Non-stationary noise cancellation with causal, stable parametric filters

Gabriele Vajente

[vajente@caltech.edu]

LIGO Laboratory Caltech

1st Conference on Machine Learning for Gravitational Waves, Geophysics, Robotics, Control System

and CA17137 MC2 meeting

Cascina, January 14th 2019

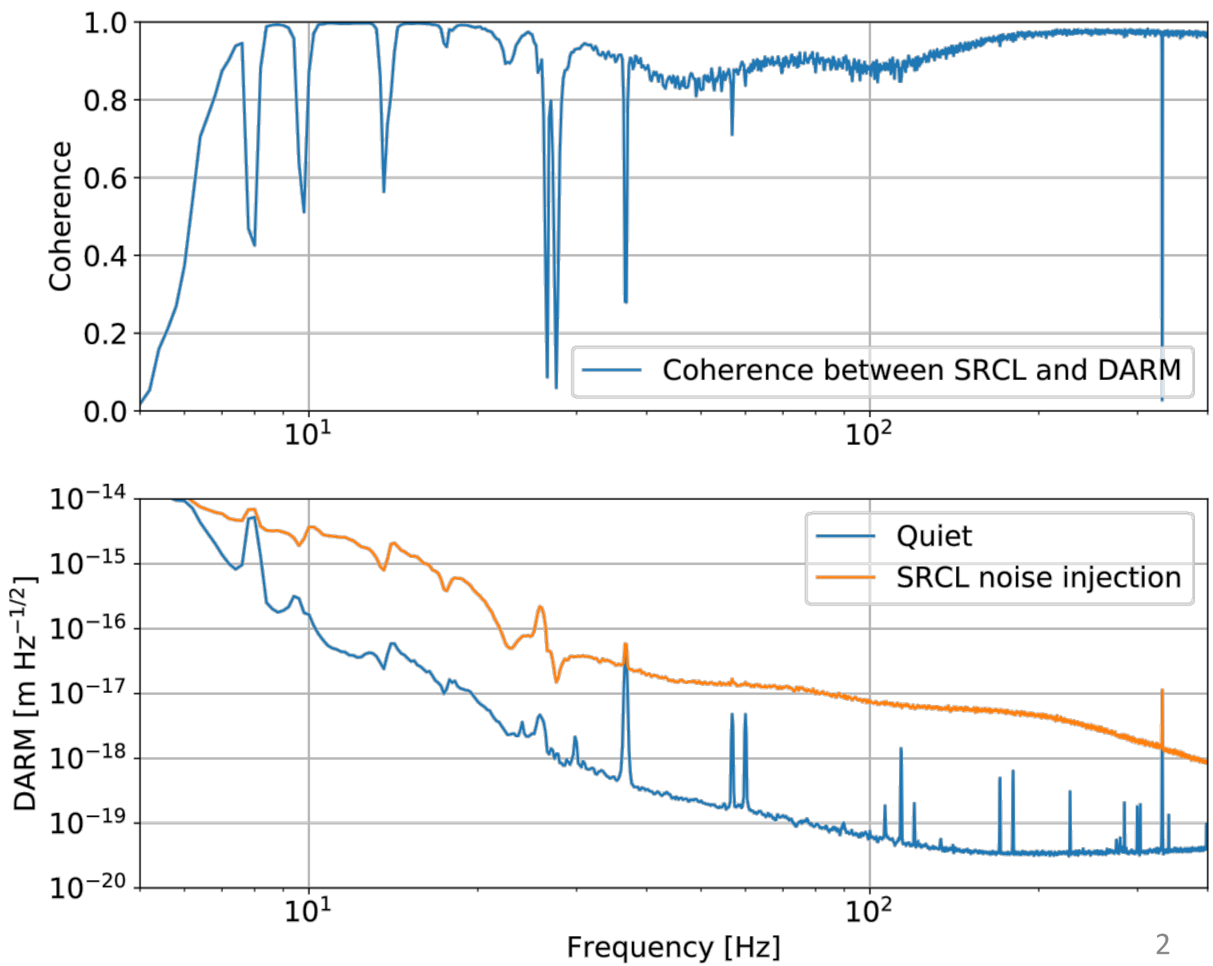
G2NET

COST ACTION CA17137
A NETWORK FOR GRAVITATIONAL WAVES, GEOPHYSICS AND MACHINE LEARNING

LIGO-G1900064-v2

Introduction

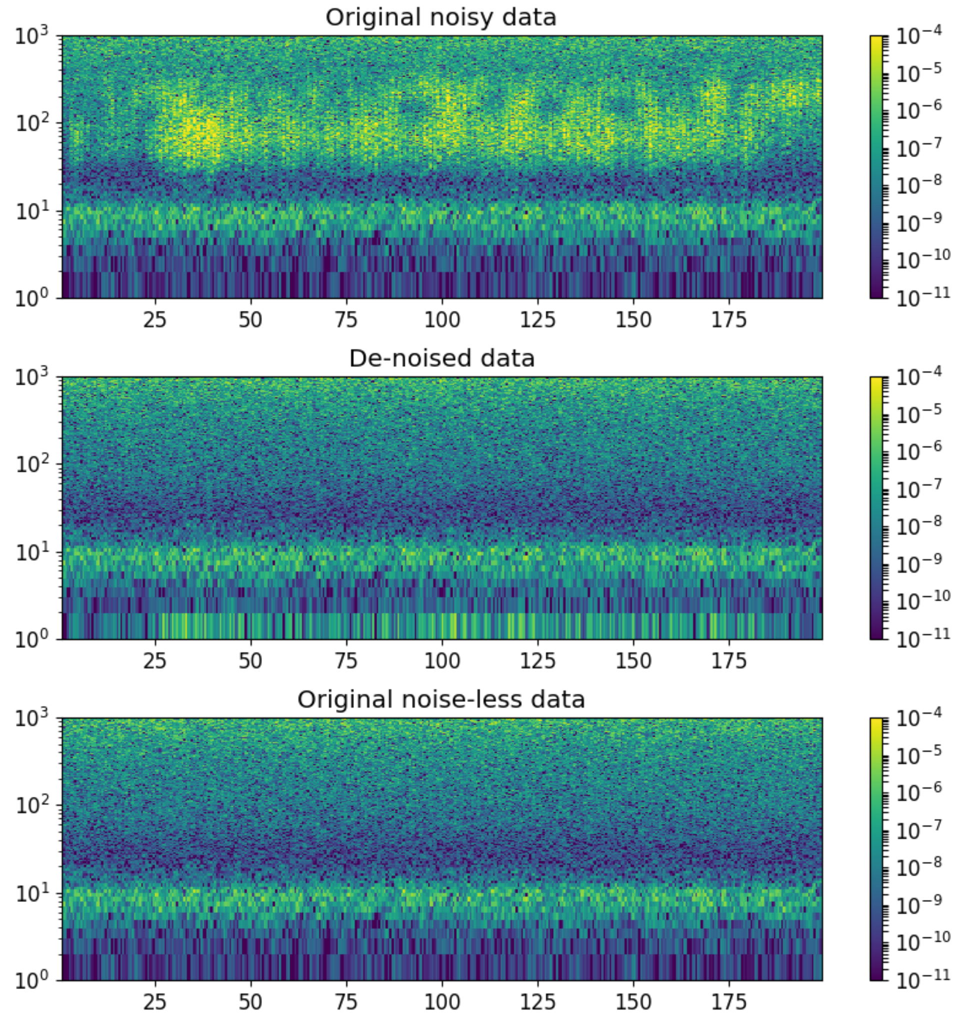
- **Motivation**:
  - noise couplings in GW detectors are often non-stationary
  - Coupling changes can often be traced to "slow" motions of the interferometer (IFO) as for example angular degree of freedom fluctuations
- Examples:
  - SRCL noise coupling at LIGO Hanford,
  - O2 jitter noise,
  - 40m lab seismic feed-forward
- **Goal**:
  - develop a technique to identify and efficiently subtract non-stationary noise couplings
- **By-product:**
  - a parametric, stable IIR noise subtraction (a-la Wiener filter)



