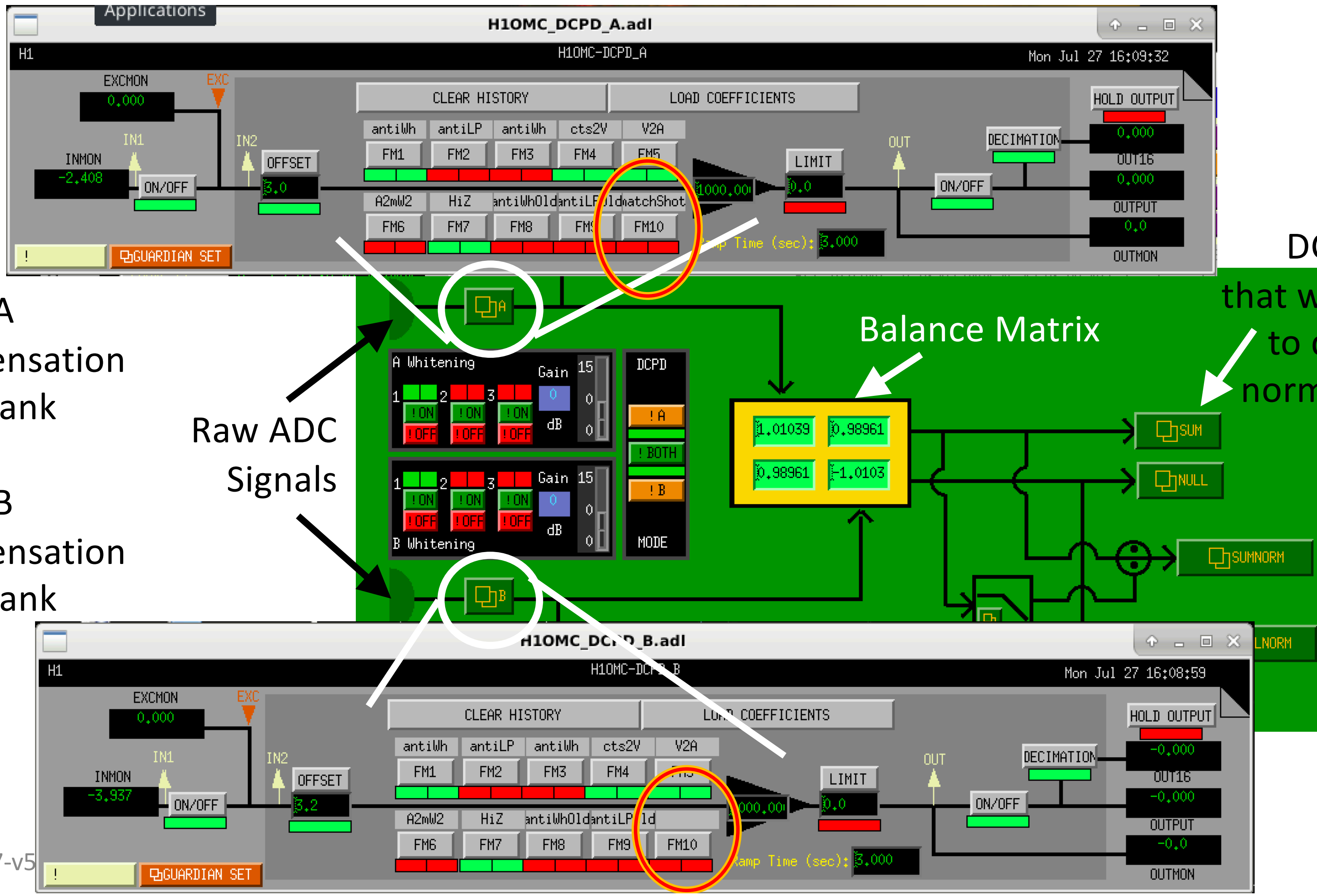
This gets applied when I create the sum (because that's how it's done in the current calibration scheme)

II.5 Fitting and Predicting Systematic Error

B.i. For each (stacked) filtering stage, sum each paths residuals (use “balance matrix” for compensation version and apply average HF poles)

Dude – you don’t think the rabbit hole is deep enough yet???

Let’s talk about the balance matrix and how the A and B paths are matched.



II.5 Fitting and Predicting Systematic Error

B.i. For each (stacked) filtering stage, sum each paths residuals (use "balance matrix" for compensation version and apply average HF poles)

- Wednesday, Dec 2018, Stefan develops methodology to populate the balance matrix: LHO:45734
- Friday, Mar 01 2019, Stefan updates and augments his methodology documentation (nice!): LHO:47217
- Mar 01 2019, (same Friday) Stefan updates the balance matrix **and turns on FM10 = gain of 0.977**
  - 2019-03-01 22:08 UTC FM10 turned on (Friday at 2pm)
- Saturday, Mar 02 2019 Craig & Stefan discover that the balance is (ah!) DCPD light level dependent! LHO:47247
- Sunday Mar 03 2019, Rich and Peter debug and fix the OMC Whitening Chassis, LHO:47254
- 2019-03-04 02:35 UTC FM10 is turned off and *never used again* (Sunday at 5:30 pm).
- Monday Mar 04, Stefan aLOGs that he's updated the compensation after Rich/Peter chassis fix: LHO:47257, but says:
  - "Finally, we still will have to re-match the photo diode light levels in lock (alog 47217)."
- **…. but I don't think this "re-match" to the photodiode light level -- i.e. re-compensate for the DCPDA channel 0.977 errant gain was ever done – or maybe it was encorporated in to the first FMs?!**

II.5 Fitting and Predicting Systematic Error

B.i. For each (stacked) filtering stage, sum each paths residuals (use “balance matrix” for compensation version and apply average HF poles)

So, in conclusion, what the compensation scheme really should be is:

$$\boldsymbol{Sum_{real} = G^{A}_{overall} * meas^{A}_{LF} * meas^{A}_{HF} * comp^{A}_{LF} * comp^{A}_{HF}\, G^{A}_{bal}}$$

$$\boldsymbol{+\, G^{B}_{overall} * meas^{B}_{LF} * meas^{B}_{HF} * comp^{B}_{LF} * comp^{B}_{HF}\, G^{B}_{bal}}$$

But what we’ve modelled in O3 assumes that in the above equation,

$$G^{A}_{bal} = G^{B}_{bal} = 1/2$$ Bad assumption

$$G^{A}_{overall} = G^{B}_{overall} = G_{overall} = G_{meas}$$ Pulled out / Folded in to H_C

$$meas^{A}_{LF} * comp^{A}_{LF} = meas^{B}_{LF} * comp^{B}_{LF} = 1$$ Bad assumption

$$comp^{A}_{LF} = comp^{B}_{LF} = comp_{HF} = (meas^{A}_{HF} + meas^{B}_{HF})/2$$ Bad assumption

$$\boldsymbol{Sum_{model} = comp_{HF}}$$ (without allowing for switchable response) Bad assumption

So let’s review the *values* for the existing compensation, so we can compare it against the fit.

II.5 Fitting and Predicting Systematic Error

B.i. For each (stacked) filtering stage, sum each paths residuals (use "balance matrix" for compensation version and apply average HF poles)

```
# DCPD SUM Balance Matrix Elements
# LHO aLOG 47228
# https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=47228
        #    PDA      PDB
balance = [1.01039,0.98961]

# Existing compensation:
# Low frequency --
# From LHO aLOG 47257
# and cross checked to be present in
# ^/trunk/Common/H1CalFilterArchive/h1omc/
# v    H1OMC_1254266332.txt
#                          "FM0"   FM1       FM2      FM3
compOMCzeros_whitening = [[ 1.0,  10.440,    49.63, 10.372], #PDA
                          [ 1.0,  10.160,    49.72, 10.467]] #PDB
compOMCpoles_whitening = [[ 1.0,   0.966,    497.5,  0.9865], #PDA
                          [ 1.0,   0.969,    497.7,  1.0000]] #PDB
compOMCgain_whitening  = [[ 1.0, 0.973400, -0.9995,  1.0020], #PDA
                          [ 1.0, 0.975458, -1.0004,  0.9973]] #PDB

# From LHO aLOG 47377
# which has been copied to, e.g. modelparams_H1_20200103.py
#                              PDA          PDB
uncompOMCpoles_whitening = np.array([(11.346e3 + 11.521e3)/2,
                                     (32.875e3 + 32.863e3)/2,
                                     (32.875e3 + 32.863e3)/2])
```

