Post-facto Noise Subtraction Pipeline, Bilinear Coupling

Keita KAWABE, LHO

Disclaimer

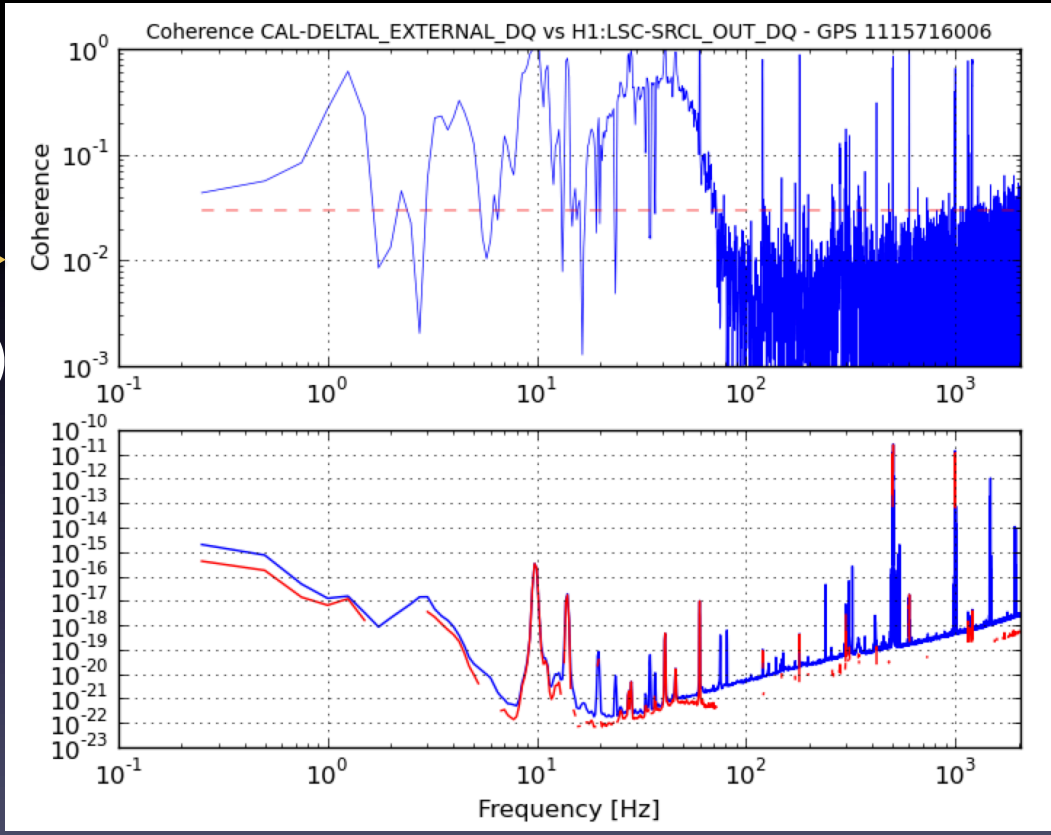
- This is about noise subtraction (FF)
- but nothing modern , nothing new, no controls topic per se, not about serious problems we face now
  - (but they’re going to be more interesting once we start producing science data)

Topics

1. Post-facto noise subtraction pipeline originally developed for S6.
2. Possibility of subtracting bi-linear (and potentially tri-and higher order) coupling.

Noise that could couple linearly

- Auxiliary control
  - (MICH , PRCL, SRCL)
- Ground motion
- PSL/IMC (ISS, FSS)
- There could be many more.