https://git.ligo.org/gabriele-vajente/nonsens

Previous attempts with neural networks

- Alberto Iess' work based largely on **[dn]²: denoising with deep neural networks**
  - **LIGO-G1800334**
  - https://git.ligo.org/gabriele-vajente/dn2
- Also DeepClean
  - **LIGO- G1801716**
- Works with simulated data fairly well
- Issue:
  - Difficult to train (mixture of slow dynamics and fast sampling)
  - Interpretability



Mathematical foundations

LIGO-T1800525

$$h(t) = H[s(t)] + \sum_{i=1}^{N} \alpha_i [x_i(t) s(t)]$$

- **h(t)**: target signal, what we want to clean (example: GW strain)
- **s(t)**: noise witness signal, i.e. a measurement of the noise that couples into h(t), through modulated transfer functions (example: SRCL control signal)
- **x_i(t)**: a set of auxiliary signals that witness the coupling modulation (example: angular degree of freedom fluctuations)

- Assuming that x_i(t) varies on time scales much slower than the noise s(t)
- The **stationary coupling** is modeled with a transfer function H
- The **non-stationary couplings** are modeled by assuming the noise couples through (many) stationary transfer functions, each one modulated by one of the witness signals

https://git.ligo.org/gabriele-vajente/nonsens

Implementation / 1

LIGO-T1800525

- Frequency domain approach
  - Find the optimal solution, independently for each frequency bin (frequency-domain a-causal Wiener filter [1])

Modulated signals

$$s_i(t) = \begin{cases} s(t) & \text{if } i = 0 \\ x_i(t)s(t) & \text{if } i = 1,\ldots,N \end{cases}$$

Noise prediction

$$y(t) = \sum_{i=0}^{N} \alpha_i [s_i(t)]$$

Residual after subtraction

$$\tilde{r}(\omega) = \tilde{h}(\omega) - \tilde{y}(\omega) = \tilde{h}(\omega) - \sum_{i=0}^{N} \tilde{\alpha}_i(\omega)\tilde{s}_i(\omega)$$

Minimize PSD of residual

$$S[r,r] = S[h,h] - (\boldsymbol{\alpha}^{+}\mathbf{H} + \mathbf{H}^{+}\boldsymbol{\alpha}) + \boldsymbol{\alpha}^{+}\mathbf{S}\boldsymbol{\alpha}$$