1st Whitening FM1 $f_z^F : f_p^F = 0.960 : 10.073$ Hz

Low pass FM2 $f_z^F : f_p^F = 503.655 : 49.954$ Hz

2nd Whitening FM3 $f_z^F : f_p^F = 0.960 : 10.073$ Hz

$f_p^{ia} = 18.95$ kHz

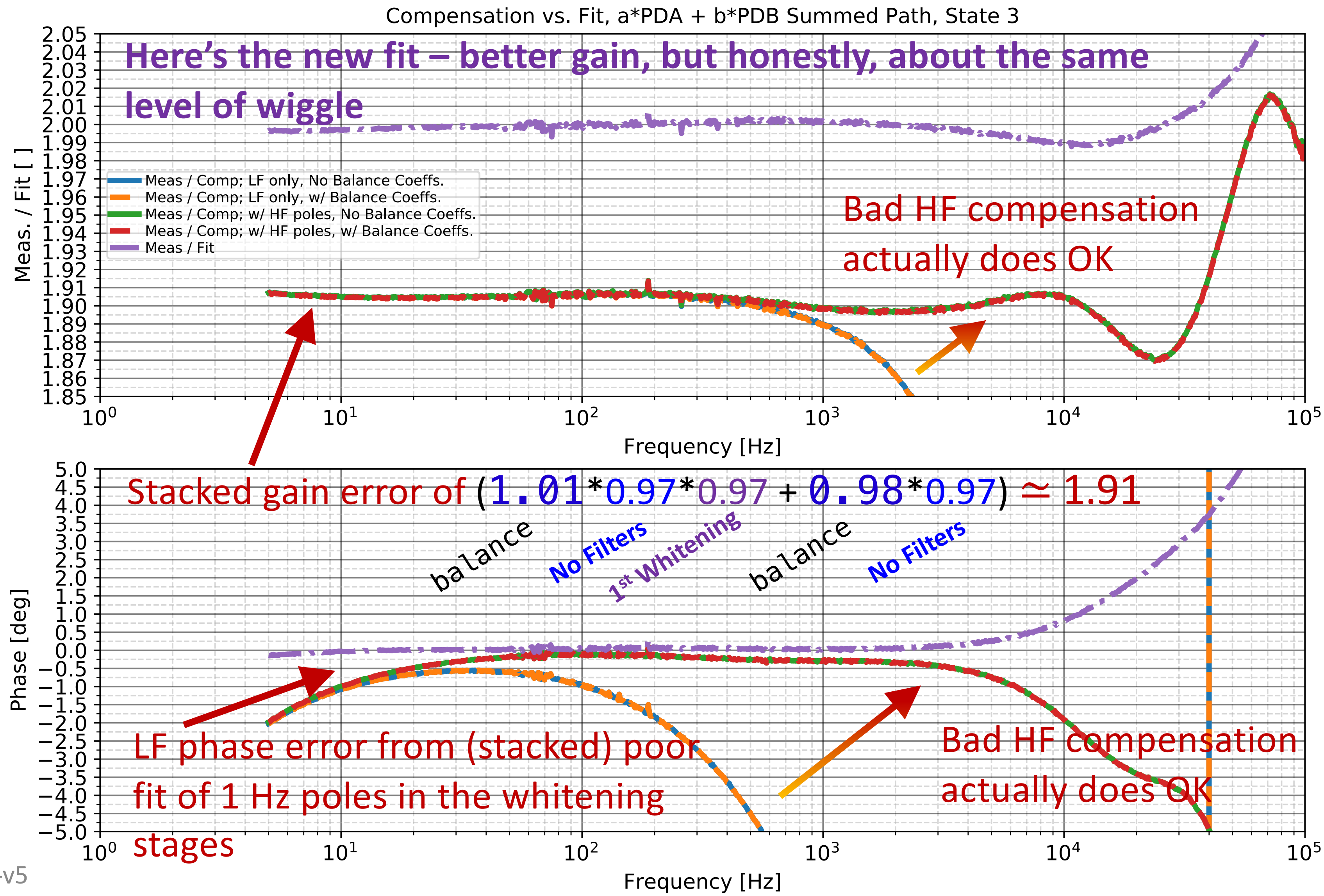
$f_p^F \approx 0.106$ MHz

$f_p^F \approx 0.106$ MHz

II.5 Fitting and Predicting Systematic Error

B.ii. For Each State create a stacked summed residual (both with existing compensation, and with fit)

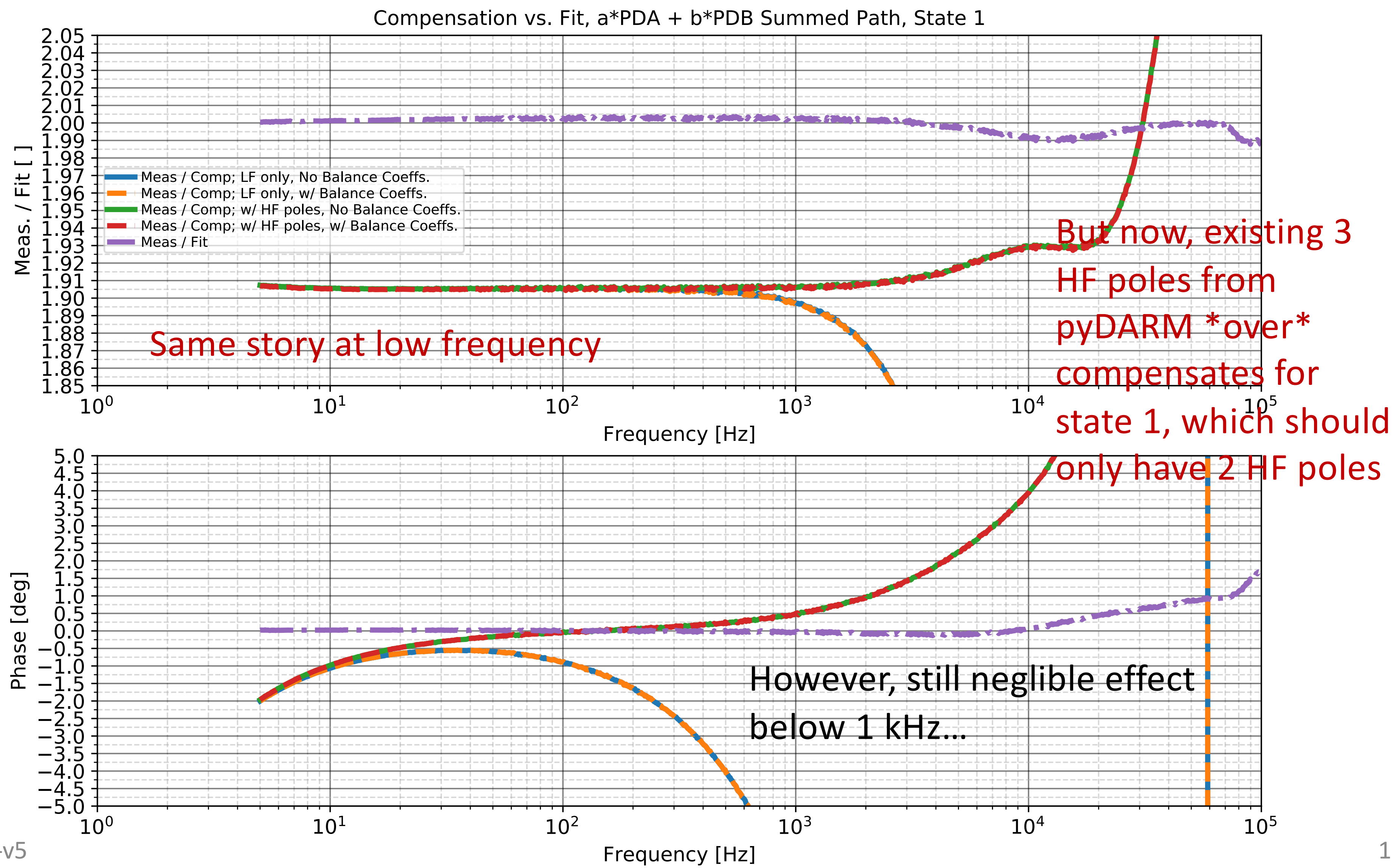
So here's where we were throughout most of O3: State 3, All three filter stages on, the correct number of HF poles, a *gain* flaw, but again is absorbed in G_{meas} (i.e. H_C)



II.5 Fitting and Predicting Systematic Error

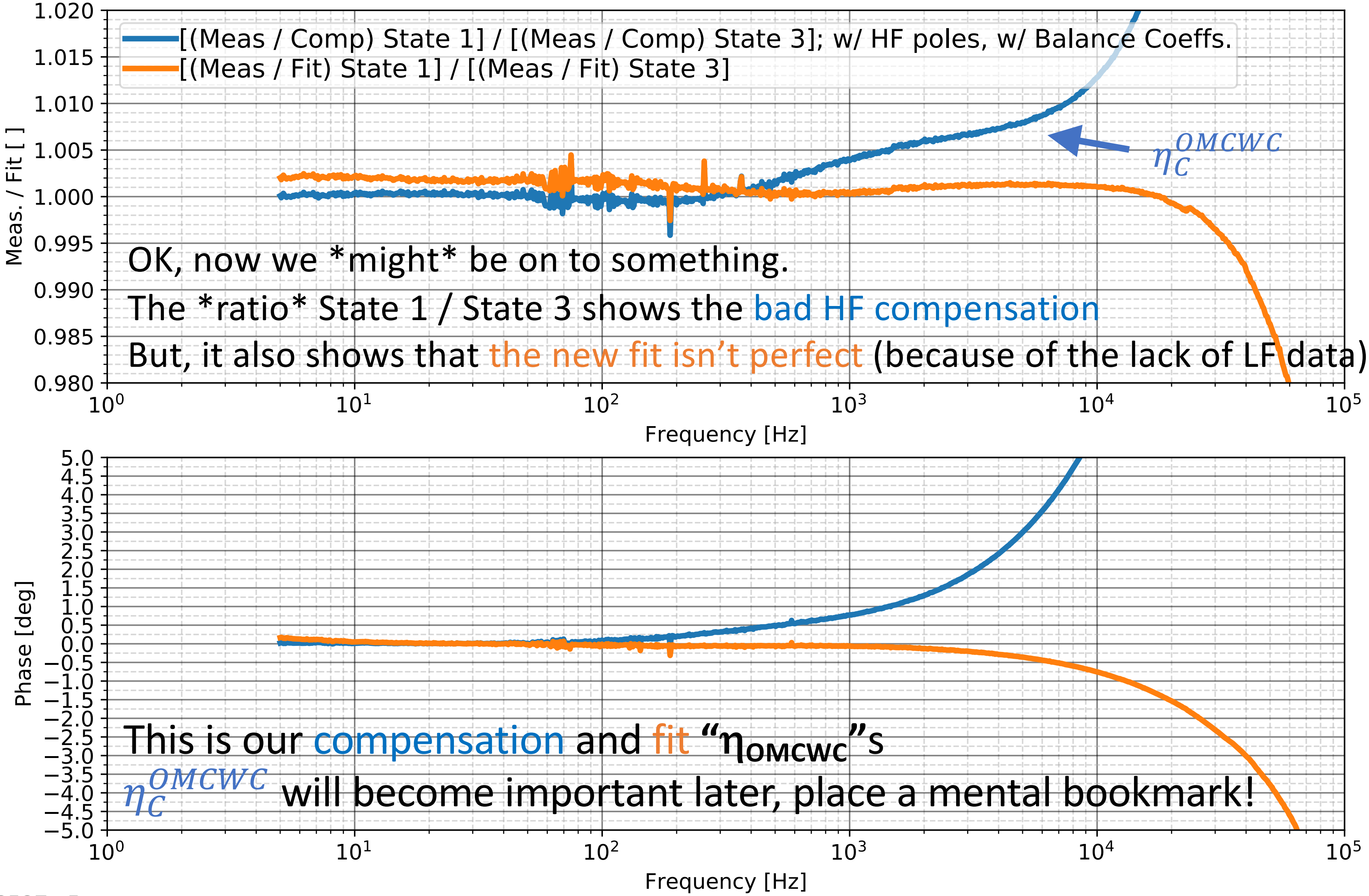
B.ii. For Each State create a stacked summed residual (both with existing compensation, and with fit)

So here's where we ended O3: State 1, only one low-pass on



II.5 Fitting and Predicting Systematic Error

B.ii. Take the ratio of State 1 (one Whitening Stage – i.e. configuration after Mar 10), and State 3 (both whitening stages and the low pass – i.e. what we had for all prior times in O3)



OK, now we *might* be on to something.