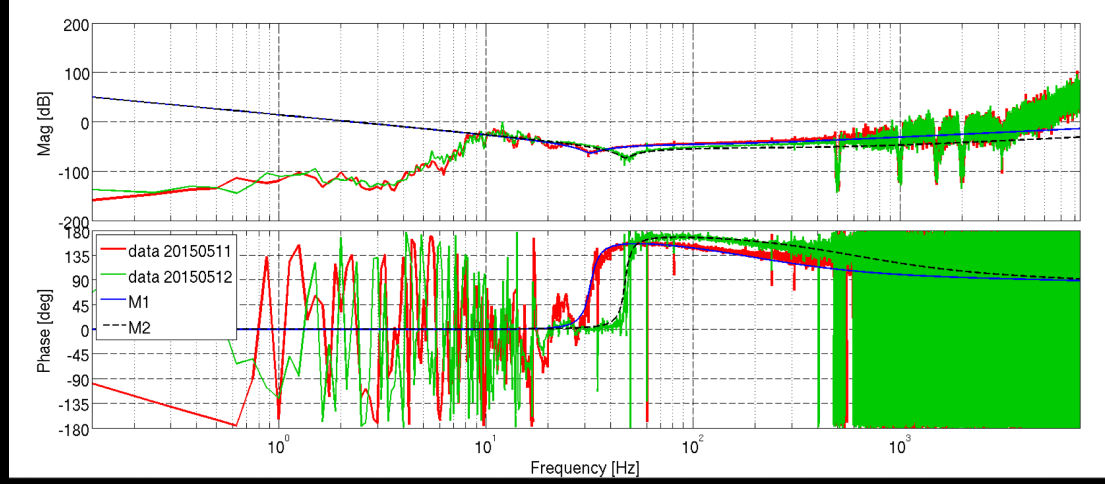
Attacking a known linear noise coupling

1. Reduce the noise if you can,
2. And/or reduce the coupling if you can,
3. And then feed forward in real time to subtract the residual and life should be good.
4. (In LIGO, FF paths can be and are built into the control system, e.g. MICH, SRCL, PRCL, tilt meter)

But what if…

- your FF is not as awesome as it should be?
  - Your TF might not be good at sub-% level.

But what if…



- the coupling changes over time for some reason?
  - Alignment drift,
  - Thermal drift,
  - Charge accumulation/loss affecting ESD strength
  - Etc.

But what if…

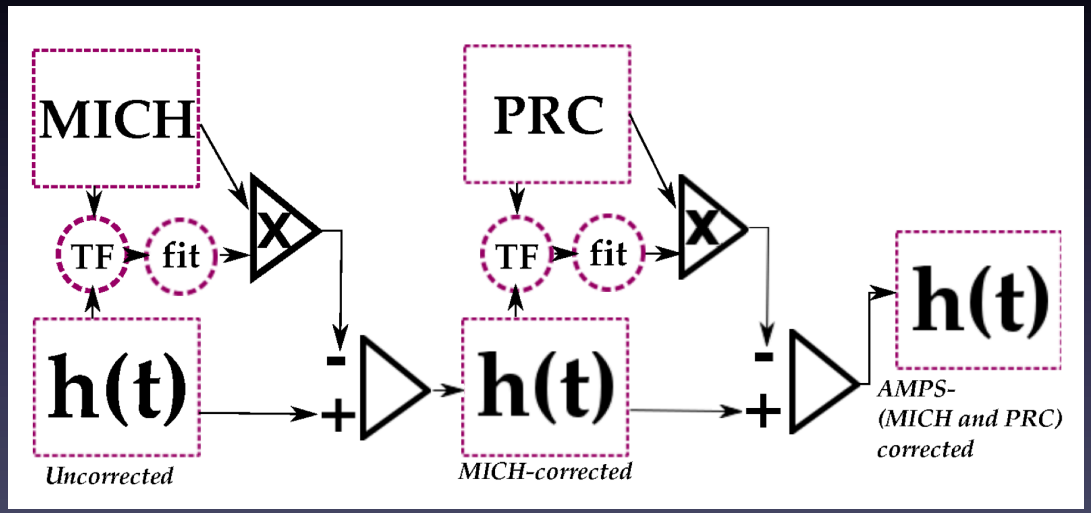
- the noise or the coupling becomes momentarily larger and pollutes your science data?
- you find a new coupling after the science data is taken?