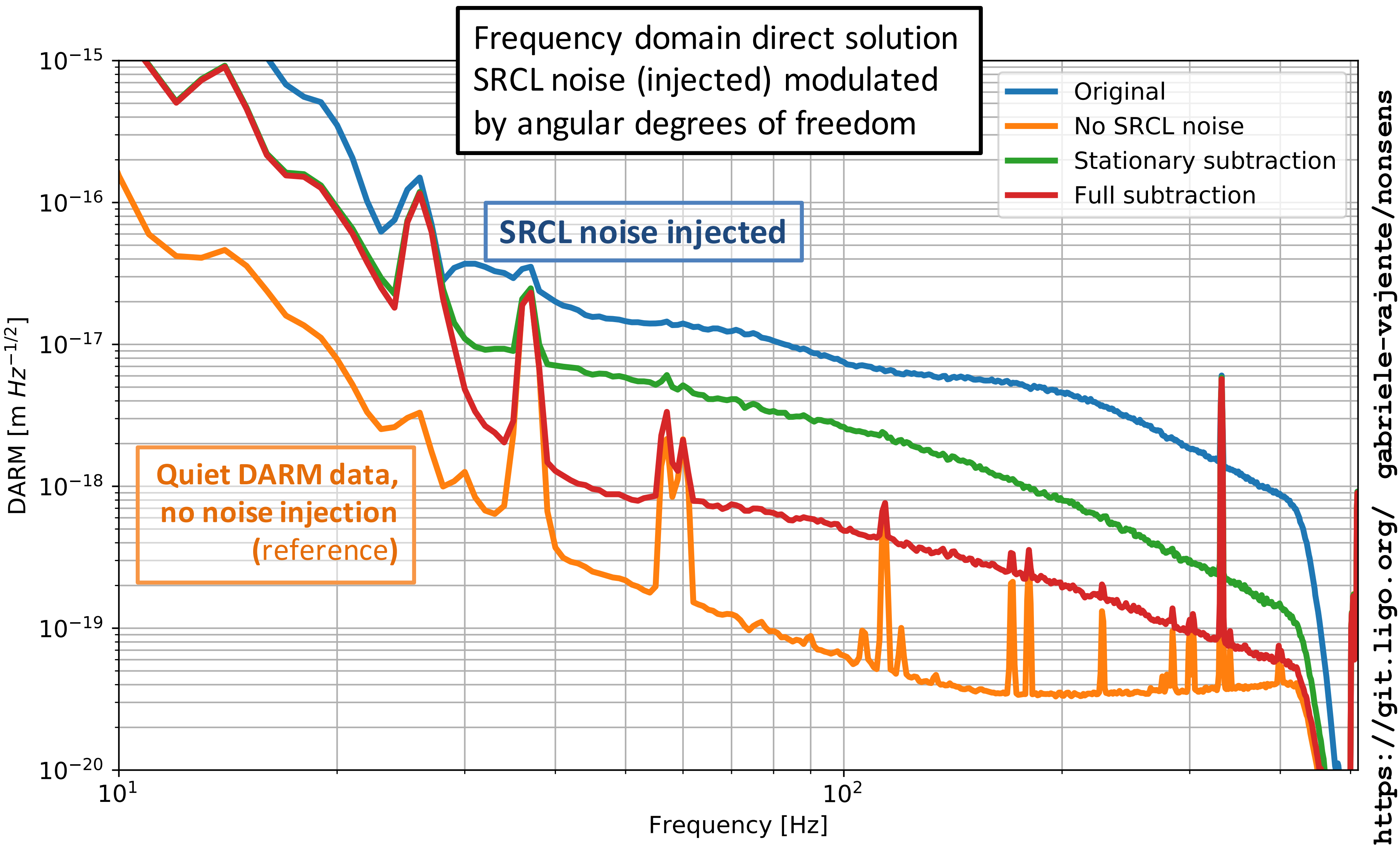
Optimal solution

$$\boldsymbol{\alpha} = \boldsymbol{S}^{-1}\boldsymbol{H}$$

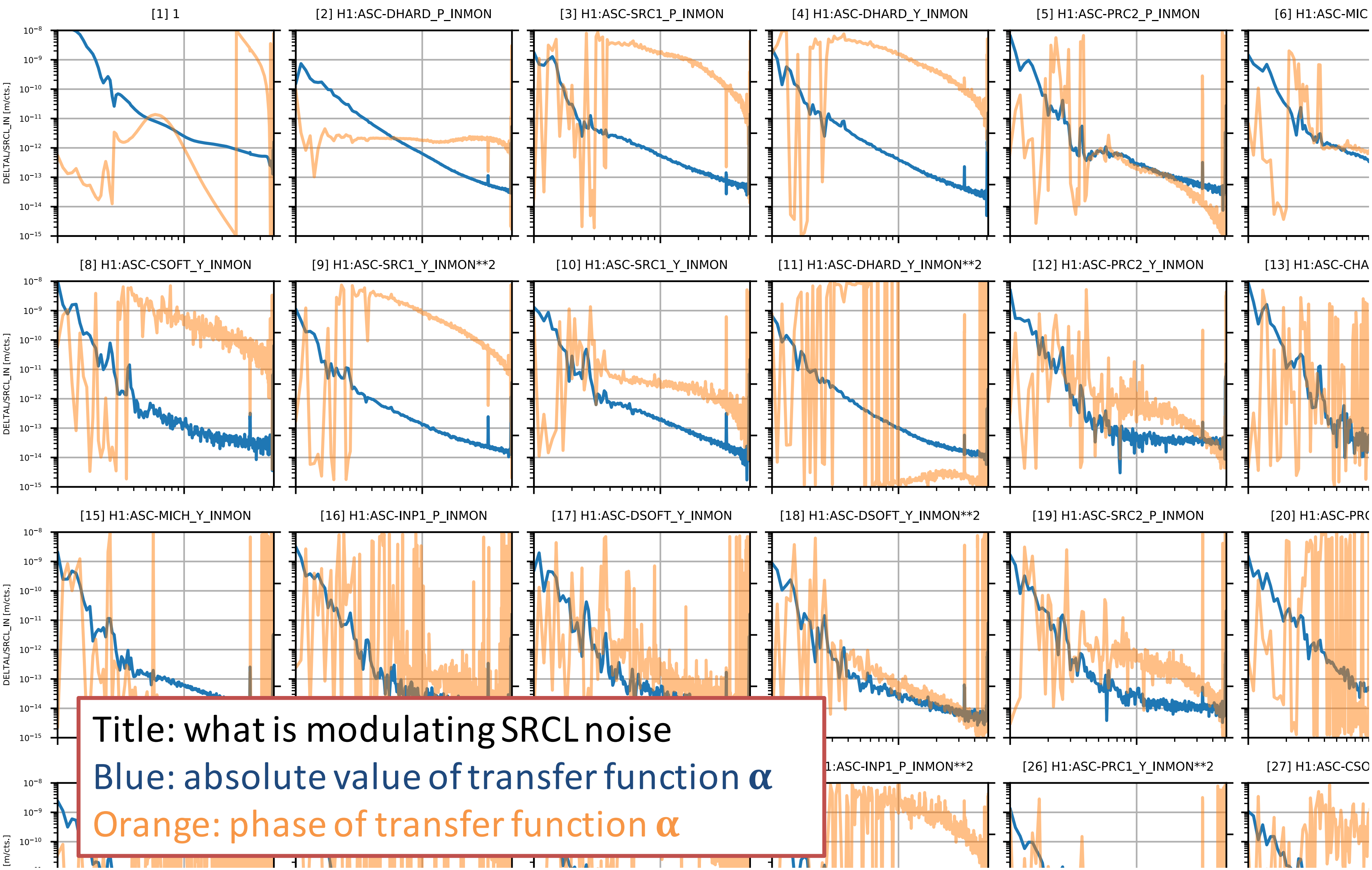
https://git.ligo.org/gabriele-vajente/nonsens

[1] Wiener, Norbert (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York: Wiley.

Example: SRCL noise



Example: SRCL noise



Title: what is modulating SRCL noise

Blue: absolute value of transfer function α

Orange: phase of transfer function α

https://git.ligo.org/gabriele-vajente/nonsens

Frequency domain approach

Pros

- Fast
- Quadratic optimization
- Guaranteed optimal solution

Cons

- Every bin is independent
- Huge number of parameters > overfitting
- No guarantee that the solutions are physically realizable filters in time domain (causality, stability, etc…)

https://git.ligo.org/ gabriele-vajente/nonsens

Parametrized approach

LIGO-T1800525

- Describe each transfer function using a (non-linear) parametrized form
- Laplace domain: $\alpha_h(s) = \frac{\sum_{i=0}^{N_N} b_i s^i}{\sum_{j=0}^{N_D} a_j s^j}$
- Define a frequency-integrated cost function

$$\mathcal{C}(\theta_{h,i}) = \frac{1}{\omega_2 - \omega_1} \int_{\omega_1}^{\omega_2} W(\omega) S_{rr}(\omega) d\omega \qquad W(\omega) = \frac{1}{S_{hh}(\omega)}$$

- Use gradient-based optimization algorithms

$$\frac{\partial \mathcal{C}}{\partial \theta_{h,n}} = \frac{1}{\omega_2 - \omega_1} \int_{\omega_1}^{\omega_2} W(\omega) \left[\frac{\partial S_{rr}}{\partial \alpha_h} \cdot \frac{\partial \alpha_h}{\partial \theta_{h,n}} + \frac{\partial S_{rr}}{\partial {\alpha_h}^*} \cdot \frac{\partial {\alpha_h}^*}{\partial \theta_{h,n}} \right] d\omega$$

https://git.ligo.org/gabriele-vajente/nonsens

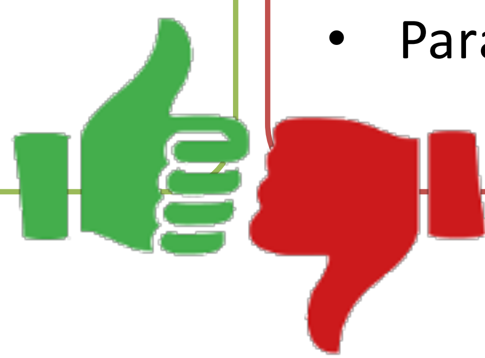
Parametric approach

Pros

- "Smooth" transfer functions
- Small number of parameters
- No overfitting

Cons

- No guarantee of convergence to optimal solution
- No guarantee that the solutions are physically realizable filters in time domain (**causality, stability**, etc...)
- Parameters are poorly scaled



https://git.ligo.org/gabriele-vajente/nonsens

Additive second order stages

- How to enforce stability and improve the coefficient scaling

$$\alpha_h(s) = \frac{\sum_{i=0}^{N_N} b_s s^i}{\sum_{j=0}^{N_D} b_s s^j}$$

$$\alpha(s) = k + \sum_{i=1}^{N_D} \frac{\rho_i}{s - p_i}$$

$$\alpha(s) = k + \sum_i \left[\frac{\rho_i}{s-p_i} + \frac{\rho_i^*}{s-p_i^*}\right] + \sum_j \frac{r_j}{s-f_j}$$

$$\alpha_h(s) = k_h + \sum_{i=1}^{N_D/2} \frac{b_{h,1}s + b_{h,0}}{s^2 + a_{h,1}s + a_{h,0}}$$