The *ratio* State 1 / State 3 shows the bad HF compensation

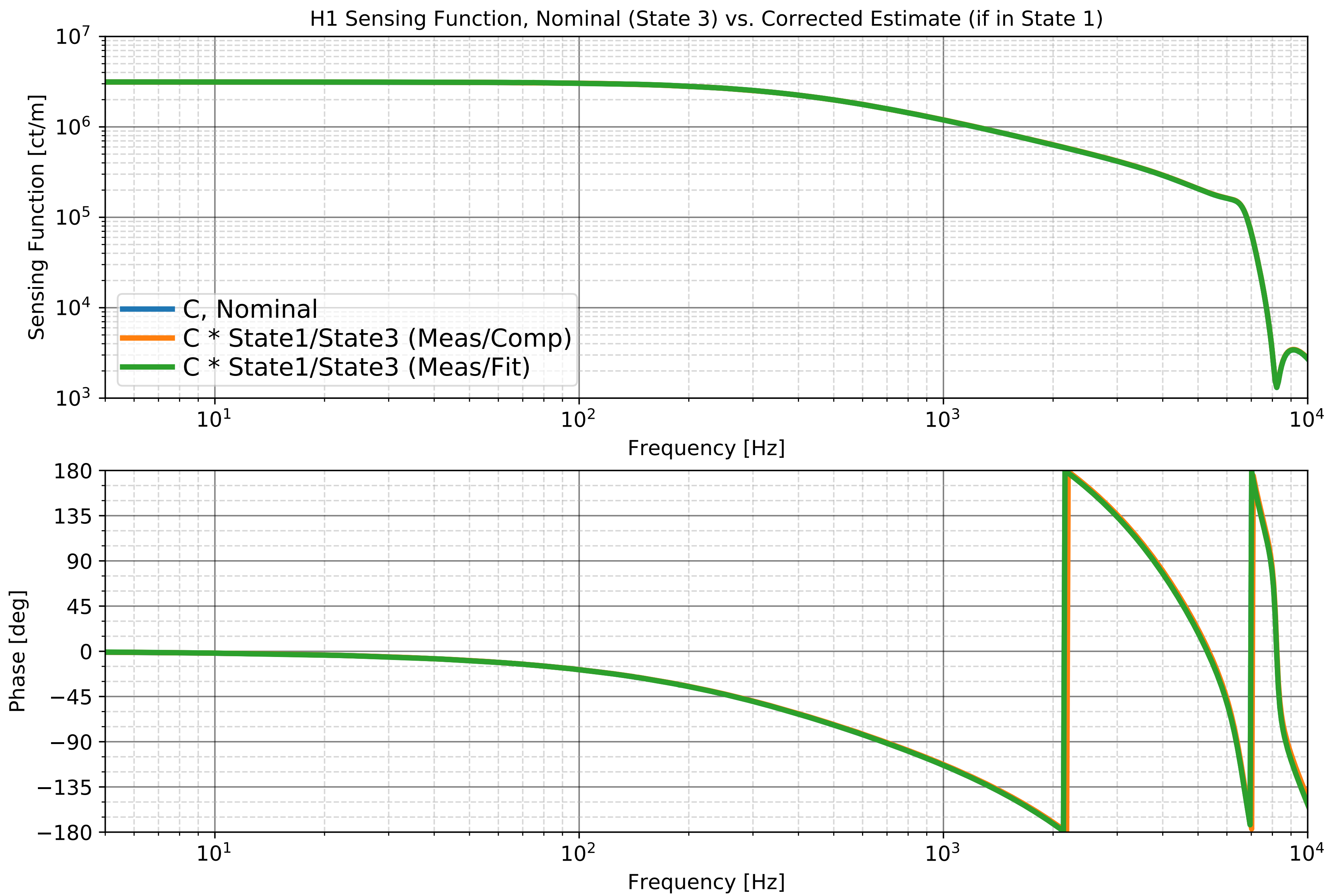
But, it also shows that the new fit isn't perfect (because of the lack of LF data)

This is our compensation and fit "η_{OMCWC}"s

η_C^{OMCWC} will become important later, place a mental bookmark!

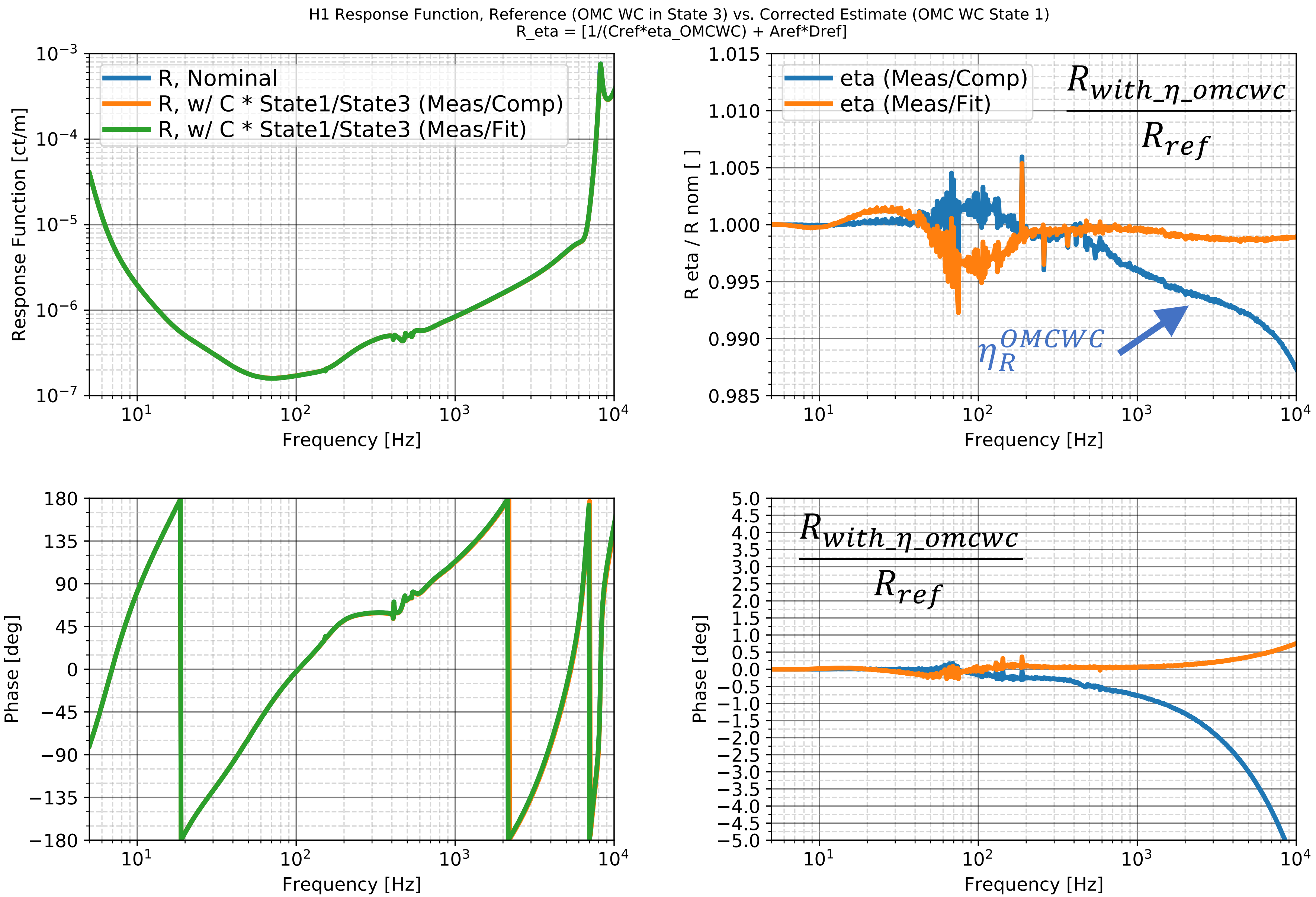
II.5 Fitting and Predicting Systematic Error