It's OK

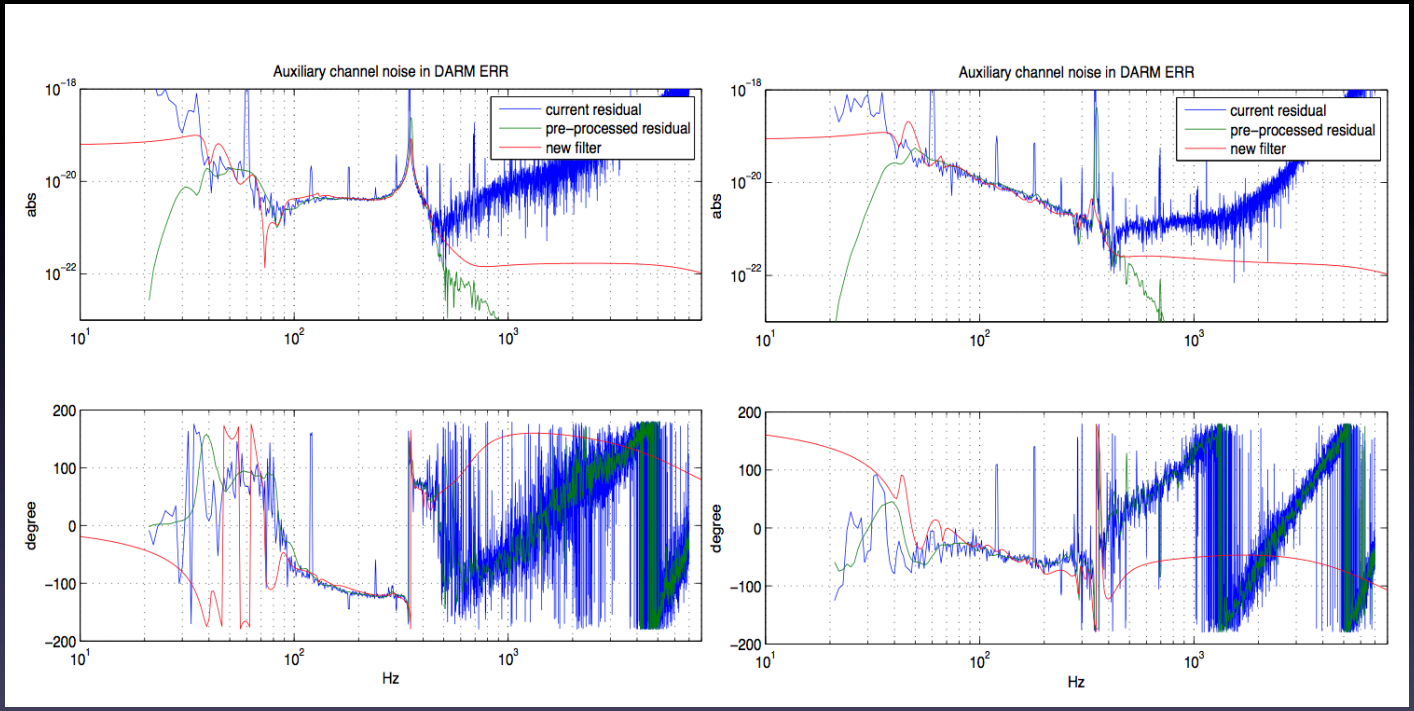
- You can still retroactively subtract these guys as far as the data is on the disk.

