LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note **LIGO-T2100342-v1** 2021/09/23

Non-Linear Noise Subtraction for Low Frequency

Yuka Lin, Gabriele Vajente (mentor)
Embry-Riddle Aeronautical University, California Institute of Technology

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW17-161
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

1 Abstract

The amplitude of the noise in laser interferometric data limits the astrophysical information that can be extracted from it. LIGO has a strong history in reducing the linear and stationary noise at different frequencies by monitoring auxiliary sensors and the correlation with the estimated strain at a given time. Recently, it was shown that nonlinear correlations could be used to reduce the noise even further for the case of the noise spectral density around 60 Hz in laser interferometers.[1] The approach involved utilizing two types of auxiliary channels, each with different spectral content. In this project, a similar methodology will be investigated for the lower part of the LIGO spectrum (below 10 HZ) which has not been explored in any previous subtractions. The gravitational wave memory from Core Collapse Supernovae and pre-merger binary star signals are known to have low-frequency emission between 10^{-5} Hz and 50 Hz, which is the main motivation for this project. [2]

2 Introduction

2.1 Astrophysics

Gravitational waves have become a widely used astronomical tool in modern astrophysics in recent years. Predicted by Albert Einstein in 1916, the existence of gravitational waves was not fully proven until 2016 by the LIGO (Laser Interferometer Gravitational-Wave Observatory) [3] scientific collaboration. LIGO utilizes laser interferometers to measure the microscopic deformations in space-time caused by transient gravitational waves. Different features of the same sources emit gravitational waves at different frequencies. Ground-based laser interferometers have a sensitivity that depends on the specific sources of noise at a certain frequency. The lower frequency regime between 10^{-5} Hz and 50 Hz is particularly challenging because of the noise (i.e. ground vibrations and control systems noise, which are likely to be relevant sources at lower frequencies) that can interfere with the detector instruments.[4] These types of noise tend to "couple" or leak into the main signal and thereby producing sources of noise disturbances which then limits the sensitivity of the detectors. Noise coupling is defined as the physical process of adding some noise sources (such as the ones mentioned previously) to the gravitational wave strain output.

Nonetheless, there are interesting sources of gravitational waves at those lower frequencies to study. In particular, the gravitational wave memory from Core-Collapse Supernovae [5] and pre-merger binary star signals are found in the lower gravitational wave frequency regime between 10^{-5} Hz and 50 Hz.[2] For an event such as a galactic supernova, a fraction of the gravitational wave memory might be above the amplitude of the noise floor at frequencies below 50 Hz; therefore, reducing the noise floor as much as possible would make it easier to extract those features.

Generally, contemporary aLIGO (advanced LIGO) [6] noise reduction methods can focus on

reducing the impact of noise sources that are linearly coupled with auxiliary channels. [7] Advancements in noise reduction techniques have allowed physicists to develop a method in which algorithms can be trained to reduce non-stationary noise couplings by using auxiliary channels from LIGO's detectors. Non-stationary noises tend to vary over a period of time while noise that is stationary would remain constant. In particular, Vajente et al.[1] demonstrated how to reduce the noise contributions to the strain channel (which contains gravitational wave signals) by also reducing noise that are coupled non-linearly and non-stationary using an algorithm he created called NonSENS (NON-Stationary Estimation of Noise Subtraction) [8]. This algorithm was then used to successfully reduce the noise produced by a 60 Hz power line and ASC (alignment sensing and controls) noise between 10-30 Hz. [8] As of right now, the only official aLIGO calibrated gravitational wave data that is below 10 Hz is the CAL-DELTAL_EXTERNAL_DQ; however, it does not exhibit the same precision as the other data that are available (calibrated above the frequency range).[9] Therefore, it is of interest to examine further if subtraction would be viable below 10 Hz, as it has not been done before. [1] The objective of this project is to utilize the NonSENS algorithm to perform noise reductions below 10 Hz.

3 Objective

The main objective of this project will be to apply the algorithm "NonSENS" for subtracting non-stationary noise to perform noise reductions between 1 and 10 Hz. This project will focus on seeing if it is possible to do this for a lower frequency range then the one done previously with the ASC subtraction [8] by figuring out which LIGO auxiliary channels can be utilize that will be able to perform the subtraction to a gravitational wave strain channel.