C.i. Use ratio of State 1 to State 3 as “η_{OMCWC}” for C



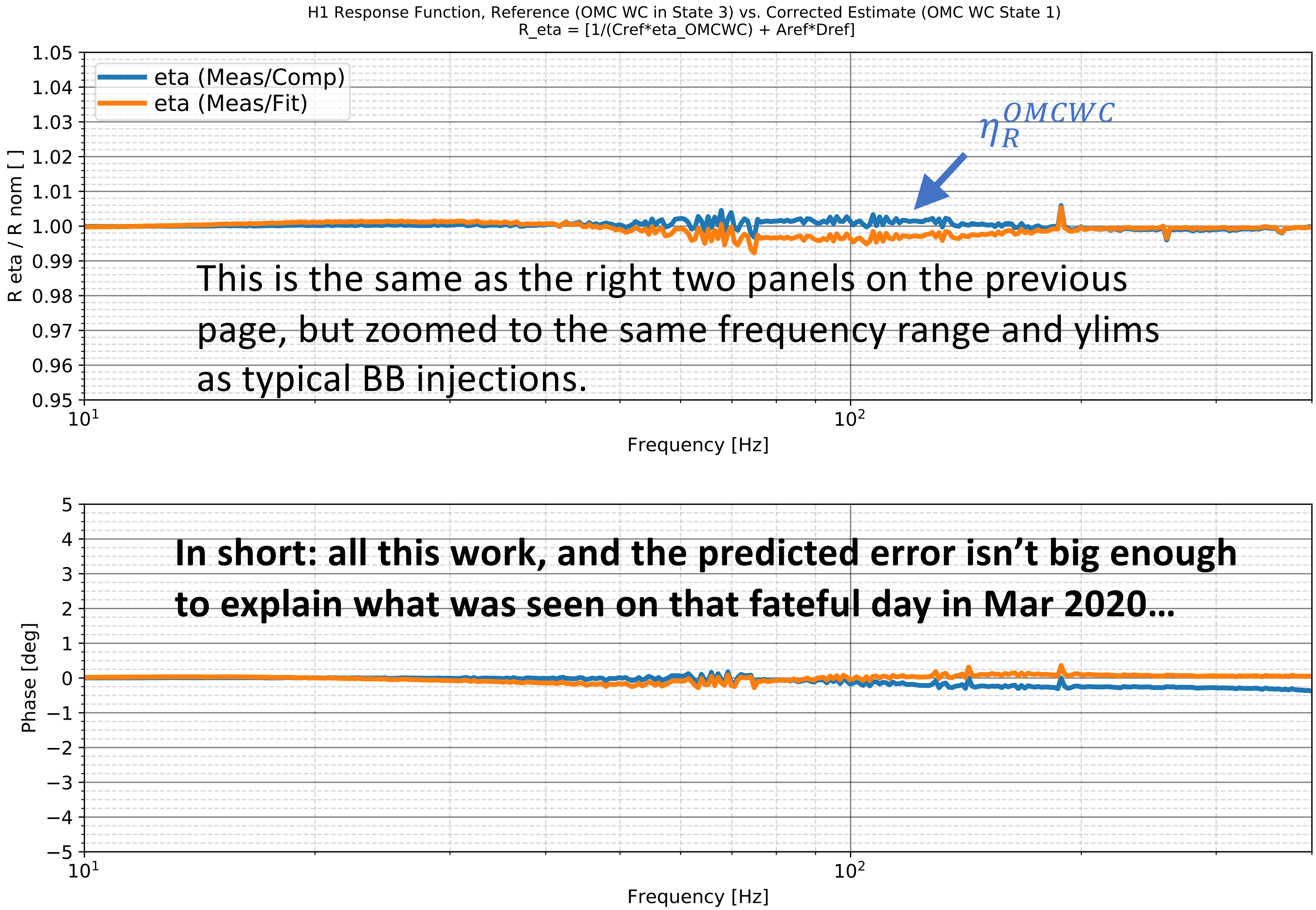
II.5 Fitting and Predicting Systematic Error

C.ii. Compute reference R from 20200103 model, then a modified R using C* η_{OMCWC}



II.5 Fitting and Predicting Systematic Error

C.ii. Compute reference R from 20200103 model, then a modified R using C* η_{OMCWC}



This is the same as the right two panels on the previous page, but zoomed to the same frequency range and ylims as typical BB injections.

In short: all this work, and the predicted error isn't big enough to explain what was seen on that fateful day in Mar 2020…

Outline

Two Parts, each quite long. *sigh*

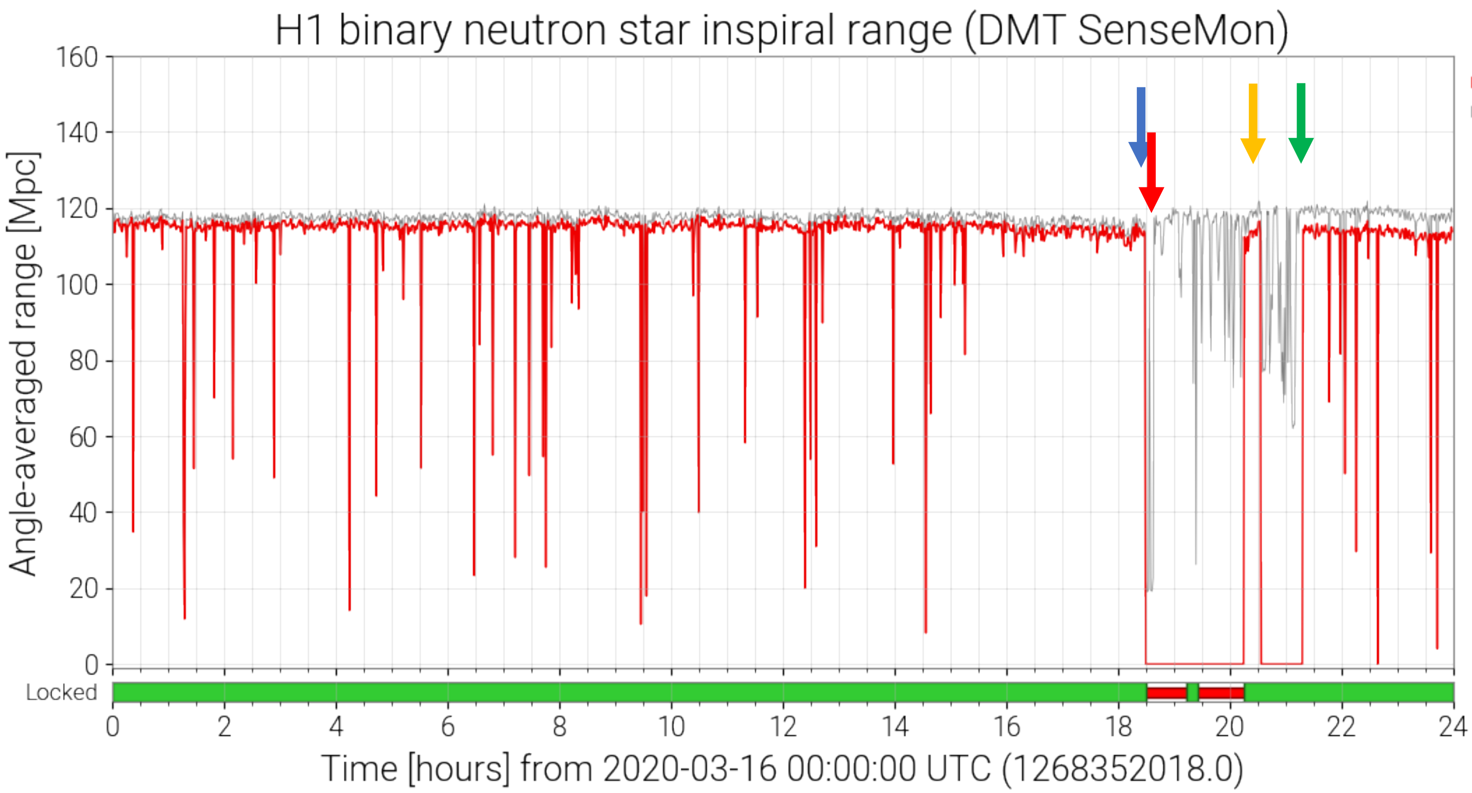
PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What's the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.6 Compare Against Measurement

Let's review what before vs. after broadband injection data is available.



DMT_ANALYSIS_READY end

BB Measurement

Brief NLN, but not DMT_ANALYSIS_READY

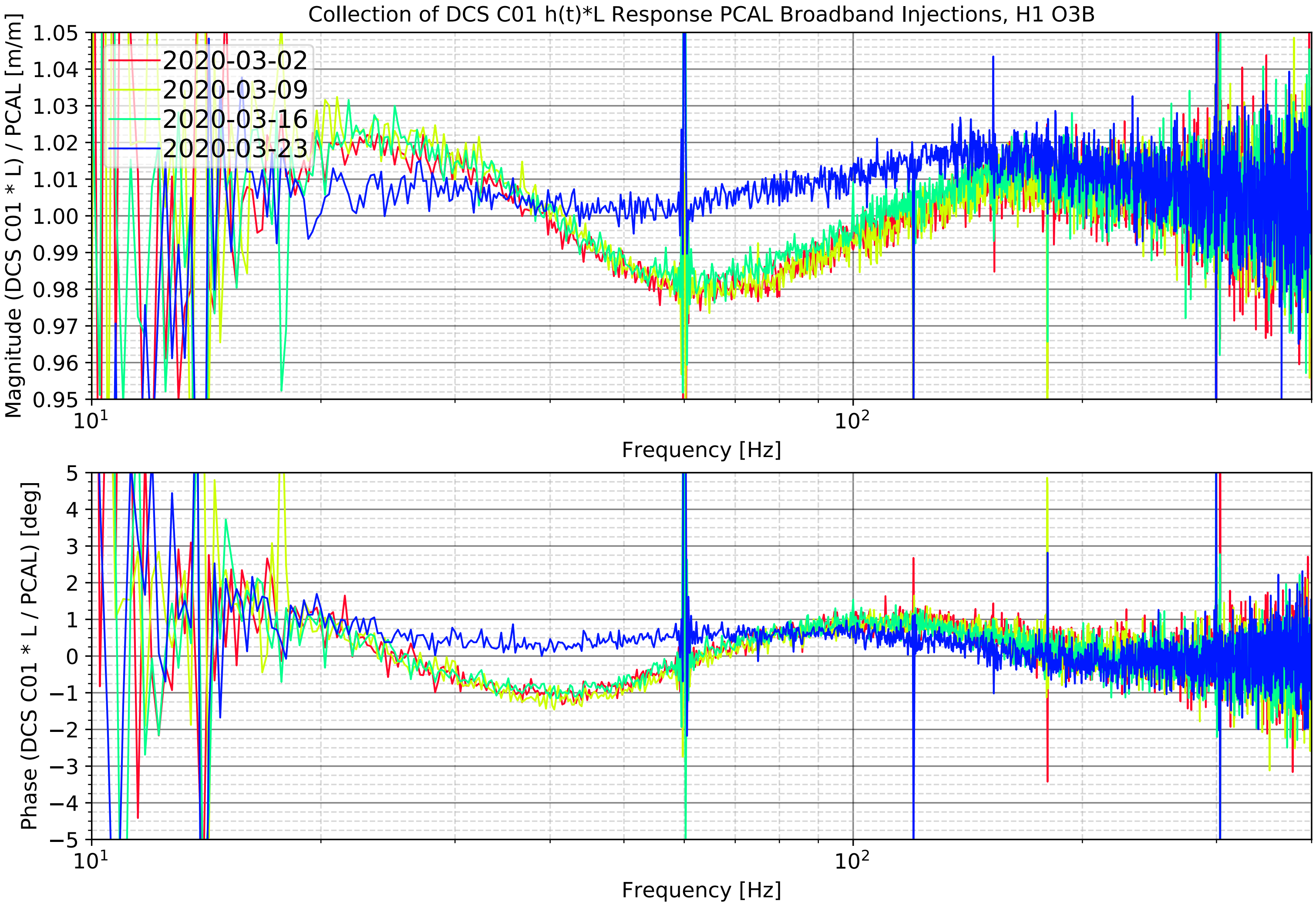
DMT_ANALYSIS_READY resumes

Detector was locked and happy for ~19 hours. Went out of OBS_READY at Mar 16 2020 18:29:59 UTC, switched whitening config, and measured broadband 30 seconds afterword.