Example: Grant Meadors' AMPS pipeline

- eLIGO S6 off-line subtraction of MICH and PRCL
- Adaptive t-domain filter automatically updated
- This is not only about S6 nor MICH/PRCL! Could be adapted to other noise/IFO



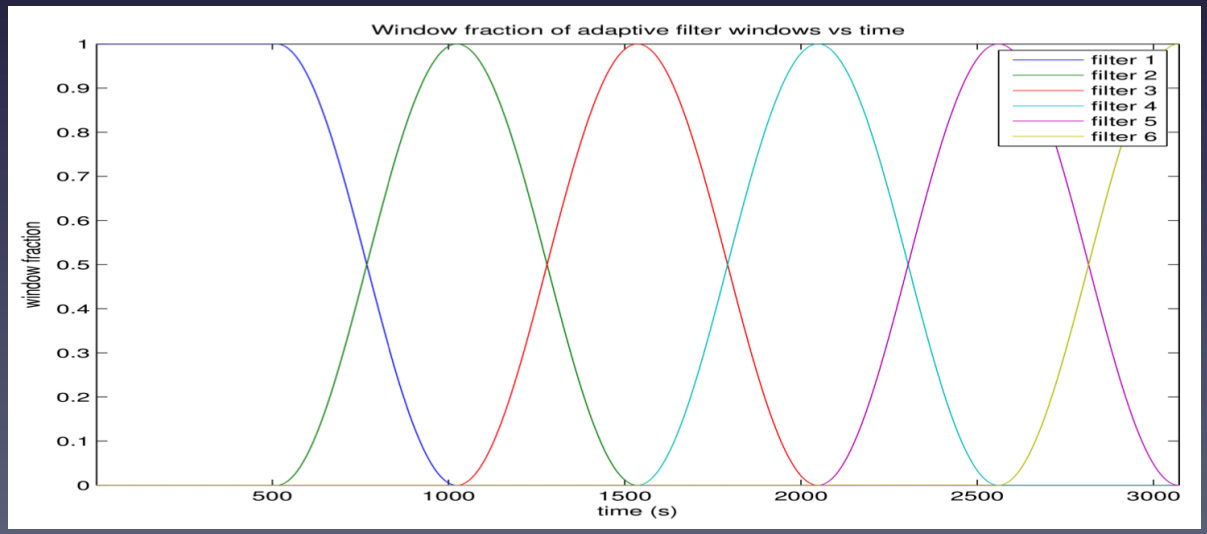
High quality noise TF



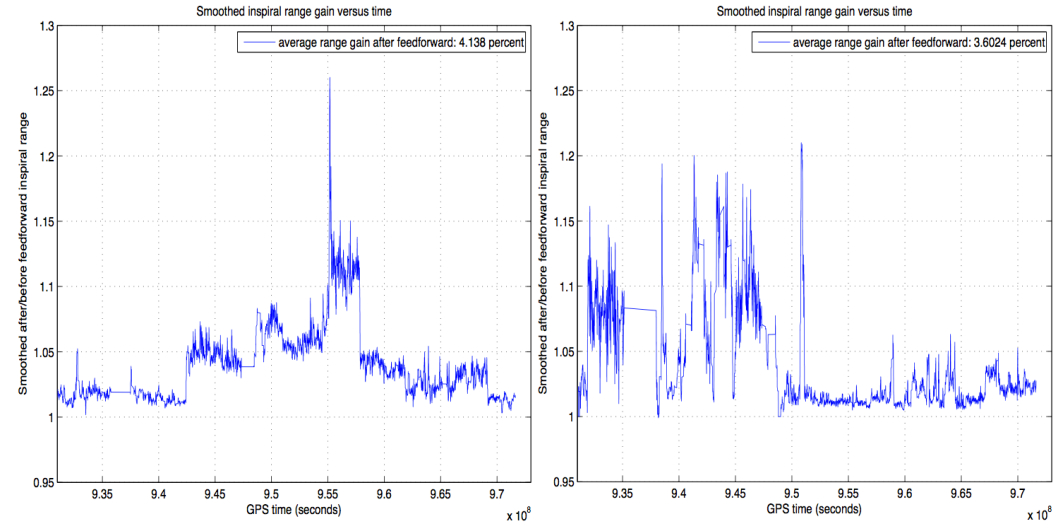
- Uses long science data to generate noise TF (as opposed to short injection TF)