Poles are stable

$\mathcal{R}(p_i) < 0$ and $f_i < 0$

if and only if

$a_{h,i} > 0$

https://git.ligo.org/gabriele-vajente/nonsens

LIGO

Implementation / 2

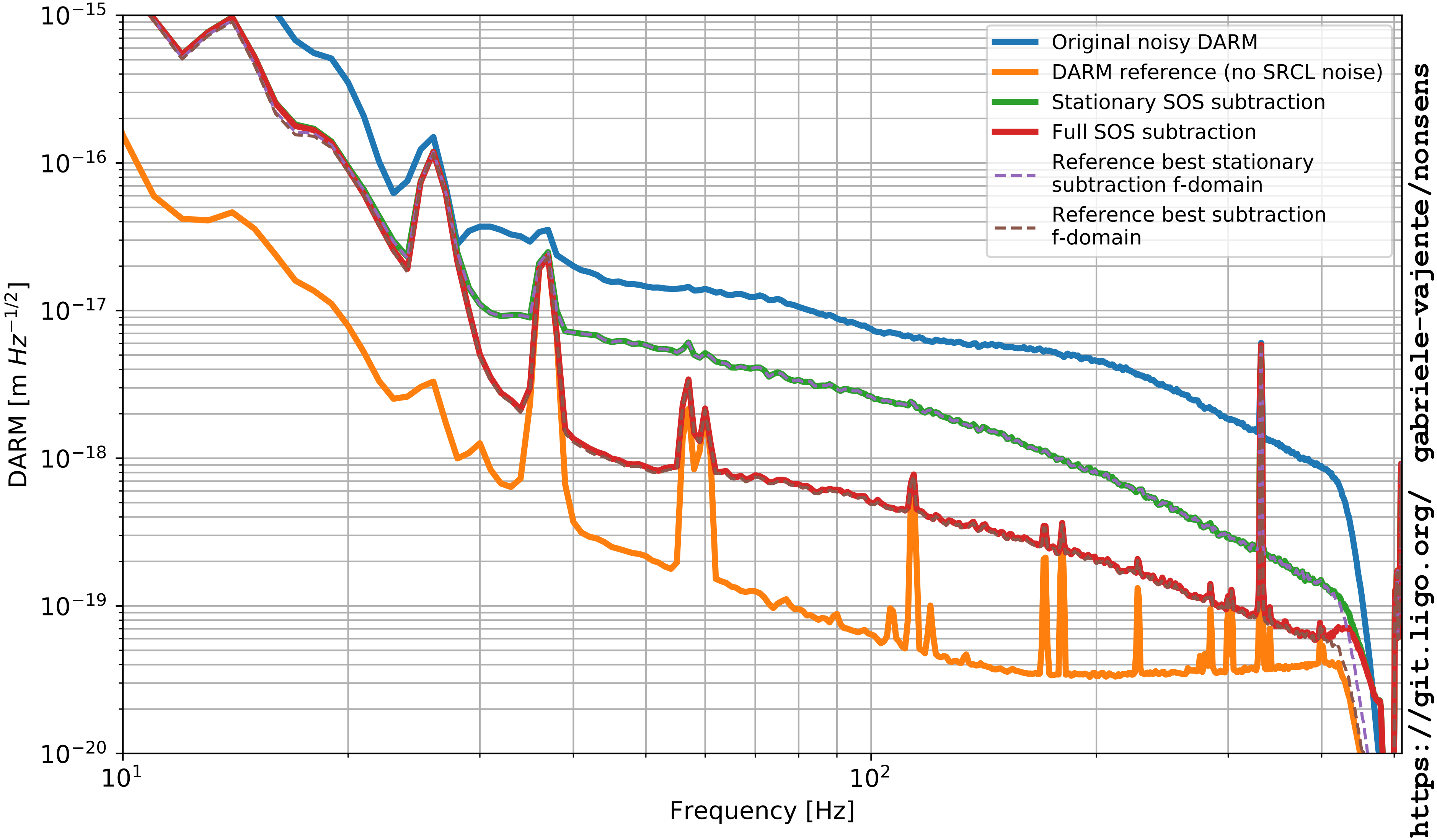
- Parametrize with sum of second order stages
- **Enforce stability** by positive mapping of a coefficients, for example $a_{h,i} \longrightarrow \exp[a_{h,i}]$ or using bounded functions such as a scaled sigmoid (allowing for limit on maximum and minimum pole frequencies)

$$\sigma(x,H) = \frac{H}{1+He^{-x}}$$

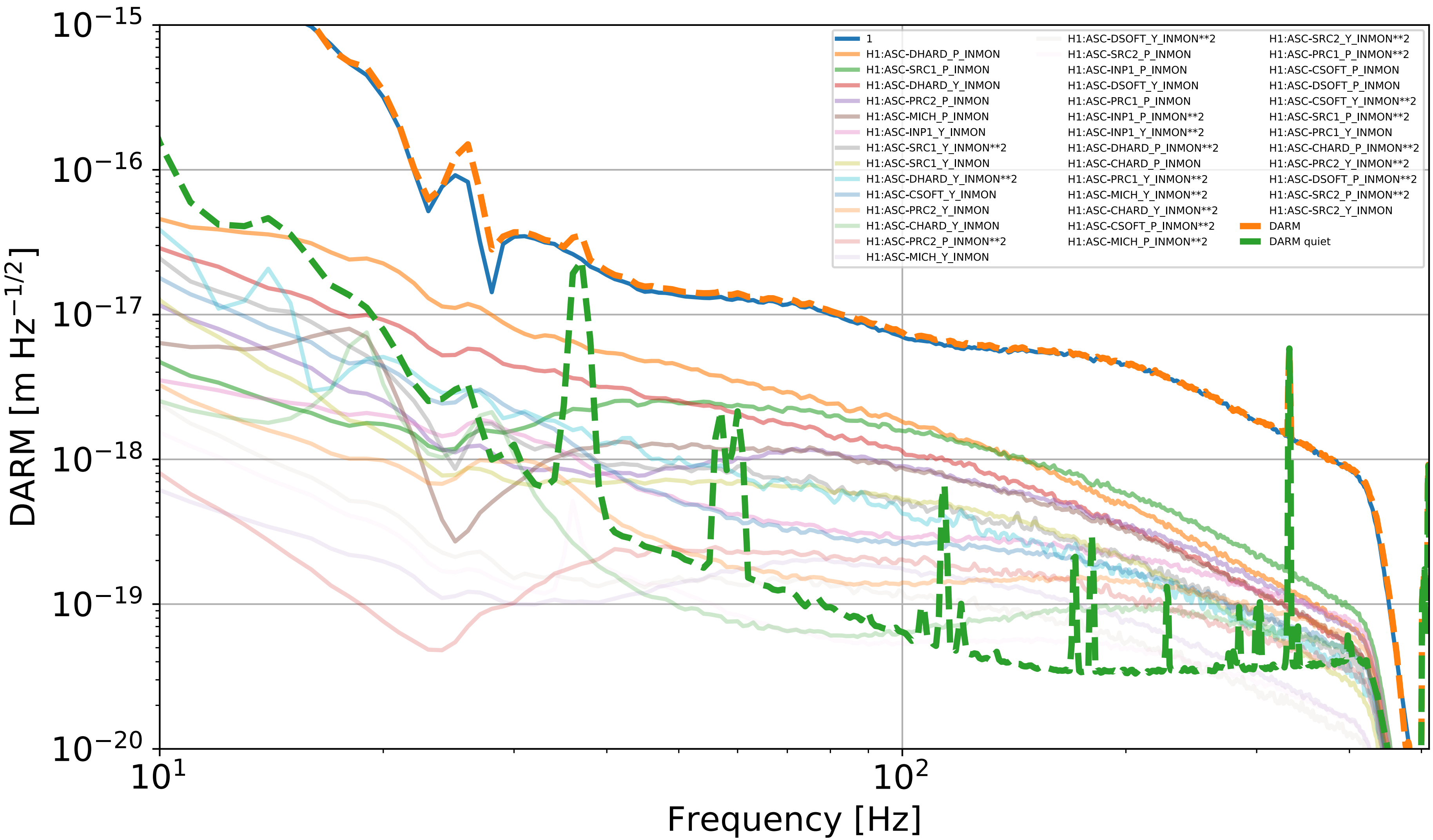
- Gradient can still be computed directly
- Use **gradient-based unconstrained minimization** methods:
  - ADAM (inspired from deep learning)
- With small additional re-parametrization one can also enforce that the Q of all poles is not large.