- Pre
  - 2020-03-02_H1_PCALY2DARMTF_BB_3min.xml: 2020-03-02 19:00:32 UTC
  - 2020-03-09_H1_PCALY2DARMTF_BB_3min.xml: 2020-03-09 18:00:33 UTC
- Post
  - 2020-03-16_H1_PCALY2DARMTF_BB_3min.xml: 2020-03-16 18:30:31 UTC ←
  - 2020-03-23_H1_PCALY2DARMTF_BB_3min.xml: 2020-03-23 18:01:20 UTC

II.6 Compare Against Measurement

But somethings already fishy here…

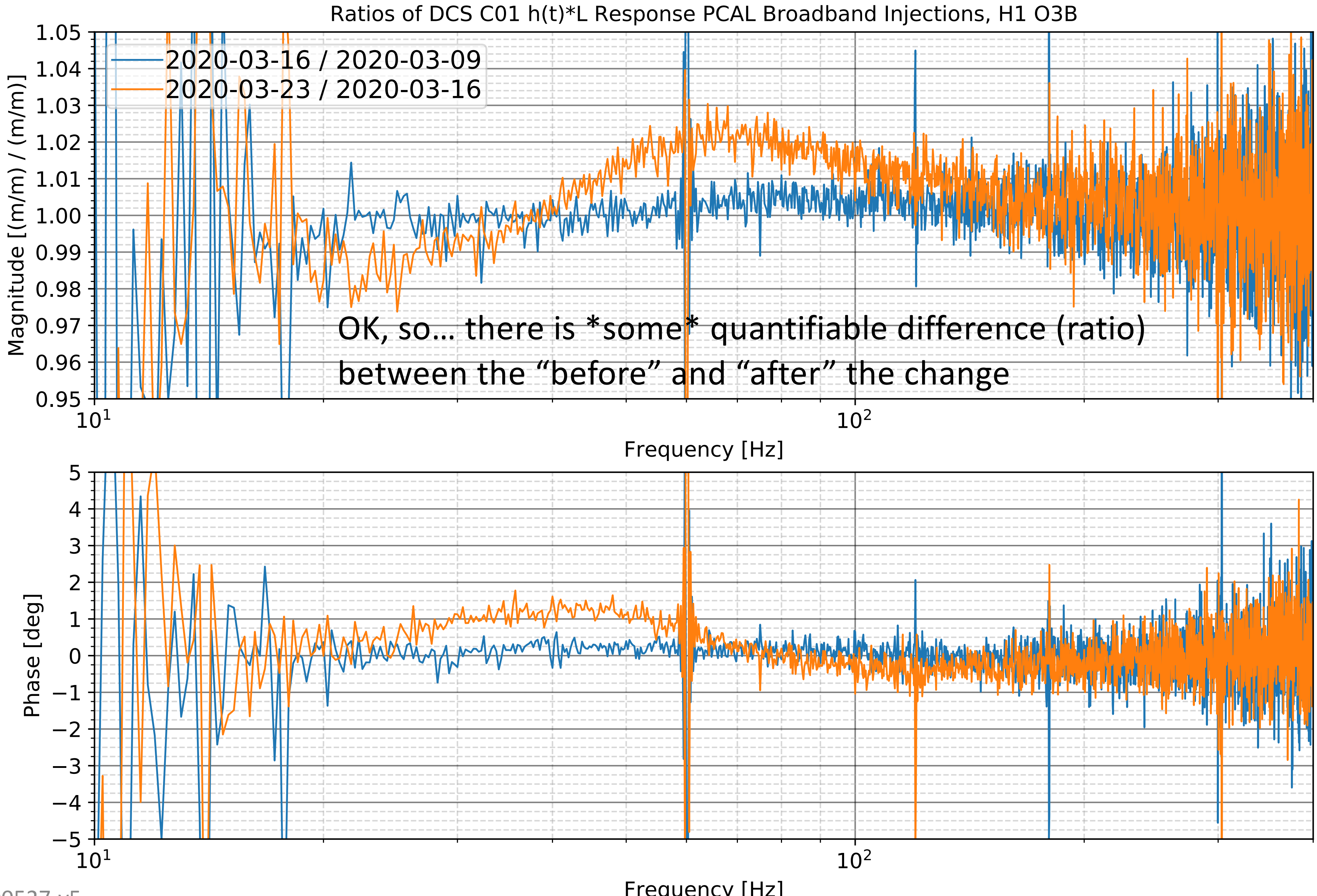


If the whitening configuration changed *before* the 2020-03-16 measurement, shouldn't there be a difference?

Ok – maybe if you squint… you can see a change between 2020-03-09 and 2020-03-16… but what the heck is going on with 2020-03-23?

II.6 Compare Against Measurement

Maybe we can make sushi out of the fish?



OK, so… there is *some* quantifiable difference (ratio) between the “before” and “after” the change

II.6 Compare Against Measurement

Now, before we compare slide 122 with 118 this is *always* super duper confusing, let's make sure we plot the right thing, and compare apples to apples instead of apples to (1/apples).

On page 115, we've defined the systematic error in C as

$$\eta_{omcwc} \equiv \frac{C_{state\ 1}}{C_{state\ 3}} = \frac{C_{after}}{C_{before}}$$

Then, on page 117 and 118 we show

$$R_{reference} = \frac{1}{C_{before}} + AD$$

$$R_{with\_\eta\_omcwc} = \frac{1}{C_{after}} + AD = \frac{1}{C_{before} * \eta_{omcwc}} + AD$$

$$\eta_R = \frac{R_{with\_\eta\_omcwc}}{R_{reference}}$$

For the measurements on page 121 and 121, remember from T1900169

$$\frac{\Delta L}{PCAL} = \frac{R_{DCS}}{R_{PCAL}}$$

DCS response function isn't changing – the PCAL excitation is measuring the change,

$$\left.\frac{\Delta L}{PCAL}\right|_{2020-03-09} = \frac{R_{DCS}}{R_{reference}}$$

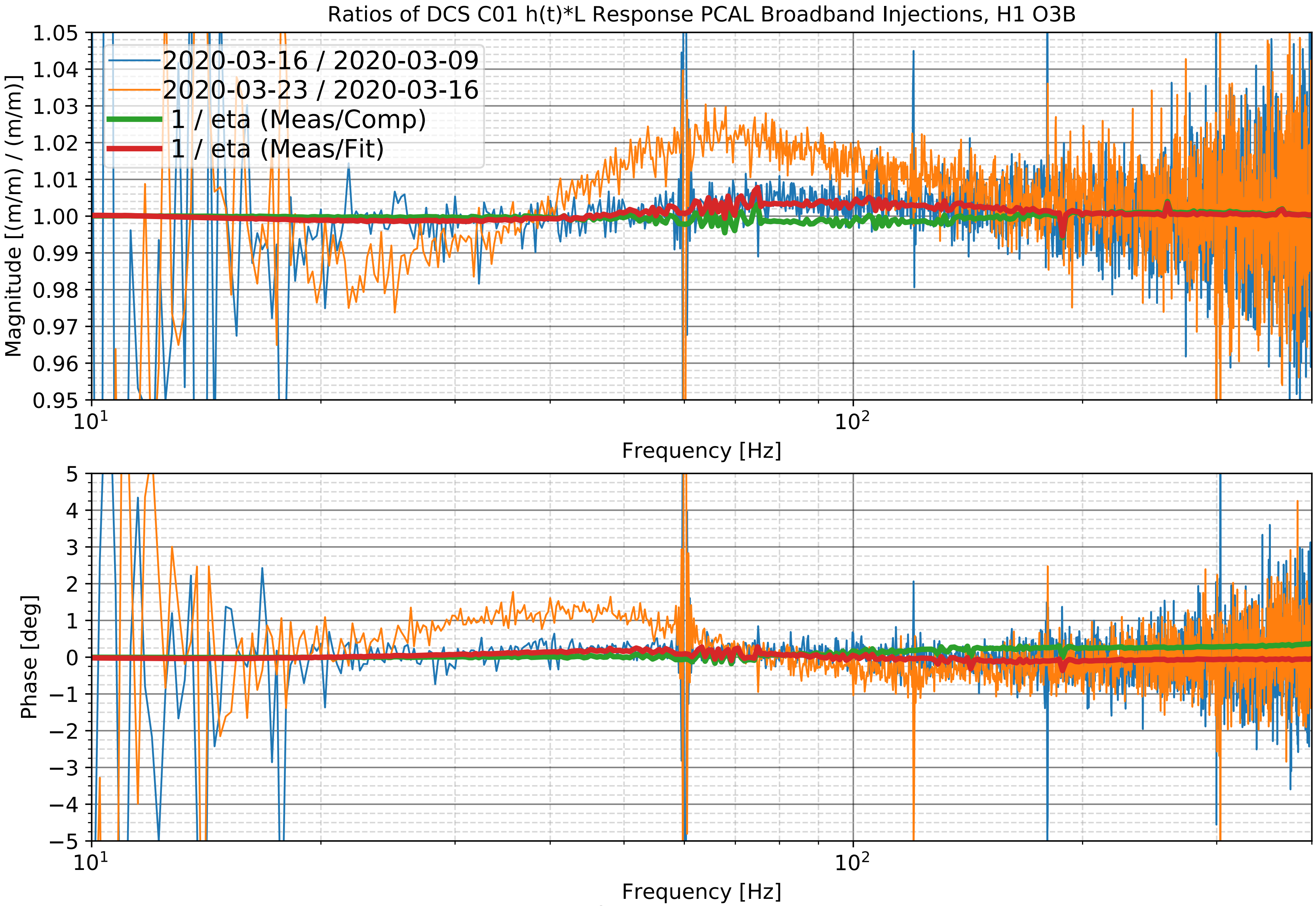
$$\left.\frac{\Delta L}{PCAL}\right|_{2020-03-16} = \frac{R_{DCS}}{R_{with\_\eta\_omcwc}}$$

Which means

$$\frac{\boldsymbol{\Delta L/PCAL|_{200316}}}{\boldsymbol{\Delta L/PCAL|_{200309}}} = \frac{\boldsymbol{R_{reference}}}{\boldsymbol{R_{with\_\eta\_omcwc}}} = \frac{\mathbf{1}}{\boldsymbol{\eta_R}}$$

II.6 Compare Against Measurement

Sushi go!