Adaptive filter generation

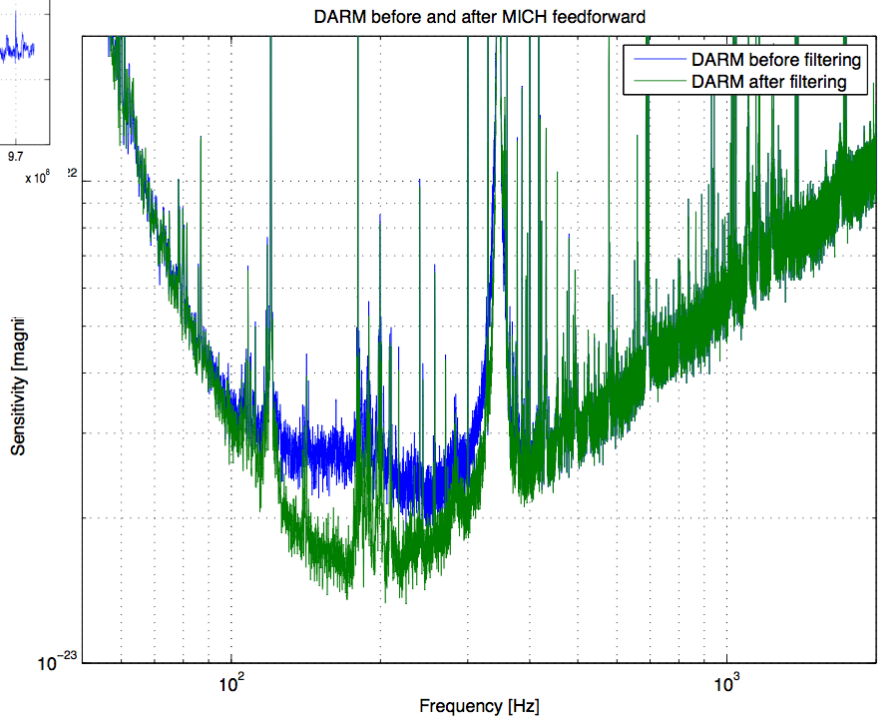
- Hands-free (according to Grant)
- Filters automatically updated and smoothly ramped/merged.



Entire S6 results: Very good



Inspiral range *fractional gain* vs time for S6
H1 (L) 4.13% better, L1 (R) 3.60% better



The code is there, mostly ready yet unused.

- Resurrect. Use. (There might be some bureaucracy regarding calibration and data analyses but it's worth it.)
- Adapt to some other noise sources. Use.