https://git.ligo.org/ gabriele-vajente/nonsens

Example: SRCL noise

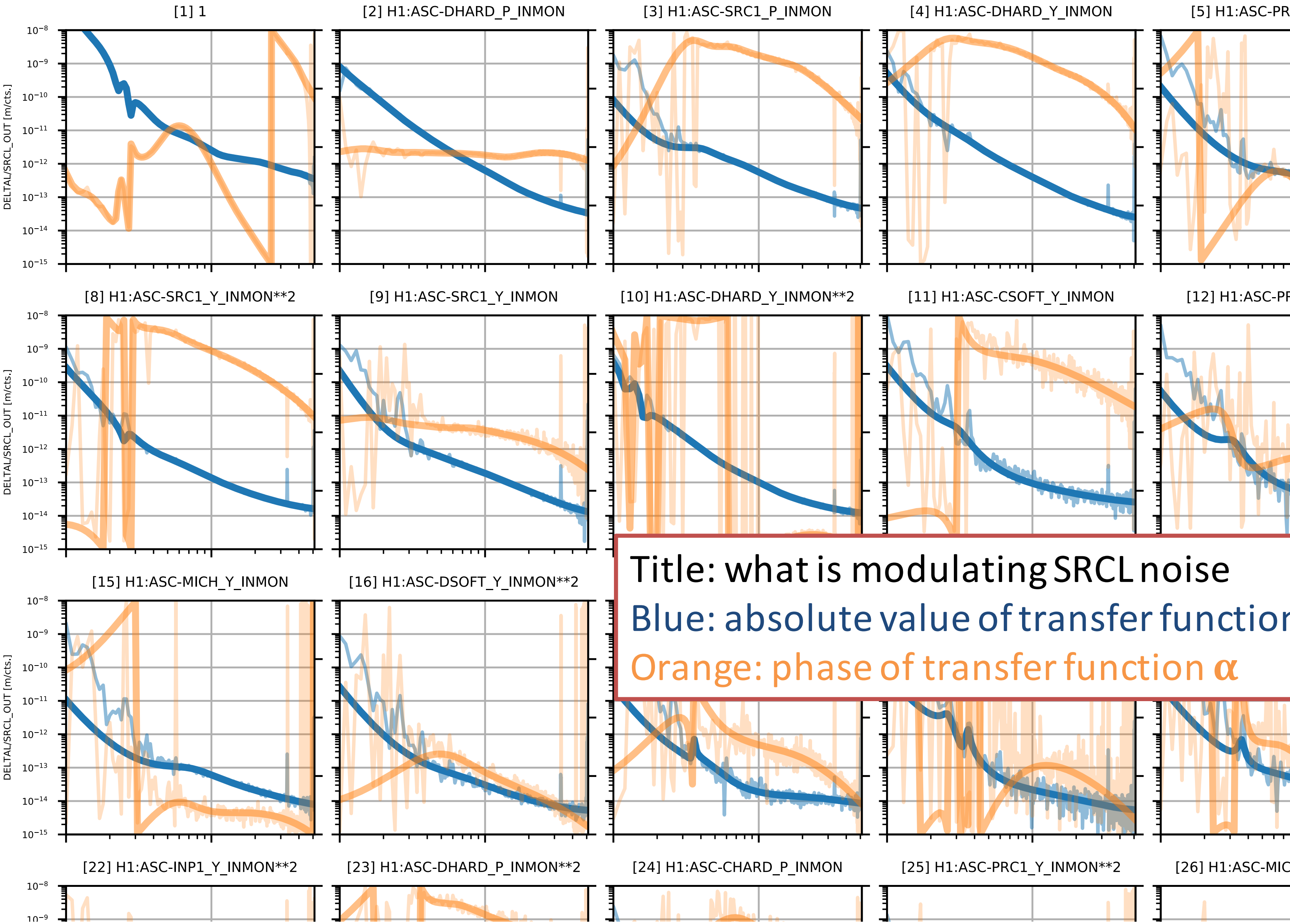


Example: SRCL noise



https://git.ligo.org/ gabriele-vajente/nonsens

Example: SRCL noise



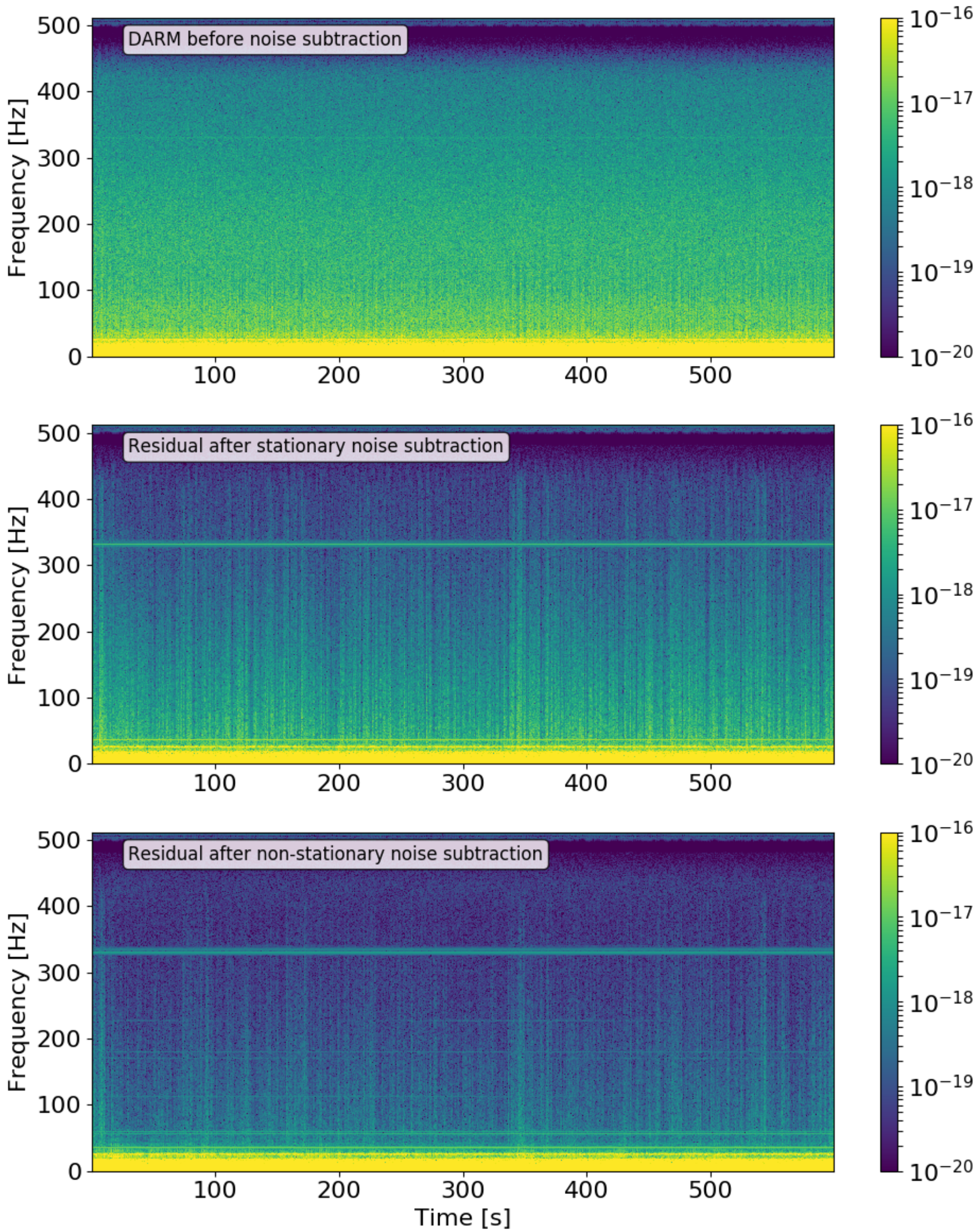
Title: what is modulating SRCL noise

Blue: absolute value of transfer function α

Orange: phase of transfer function α

https://git.ligo.org/gabriele-vajente/nonsens

Example: SRCL noise



- After stationary noise subtraction: still large non-stationary noise
- Non-stationary noise subtraction improves a lot
- There is still some non-stationarity left: modulations that are not captured by the angular signals