4 Project Outline and Approach

4.1 Linearity and Stationarity

A *system* in signal processing is a process in which an output signal is produced as a result of the response to an input signal. The simplest example of a system is as follows:

$$x(t) \longrightarrow system \longrightarrow y(t) \tag{1}$$

where $x(t)$ is the input signal that goes through a system and the $y(t)$ is the resulting output signal that is produced. Systems can be categorized as being linear or non-linear. A system is considered linear if it obeys the *Principle of Superposition*. In particular, this principle holds two mathematical properties: additivity and homogeneity (illustrated in FIG 1 and 2, respectively). If the system does not follow either one of these properties, then it is considered to be nonlinear.

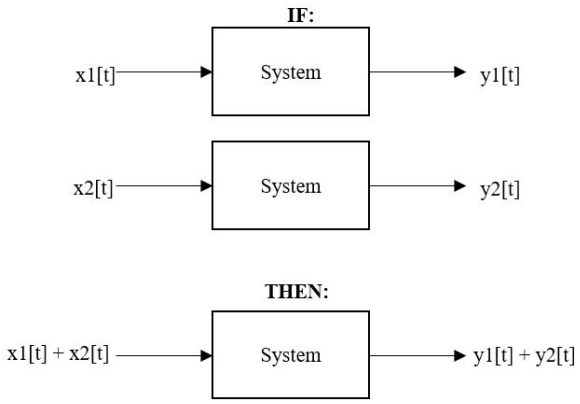

Figure 1: This diagram is illustrating the additivity property of the superposition principle. In other words, if input $x1[t]$ produces output $y1[t]$ and input $x2[t]$ produces output $y2[t]$ (both going through the same system), then $x1[t] + x2[t]$ will produce $y1[t] + y2[t]$.

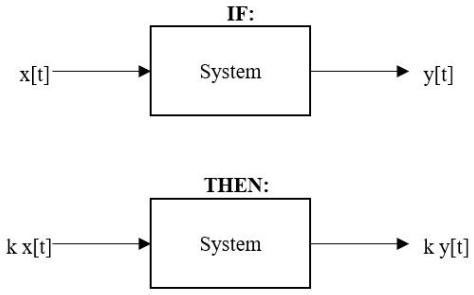

Figure 2: This diagram is illustrating the homogeneity property of the superposition principle. In other words, if k is some constant, than the input $kx[t]$ will produce output $ky[t]$.

It is important to also understand the distinction between the characterization of stationary and non-stationary processes as well. The frequency and statistical contents (i.e. mean, standard deviation, auto- correlation, or power spectral density) in *stationary* processes do not change over a period of time. In other words, the time series generated can change over time, but the properties should remain static. *Non-stationary* processes, on the other hand, does vary over a period of time.

4.2 Linear and Time Invariant System

If the behavior of the system's inputs and outputs does not change due to time, then the system is considered to be time-invariant. A linear time invariant (LTI) system is valid if the previous statement holds true as well as the requirements for a linear system. The LTI model also introduces another property called the shift invariance, which is illustrated in Fig.

3. This property confirms that the behavior of the time-invariant system does not change when time is shifted.

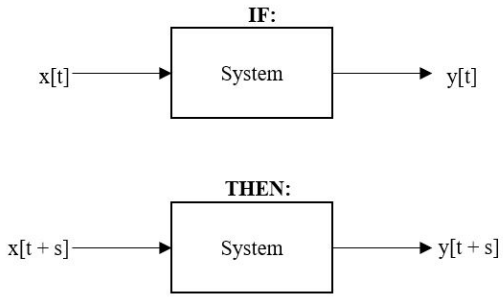

Figure 3: This diagram is illustrating the shift invariance property of a system. If the input $x[t]$ is shift by some constant s, then the output $y[t]$ should also be shift by s. The shift invariance property only holds true if the system is an LTI system.

LTI systems can be described by their impulse response. Below shows another system diagram which is specified to be an LTI system:

$$\delta(t) \longrightarrow LTIsystem \longrightarrow h(t) \tag{2}$$

where Dirac Delta function $\delta(t)$ is the impulse and $h(t)$ is the overall output response to the impulse, which are shifted and scaled. In a physical system, the impulse can be some added physical object that changes the position of the system (i.e., a box sliding across the floor after being hit once by a hammer — the hammer is the impulse in this case).

Recalling the shift invariance property, the input $\delta(t)$ becomes $\delta(t-a)$ while the output is also shifted by $h(t-a)$, where a is some constant number. The assumption being made here is that the system is causal. A system whose present response depends on present and past values of the inputs is called a causal system, while a non-causal system depends on future inputs.