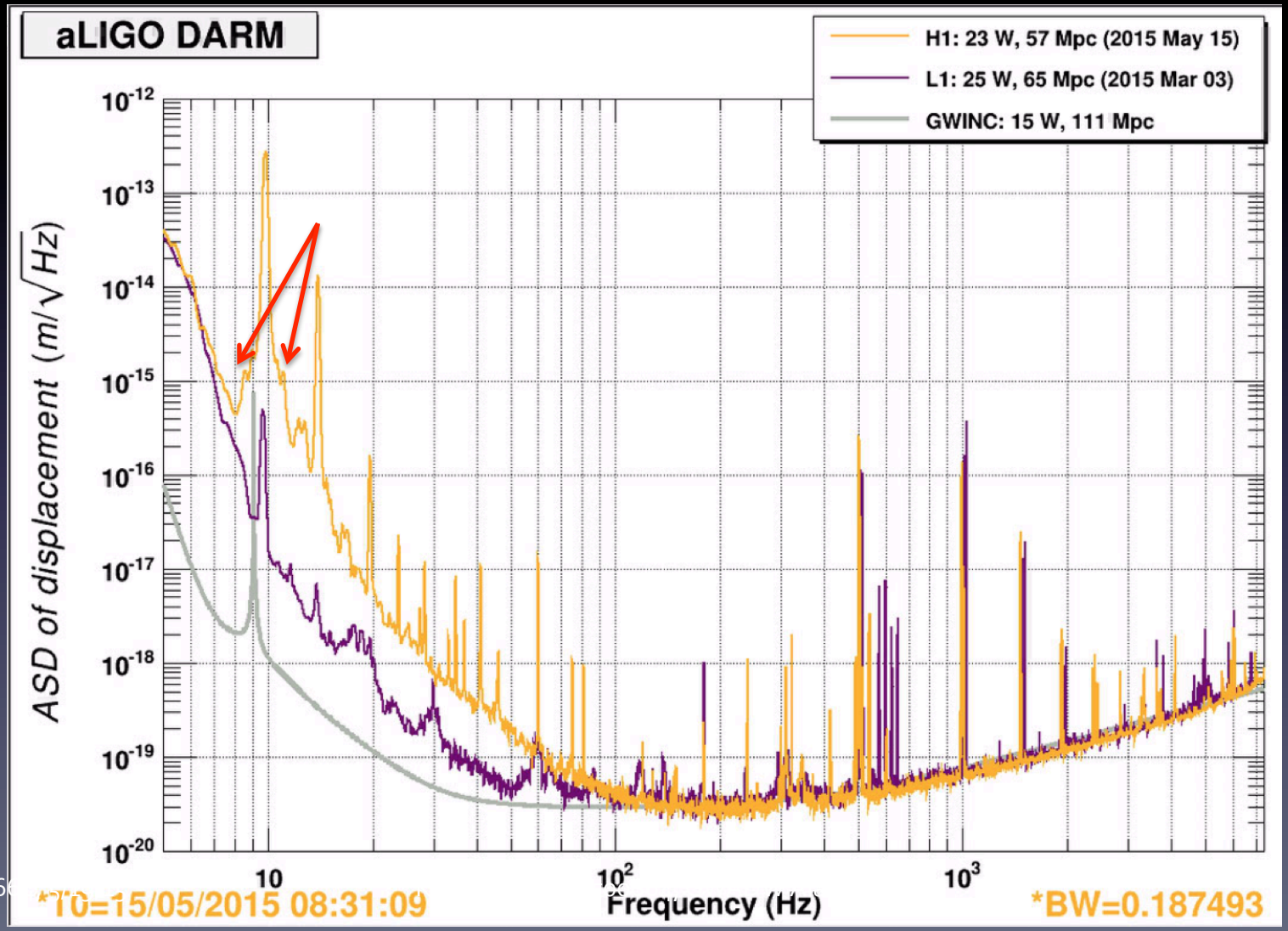
Bilinear coupling

- Noise = x(t) * y(t), e.g. A2L
- Many, many examples of these in IFO noise.
- (and of course there should be trilinear(??) and higher order coupling like x*y*z etc.)