https://git.ligo.org/gabriele-vajente/nonsens

Time domain implementation

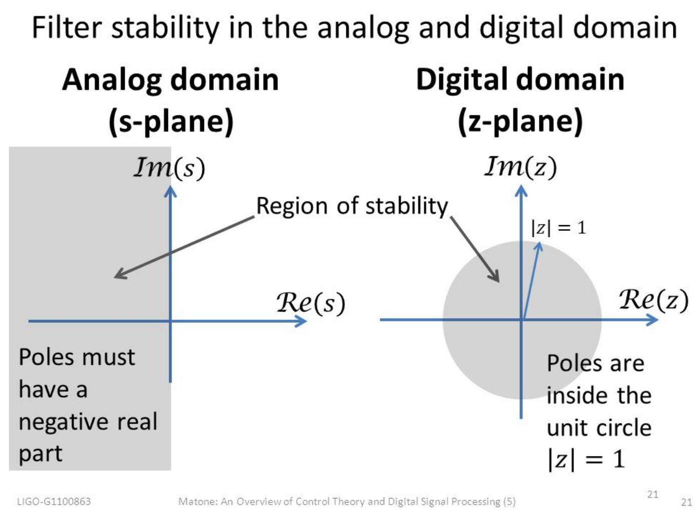
LIGO-T1800525

Additive SOS coefficients → s-domain transfer function → Bilinear transform → IIR filter coefficients

- Or parametrize directly in z-domain using starred transform

$$z = e^{sT} \qquad T = f_s^{-1}$$

$$\alpha_h(z) = k_h + \sum_{i=1}^{N_{SOS}} \frac{b_{h,1} z + b_{h,0}}{z^2 + a_{h,1} z + a_{h,0}}$$



https://slideplayer.com/slide/4469125/

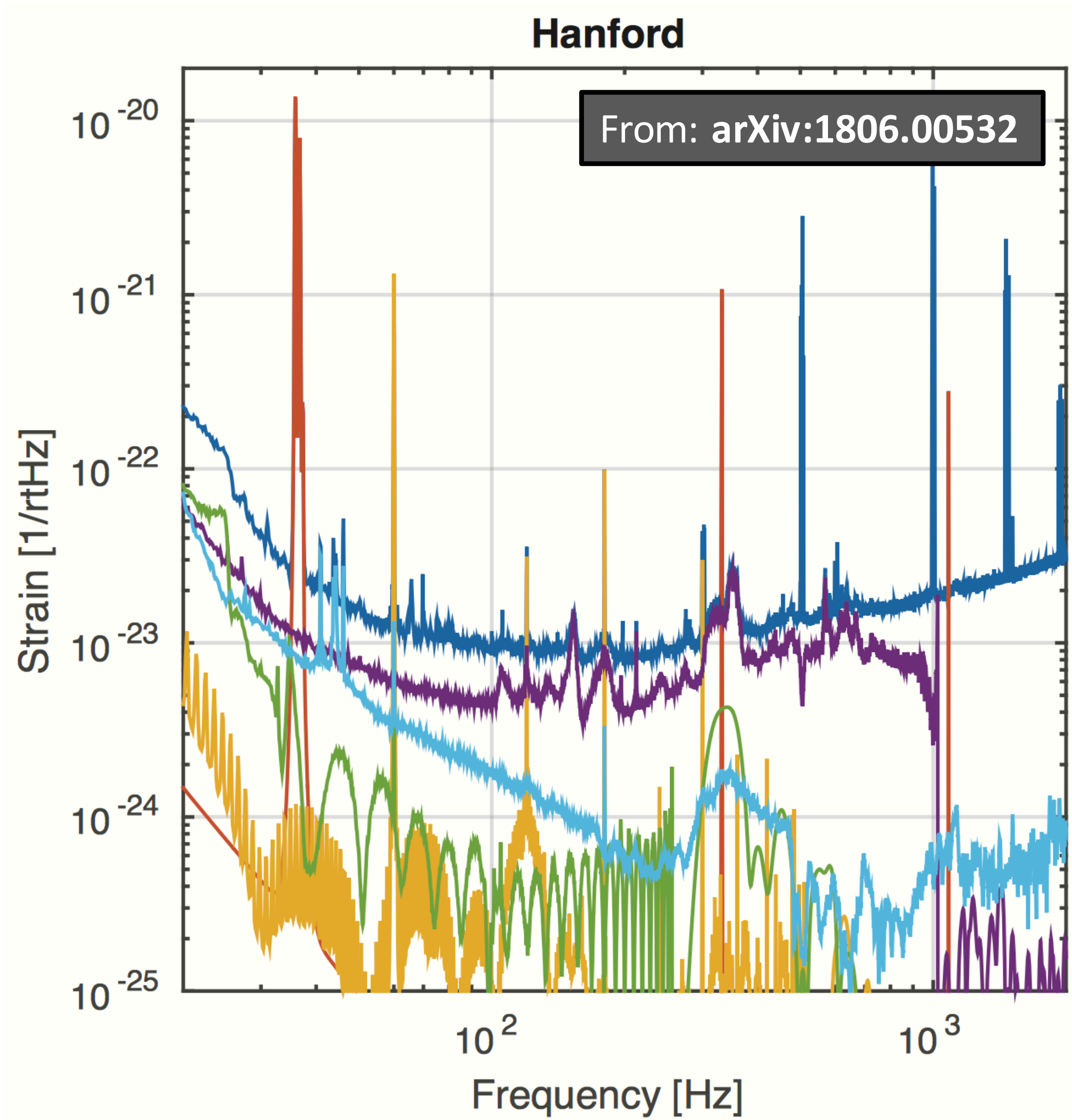
- Poles are stable (inside the unit circle in z-plane) if and only if