Now, using all three properties (additivity, homogeneity, and shift invariance), some output signal $y(t)$ in an LTI system can be represented in its superposition form as follows:

$$y(t) = k_1x(t-t_1) + k_2x(t-t_2) + ... + k_ix(t-t_i) \tag{3}$$

where the scaling constants are $(k_1, k_2, ..., k_i)$. The same equation above can be written as:

$$y(t) = \sum_{i=0}^{N} k_ix(t-t_i) \tag{4}$$

which in turn can be written in an integral form:

$$y(t) = \int_0^\infty k(\tau)\delta(t-\tau)d\tau \tag{5}$$

where $\delta(t-\tau)$ is the input shown by the shift invariance property and $k(t)$ is the impulse response. This means that the general form of the output $y(t)$ will look like:

$$y(t) = \int_0^\infty k(\tau)x(t-\tau)d\tau \tag{6}$$

Eqn.6 is known as the convolution integral. The mathematical operation of convolution is basically combining two different signals (the input and the impulse response) to output a third signal (the output). Convolution depicted in this form: $(k * x)(t)$.

4.3 Transfer Function

The Laplace Transform is the transformation of a function from the time domain into the s-domain. It is defined mathematically by:

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}dt \tag{7}$$

Therefore, eqn. 6 can be transformed into the Laplace domain as shown:

$$Y(s) = K(s)H(s) \tag{8}$$

This also expresses the Convolution Theorem, which says that if two functions that are being convoluted are Laplace transformed into the s-domain, then the convolution operation simply becomes multiplication.

The transfer function of the system can be obtained in the following form by assuming that it can be written in terms of rational functions:

$$K(s) = \frac{Y(s)}{H(s)} = \frac{b_0 + b_1 s + b_2 s^2 + ... + b_N s^N}{a_0 + a_1 s + a_2 s^2 + ... + a_M s^M} \tag{9}$$

The roots of the polynomial in the numerator of the transfer function are zeroes, while the roots of the polynomials in the denominator of the transfer function are called poles.

4.4 Non-Stationary Noise Model

Below is the equation that describes the total strain $h(t)$ with both the linearly and non-linearly correlated parts of the noise that goes into the detector:

$$h(t) = h_{GW}(t) + \epsilon_L + \epsilon_{NL} + \epsilon_F \tag{10}$$

where ϵ_L is the linear noise coupling, ϵ_{NL} is the non-linear noise coupling, ϵ_F is the fundamental noise that is neither linearly nor non-linearly coupled, and $h_{GW}(t)$ is the actual GW signal. [10].

The linear coupling $\epsilon_L(t)$ can be similarly described in the convolution integral form shown in eqn. 6 as:

$$\epsilon_L(t) = H[s(t)] = \int_0^\infty h(\tau)s(t-\tau)d\tau \tag{11}$$

while the non-linear case is shown as the non-stationary coupling $\epsilon_{NL}(t)$ described by the algorithm is:

$$\epsilon_{NL}(t) = \sum_{i=1}^{N} \alpha_i[x_i(t)s(t)] = \sum_{i=1}^{N} \int \alpha_i(\tau)n_i(t-\tau)d\tau \tag{12}$$

where the H is the linear coupling, the α_i is the non-linear coupling, the $x_i(t)$ is the slow modulation witness channel, $s(t)$ are the fast modulation witness channel, and n_i is the modulated signal of the combined fast noise witness signals and the modulation witness signals. These signals are then coupled into the non-stationary transfer function α_i. This is *not* the most general form of a non-linear case – however, for this algorithm's purpose, the Eqn.12 is describing how each of the modulated signals $x_i(t)s(t)$ are coupling with the non-stationary transfer function α_i.

As shown in Eq.10, the non-stationary correlated part requires two different sets of auxiliary channels for filtering. These two auxiliary channels are the fast noise witness channels and the slow modulation witness channels, each containing different spectral content at different frequency bands. The fast noise witnesses [8] are the channels that "witness" the faster noise, while the slow modulation witnesses [8] are the channels that "witness" the modulation of the noise couplings. For this project, the fast noise contain content in the 1 to 10 Hz frequency band while the slow noise contain content below 1 Hz.