HaHA! Vindication! **The new fit does a good job at predicting the *miniscule* change in the response function. Great!**

So… what the heck is going on with the 2020-03-23 measurement??

Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What’s the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.7 Collateral Damage on/from TDCFs

- G2001293 started out as an unrelated investigation in to why the cavity pole changed by 7 Hz after the OMC whitening chassis change.

- However, the result of that investigation concludes that **the OMC whitening chassis error** (predicted in this talk) **at 410.3 Hz actually causes the sensing function TDCFs to falsely report a change in the optical gain and cavity pole.**

- These poorly informed TDCFs were then applied to h(t), and thus create a self-inflicted systematic error. In G2001293, we show that **accounting for this error almost entirely accounts for the change between the 2020-03-16 measurement and 2020-03-23 measurement.** Mystery solved!

- But, one last thing (picking up after the conclusion of G2001293): to completely account for impact, we will apply both the (negligible) η_C^{OMCWC} and the (more impactful) η_C^{TDCFs} to the Chunk 2, Period c systematic error and uncertainty budget.
  - (where the application of η_C^{OMCWC} will mostly just account for the small amount of error above 1 kHz)

II.7 Collateral Damage on/from TDCFs

The **maaaaaath** covered on page 14 of G2001293 reminds us that, because of the "1+" part of the response function, the modification to C is not so straightforwardly propagated to a modification in R. So let's pick up where we left off, but include η_C^{OMCWC} as well as η_C^{TDCFs}.

$$\eta_C^{total} \equiv \eta_C^{OMCWC} \eta_C^{inc \gg cor} \eta_C^{ref \gg inc}$$

$$R_{ref} = \frac{1+ADC}{C}$$

$$\eta_R^{ref \gg inc} \equiv \frac{R_{incorrect}}{R_{ref}}$$

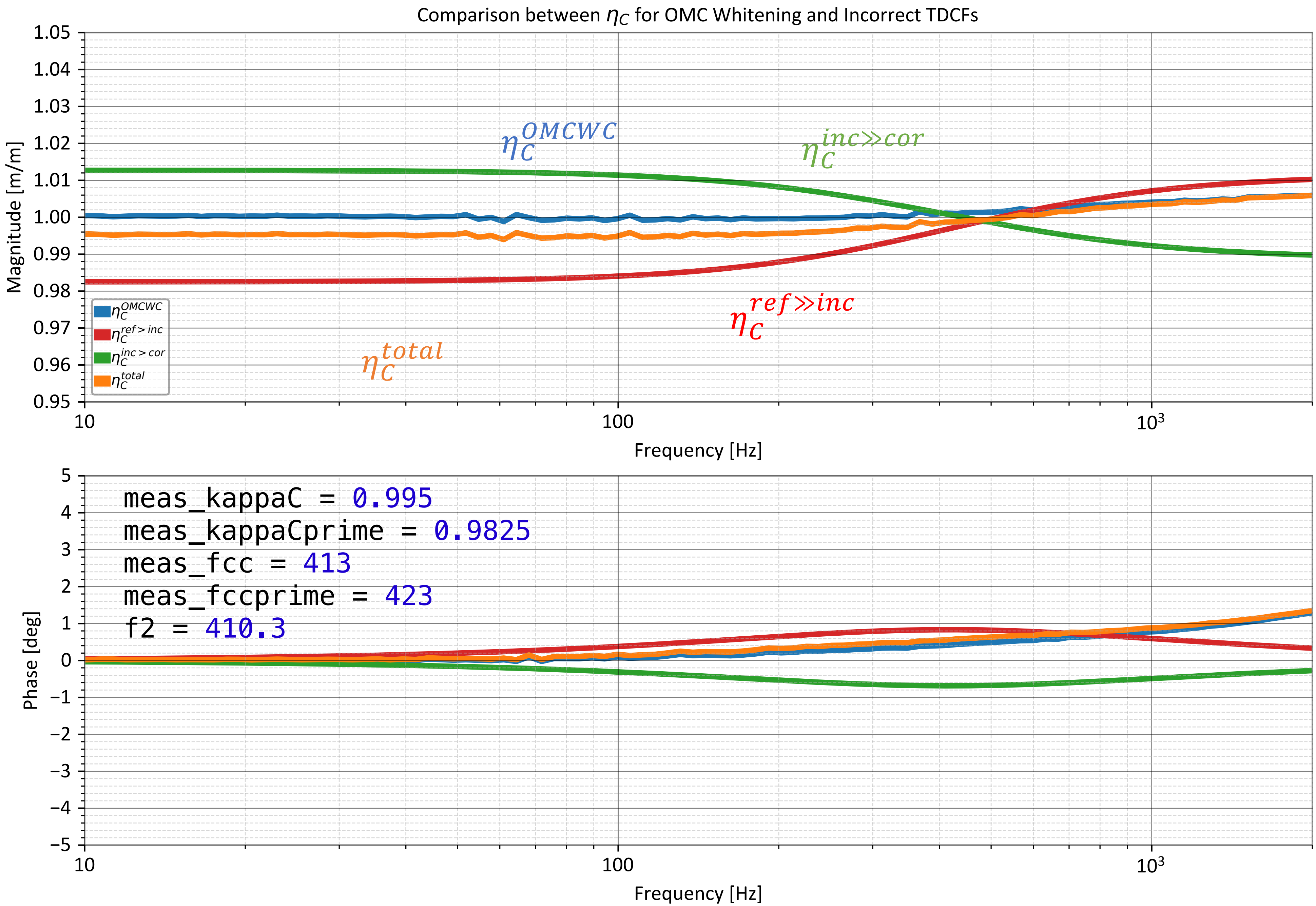
$$R_{incorrect} = \frac{1+\kappa_C' \frac{(1+if/f_{cc}^{ref})}{(1+if/f_{cc}')} ADC}{\kappa_C' \frac{(1+if/f_{cc}^{ref})}{(1+if/f_{cc}')} C} = \frac{1+\eta_C^{ref \gg inc} ADC}{\eta_C^{ref \gg inc} C}$$

$$\eta_R^{ref \gg cor} \equiv \frac{R_{correct}}{R_{ref}}$$

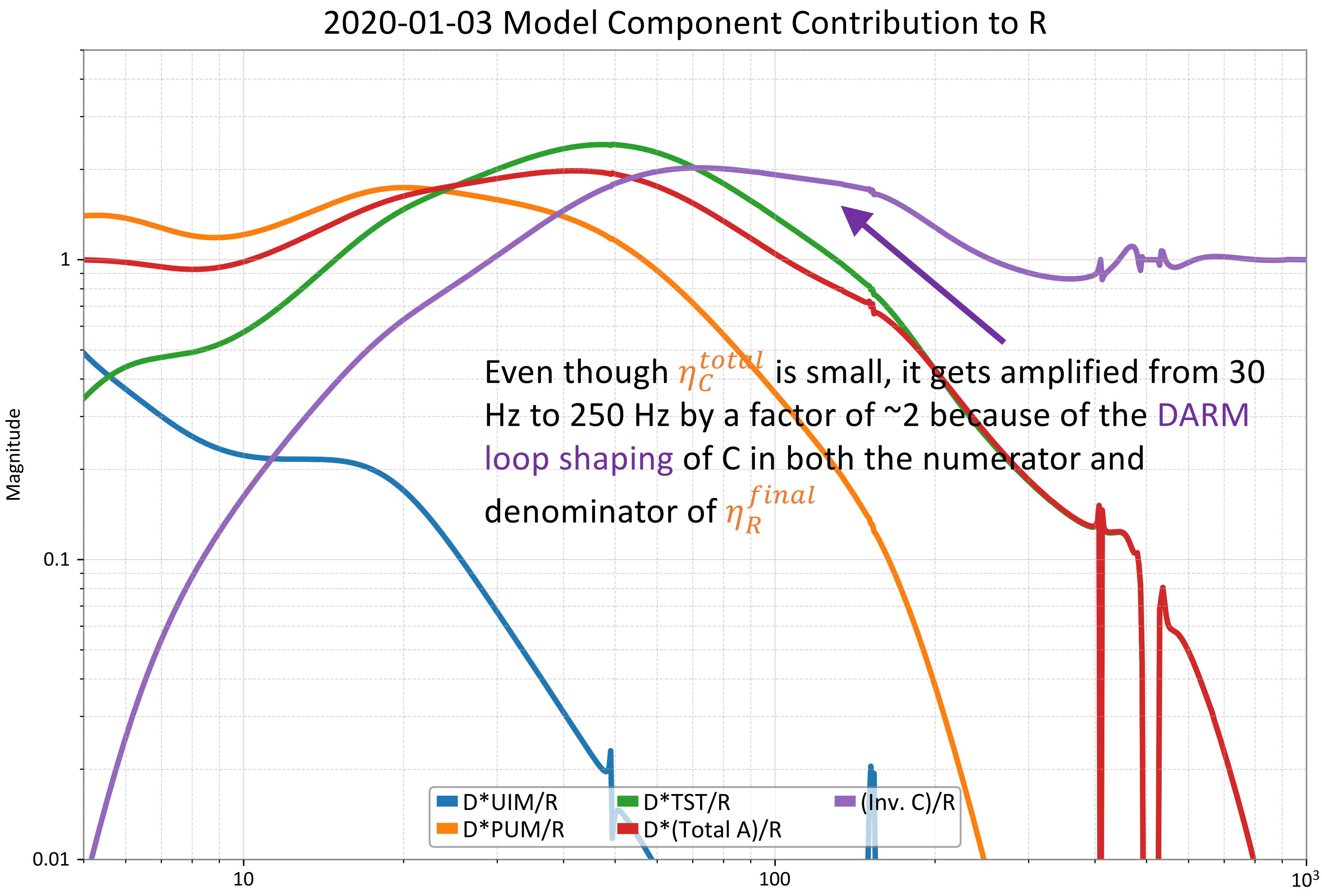
$$R_{correct} = \frac{1+\eta_C^{OMCWC} \frac{\kappa_C}{\kappa_C'} \frac{(1+if/f_{cc}')}{(1+if/f_{cc})} \kappa_C' \frac{(1+if/f_{cc}^{ref})}{(1+if/f_{cc}')} ADC}{\eta_C^{OMCWC} \frac{\kappa_C}{\kappa_C'} \frac{(1+if/f_{cc}')}{(1+if/f_{cc})} \kappa_C' \frac{(1+if/f_{cc}^{ref})}{(1+if/f_{cc}')} C} = \frac{1+\eta_C^{OMCWC} \eta_C^{inc \gg cor} \eta_C^{ref \gg inc} ADC}{\eta_C^{OMCWC} \eta_C^{inc \gg cor} \eta_C^{ref \gg inc} C}$$

$$\eta_R^{final} \equiv \frac{R_{correct}}{R_{incorrect}} = \frac{1+\eta_C^{OMCWC} \eta_C^{inc \gg cor} \eta_C^{ref \gg inc} ADC}{1+\eta_C^{ref \gg inc} ADC} \frac{1}{\eta_C^{OMCWC} \eta_C^{inc \gg cor}}$$