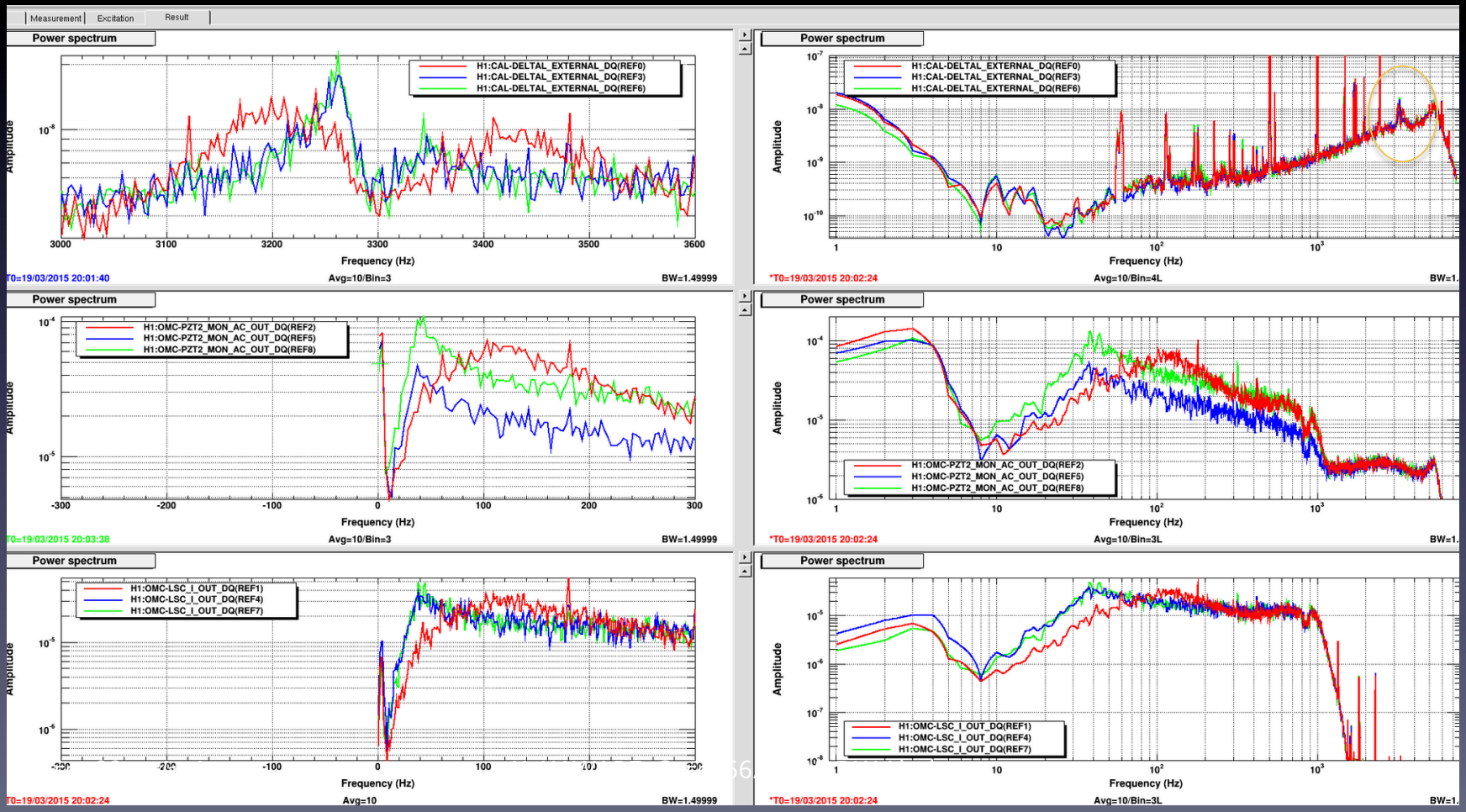
Ex) Low frequency noise upconverted by a line

- Around various lines (angle dither, length dither, mirror resonances, sus resonances etc.)

example


aLIGO DARM
H1: 23 W, 57 Mpc (2015 May 15)
L1: 25 W, 65 Mpc (2015 Mar 03)
GWINC: 15 W, 111 Mpc
ASD of displacement (m/√Hz)
10^-12
10^-13
10^-14
10^-15
10^-16
10^-17
10^-18
10^-19
10^-20
10
10^2
10^3
Frequency (Hz)
*T0=15/05/2015 08:31:09
*BW=0.187493

(example of self-inflicted damage which was already fixed: OMC dither and LF OMCL)