$$-1 < a_0 < 1$$

$$-1 - a_0 < a_1 < 1 + a_0$$

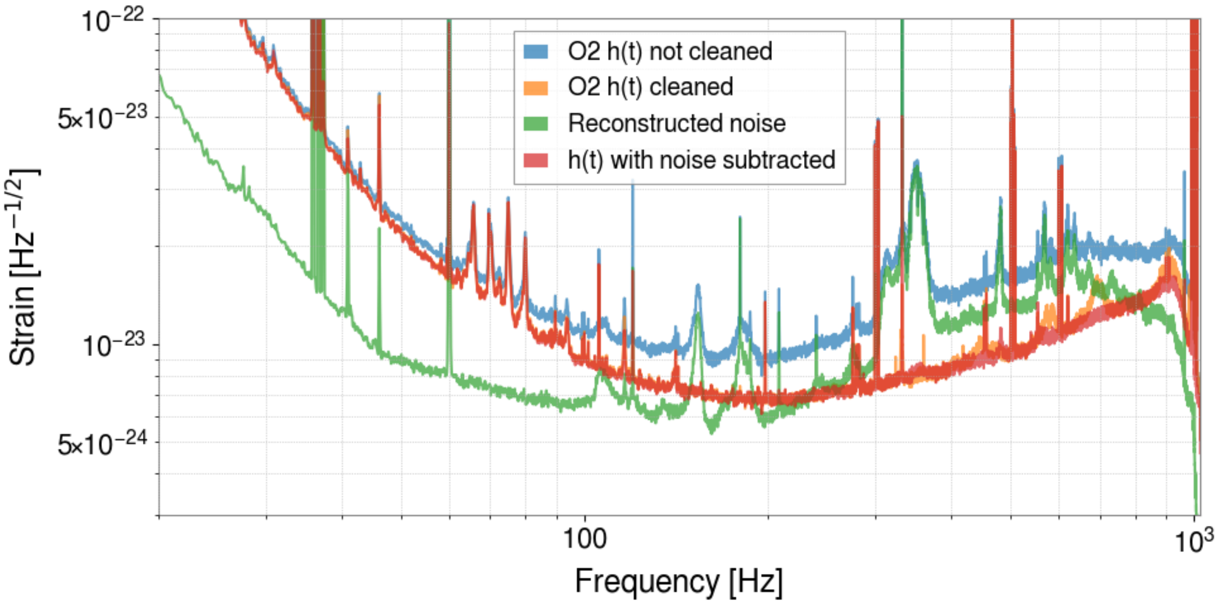
https://git.ligo.org/gabriele-vajente/nonsens

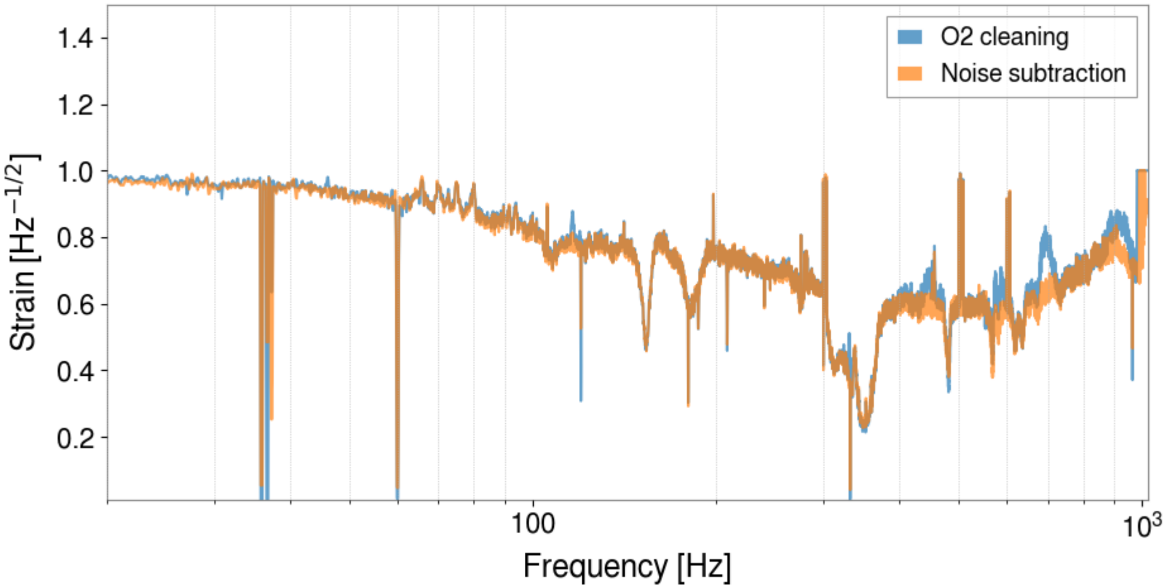
O2 noise subtraction at LIGO Hanford



- During O2, Hanford sensitivity **limited by input beam jitter** (mostly)
- Solved by **offline noise subtraction**
- Using signals witnessing beam jitter (and other noises)
- Performed in **frequency domain** (a-causal Wiener filters)
- Worked very well