II.7 Collateral Damage on/from TDCFs



II.7 Collateral Damage on/from TDCFs



Even though η_C^{total} is small, it gets amplified from 30 Hz to 250 Hz by a factor of ~2 because of the DARM loop shaping of C in both the numerator and denominator of η_R^{final}

Outline

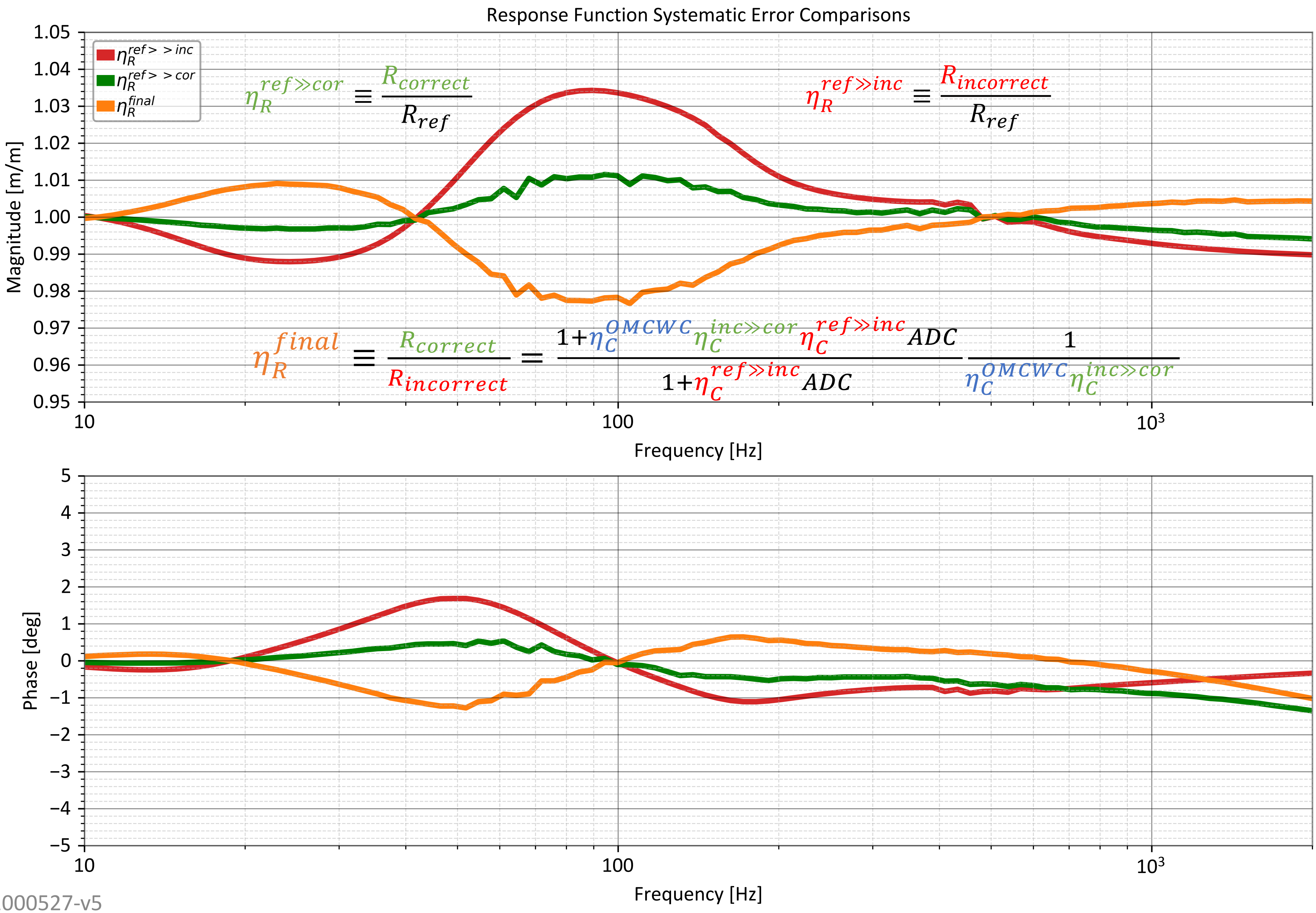
Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

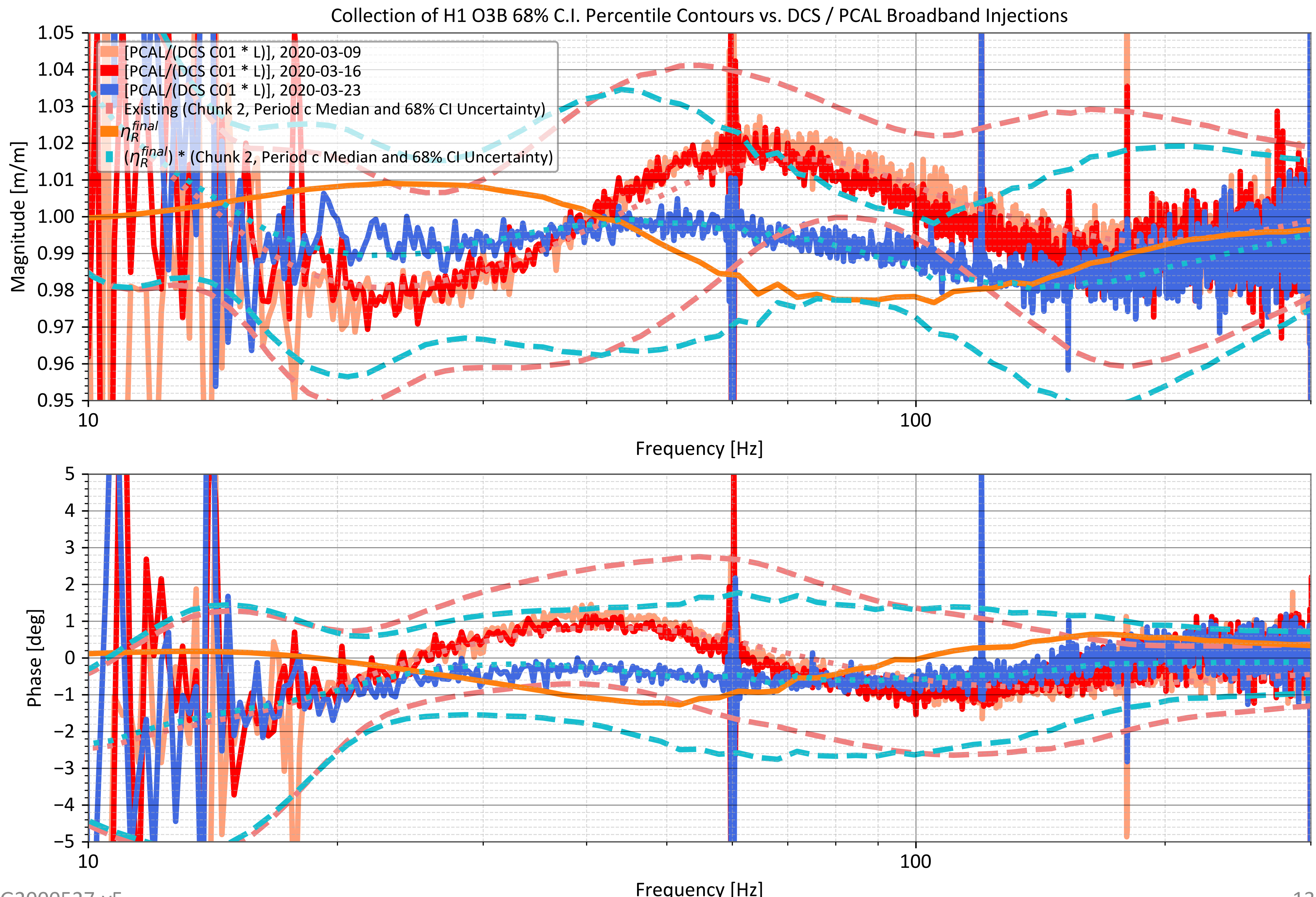
PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What’s the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