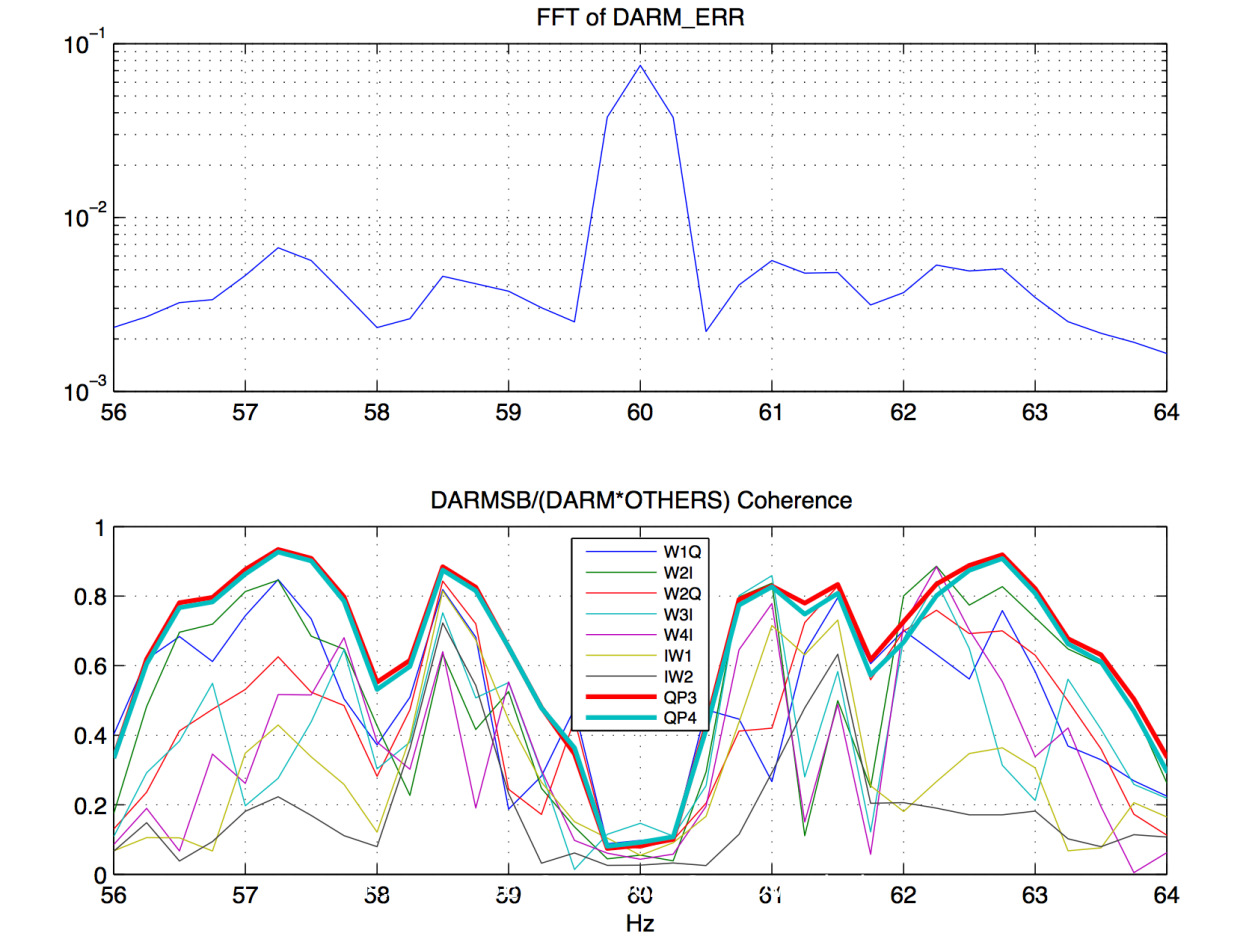


- Is bilinear (and higher order) coupling a major problem as of now for aLIGO?
  - No, there are more serious offenders than these at the moment (e.g. scattering)
  - Maybe yes, if e.g. high frequency srcl coupling is x*y^2 type
- Are these worth tackling?
  - Immediately: Depends (see above)
  - For improving science data quality: Yes

Tackle how?

- Look at easier ones first, i.e. line*LF
  - Simple convolution should (and does) allow you to see coherence between DARM and line*LF.

eLIGO Example:
coh[DARM(60Hz peak)*ASC(XHz), DARM(60+-XHz)]



Bilinear coupling Non-conclusion

- We can measure the coherence of bi-linear coupling noise and DARM.
- Coherence = the noise could be subtracted
  - Off-line or in real time
- I'll see (in the future) if I can make a reasonable subtraction of easy targets around aLIGO lines.