Parametric time-domain subtraction

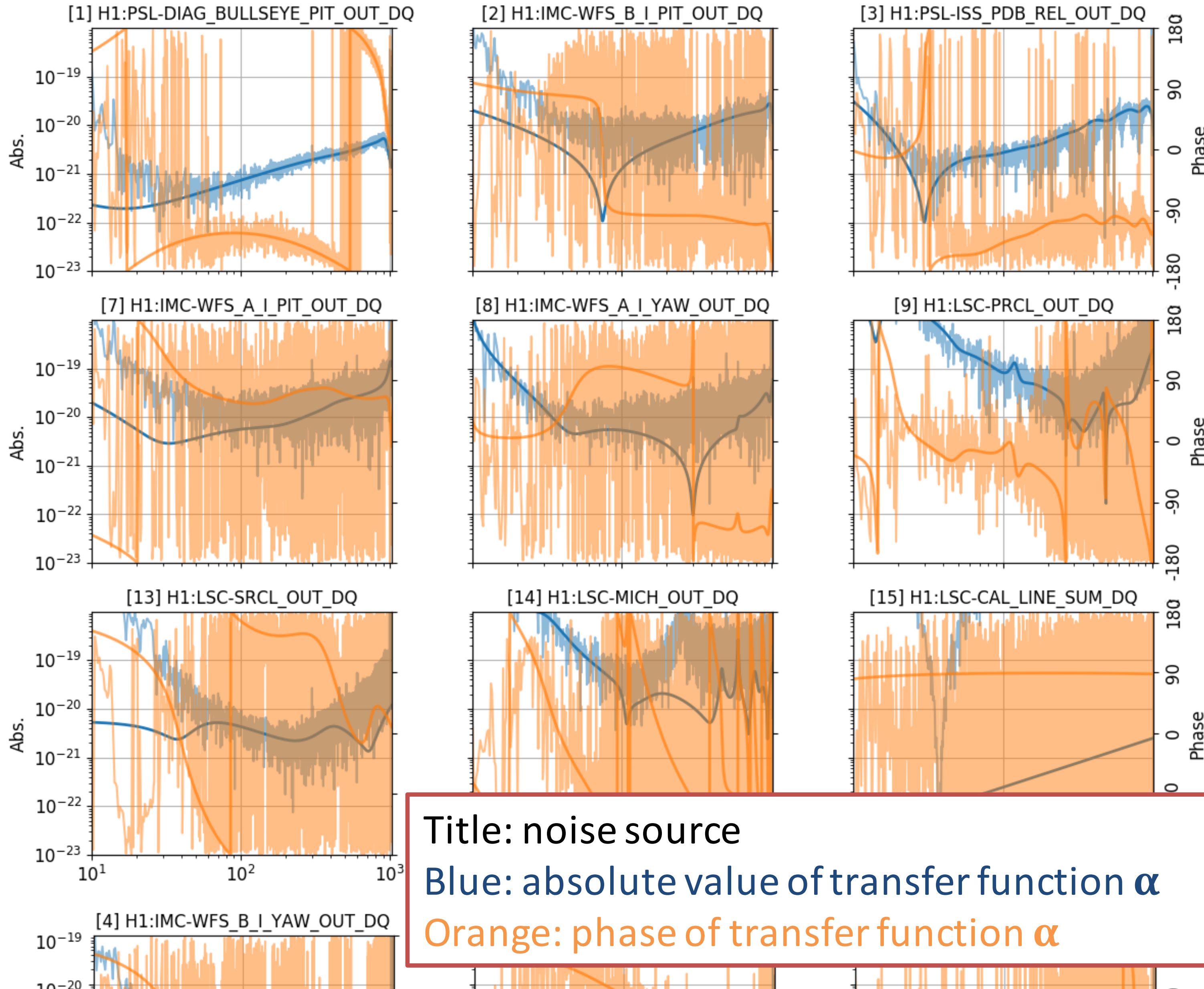




- Solution that uses IIR filters in time domain: stable and causal
- Same (maybe *a bit better*) performance as O2 offline subtraction
- **IIR filters** could be implemented in real time if desired

LIGO-T1800552

Some of the transfer functions



Title: noise source

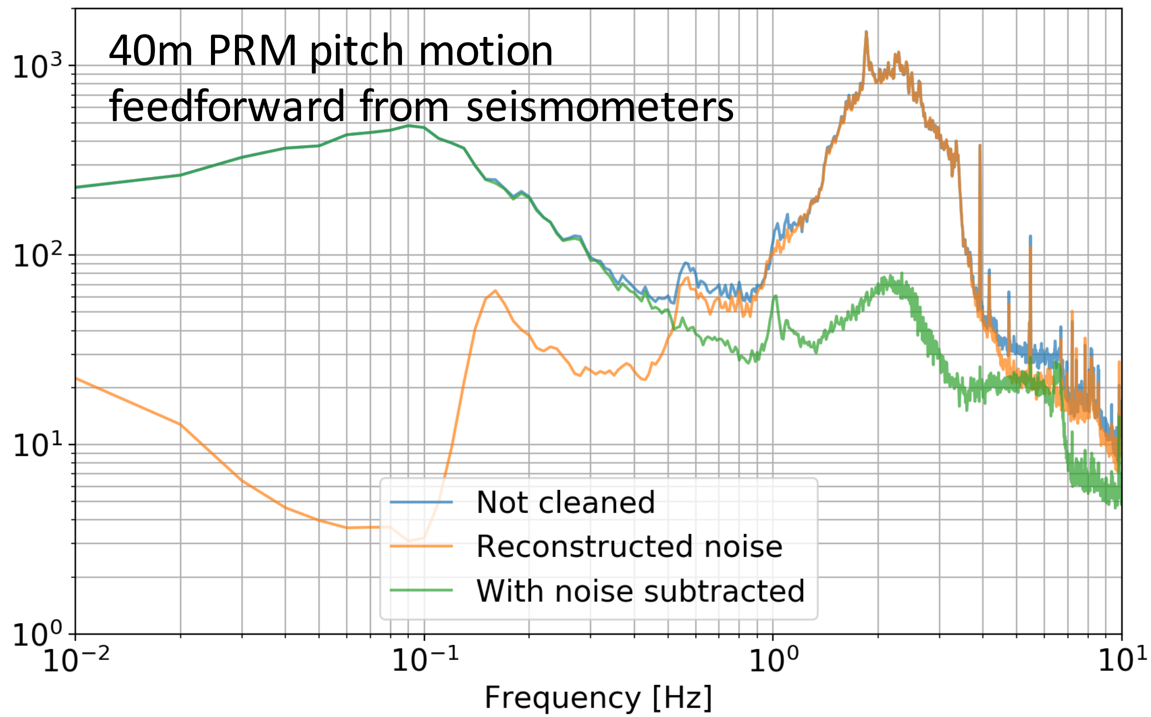
Blue: absolute value of transfer function α

Orange: phase of transfer function α

LIGO-T1800552

Other applications / next steps

- Tested also on seismic feed-forward at the 40m interferometer
- Other possible applications of the non-stationary subtraction:
  - Angular noises
  - Environmental noises
  - Intensity and frequency noise



- Next steps:
  - This algorithm can be made <u>**adaptive**</u>: starting from an estimate of the transfer functions, refinements can be computed from cross-spectral densities (see LIGO-T1800525-v3)

References

The basic idea and its application to non-stationary noise subtraction
https://dcc.ligo.org/LIGO-T1800525
and the application to stationary noise subtraction (using O2 as example)
https://dcc.ligo.org/LIGO-T1800552

Some details were described in the LHO elogs
https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=45403
Characterizing SRCL non stationary noise coupling
https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=45508
Time-domain non-stationary subtraction
https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=45803
SRCL noise subtraction
https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=45823
SRCL non-stationary noise coupling

Code available on git.ligo
https://git.ligo.org/ gabriele-vajente/nonsens