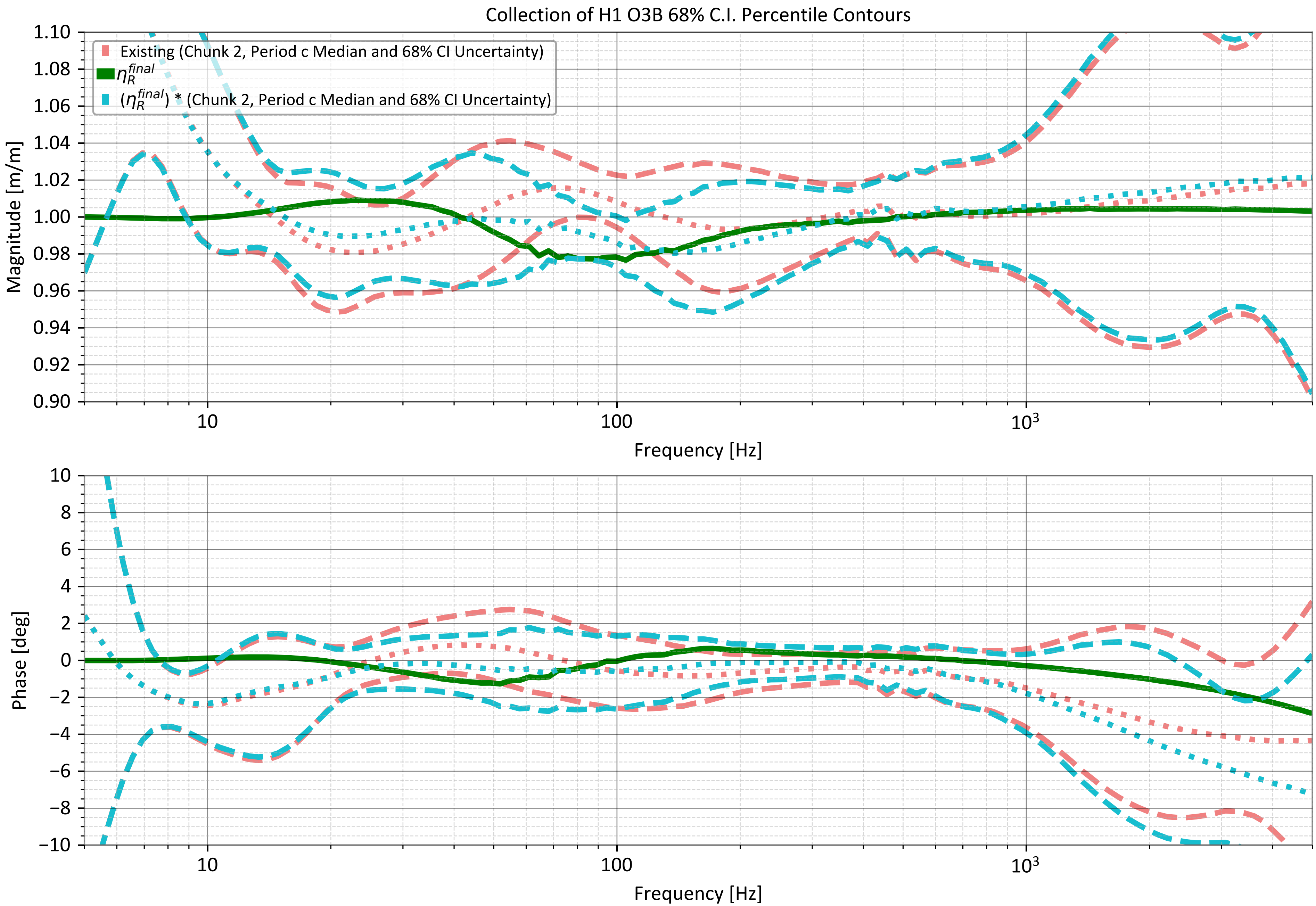
II.8 Final Answer: The construction of η_R^{final}



II.8 Final Answer: succession of BB vs. Modified Percentile for Period c



II.8 Final Answer: Predicted Impact across the Entire Band



II.8 Final Answer

- η_C^{OMCWC} and the bulk of the fitting/modeling plots was produced by

/ligo/svncommon/CalSVN/aligocalibration/trunk/Common/Electronics/H1/Scripts/

fit_OMCDCPDWhiteningChassis_20190304_forG2000527.py

and exported to

/ligo/svncommon/CalSVN/aligocalibration/trunk/Runs/O3/H1/Results/Uncertainty/

2020-08-25_measDate2019-03-04_H1OMC_WhiteningChassisTFs_eta_C_omcwc.txt

- $\eta_C^{inc \gg cor} \eta_C^{ref \gg inc}$ and the work presented in G2001293 was produced by

/ligo/svncommon/CalSVN/aligocalibration/trunk/Common/Documents/G2001293_O3BChunk2Periodc_OMCWhitening_and_TDCF_SysError/

showimpact_omcwhiteningchassiserror_on_sensingTDCFs_and_R.py

- η_C^{OMCWC}, $\eta_C^{inc \gg cor}$, and $\eta_C^{ref \gg inc}$ were all combined as described on page 127 by

/ligo/svncommon/CalSVN/aligocalibration/trunk/Runs/O3/H1/Scripts/Uncertainty/

combine_OMCWC_and_BadSensingTDCF_syserror_H1O3BChunk2Periodc_20200316-20200327.py

- The resulting η_R^{final} produce by that script was exported as

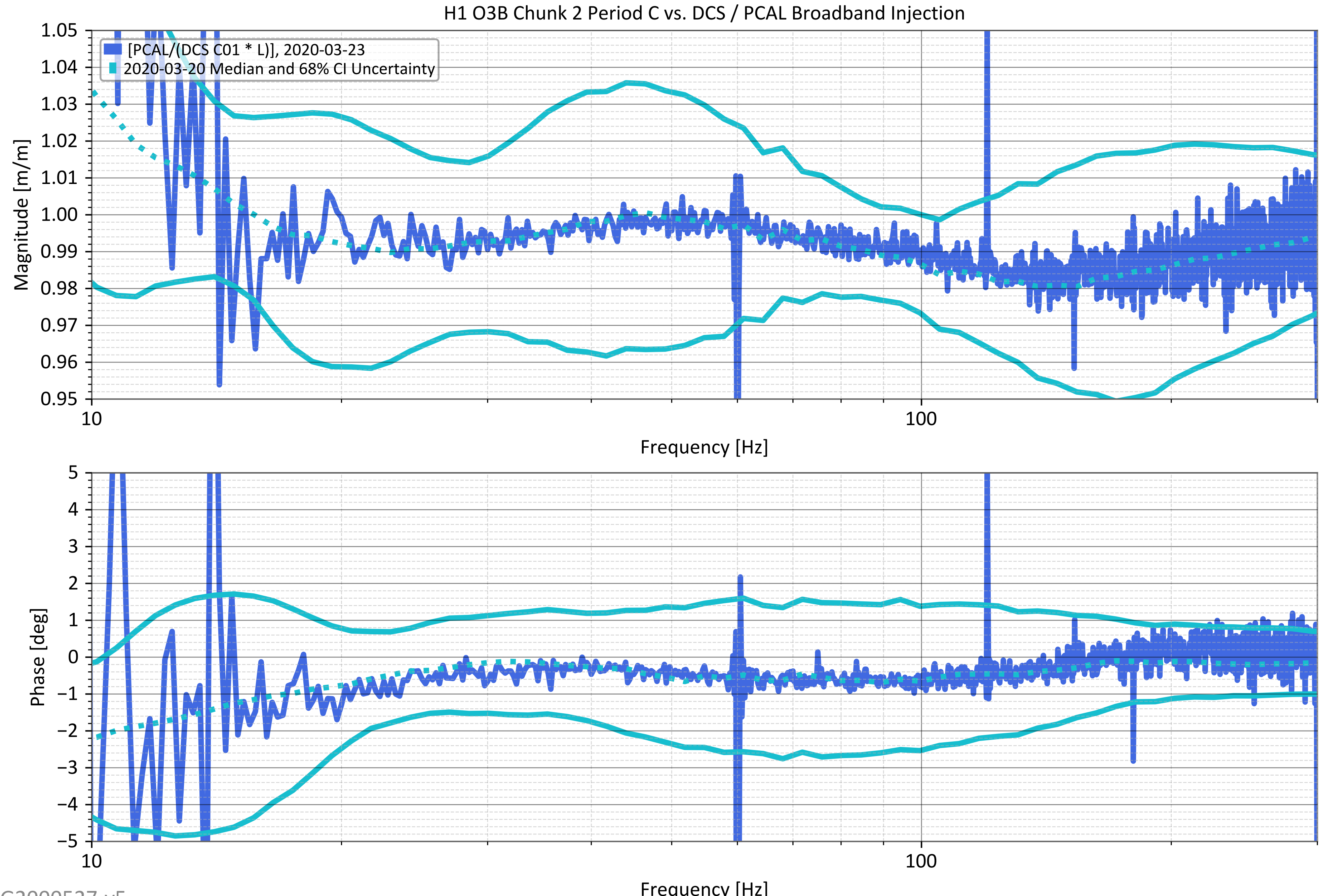
/ligo/svncommon/CalSVN/aligocalibration/trunk/Runs/O3/H1/Results/Uncertainty/

2020-08-25_H1O3BChunk2Periodc_ResponseFunctionSysError_OMCWC_and_BadTDCFs_eta_R_omcwc_badtdcfs.hdf5

- RRNom was modified in rev 11150

II.8 Final Answer

LHO aLOG 56582 details this, the 2020-03-23 broadband injection against the official, RRNom produced budget



Outline

Two Parts, each quite long. *sigh*

PART I: The ETMX UIM Driver, from Nov 27 to Dec 03 2019

PART II: The OMC Whitening Chassis, from Mar 16 to 27 2020

1. What's the issue?
2. Game Plan / Review of the Original Data
3. Review of the Circuit
4. Review of the Existing Compensation Scheme
5. Fitting and Predicting the Response Function Systematic Error
6. Comparison against measured Response Function Error
7. Collateral Damage on TDCFs
8. Final Answer
9. Conclusions

II.8 Conclusions

- Lessons learned:
  - We really need to take the time to get to know all of the circuits that matter, and understand how to best measure them
  - Yes, we can predict the frequency-dependent change caused by electronics errors in the sensing function.
  - We should take a new measurement of the OMC Whitening Chassis,
    - be sure to get data down to at least 0.25 Hz (and if patience allows 0.1 Hz), and update the compensation filters.
    - be sure to take both cascading response and single filter response measurements.
- We've found some insidious collateral damage of the OMC whitening chassis configuration change itself
  - the sensing function TDCFs were impacted by this, incorrectly reporting plant change, and cause *much worse* systematic error
  - Discovered and reviewed in G2001293
- Final answer (where we include OMC error itself *and* the correction for bad TDCFs) agrees nicely with 2020-03-23 measurement.

FIN