As shown in Eq.10, the assumption can be made that some of the noise that is witnessed by an auxiliary channel is coupled to the strain through a linear and stationary coupling, H. In this case, the step is to find a fast noise witness channel that have power in the frequency

band of interest (i.e., the frequency range desired for the noise subtraction) by calculating the linear coherence between different witness channels and the target channel. If there is coherence between the channels in the frequency range of interest, then linear and stationary subtraction can be implemented. However, in a realistic scenario, most of the noise coupling is changing over time, (i.e., non-stationary coupling α_i). This is where it becomes easier to make the distinction between the fast witness and the modulation witness signals. In particular, "modulation witness" signals that have time variations that follow the changing coupling parameters. This is shown in Eq.10, where the modulation witnesses are multiplied by the fast noise signals, thus producing a time-varying gain of the filters. Therefore, if a modulation signal follows the way the non-stationary coupling on the signal changes over time, then it is possible to predict the model for subtraction.

As a starting point, the SUS (suspension), ISI (internal seismic isolation in vacuum chamber), ASC, and SEI (seismic) auxiliary channels are good possible candidates for the fast noise witness since they are channels relating to seismic motion and control systems, which can contain useful signals for the lower frequency regime. ASC error signals was suggested to use for the slow modulation witness. Originally, some of the ASC signals have also been used as fast noise witnesses to successfully subtract noise between 10 and 30 Hz as shown in Fig. 4.4, which makes it a decent starting point. [8]

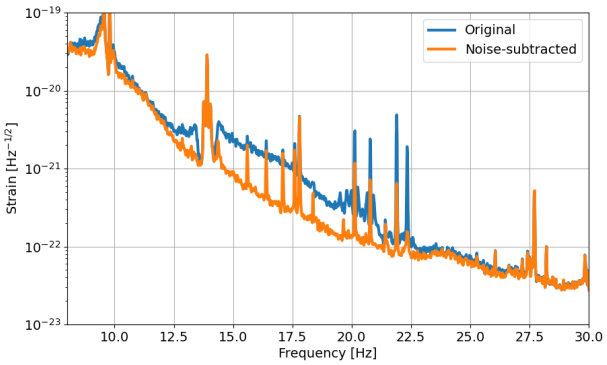

Figure 4: Graph of the ASC subtraction done between 10 to 30 Hz. Taken from logbook: https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=57423

The target strain channel $h(t)$ is the channel that the linear and non-linear noise couplings would be subtracted from. For this project, the CAL-DELTAL_EXTERNAL_DQ channel is the best candidate for the "target" channel. This channel is derived from control signals that are then modified to produce the calibrated strain signal that is correct below 10 Hz, which is the frequency of interest here. The downside to using this calibrated signal is that is less accurate as oppose to the GDS-CALIB_STRAIN channel, which is the main product of the calibration pipeline and is generally used for all data analysis. However, the GDS-CALIB_STRAIN channel is only useful for subtraction above 10 Hz since the strain is not corrected below that range. At this time, the CAL-DELTAL_EXTERNAL_DQ channel is the only calibrated strain that is available below 10 Hz.

4.5 Second Order Stages

The non-linear coupling is given by eqn. 12. This reflects eqn. 6, and therefore it can be inferred that α_i is the kernel. The relationship between the kernel and the transfer function was shown in eqn. 9. Transforming α_i back into the s-domain as $\alpha_i(s)$, the transfer function can be expanded into:

$$\alpha_k(s) = \frac{b_0 + b_1 s + b_2 s^2 + ... + b_N s^N}{a_0 + a_1 s + a_2 s^2 + ... + a_M s^M} = \frac{\sum_{j=0}^{N} b_j s^j}{\sum_{j=0}^{M} a_j s^j} \tag{13}$$

where $(M > N)$ since this is a casual system. Here, the j roots of b are the zeroes of $\alpha_i(s)$ and the j roots of a are the poles of $\alpha_i(s)$. Therefore we can write this as:

$$\alpha_i(s) = \frac{b_j}{a_j} \frac{(s - z_1)(s - z_2)....(s - z_M)}{(s - p_1)(s - p_2)....(s - p_N)} \tag{14}$$

where $z_1, z_2, ..., z_M$ are the zeroes (roots of b), $p_1, p_2, ..., p_N$ are the poles (roots of a), and $\frac{b_j}{a_j}$ represents the scalar gain. Then, by decomposing it:

$$\alpha_i(s) = c + \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + ... + \frac{r_N}{s - p_N}$$

$$\alpha_i(s) = c + \sum_{j=1}^{M} \frac{r_j}{s - p_j} \tag{15}$$

where r_i is the complex residual and p_i is the complex pole. In order to make the time-domain response of the transfer function real, there are two possible conditions: a) r_i and p_i must have conjugate pairs and b) r_i and p_i be real. Therefore, those two conditions are shown in the equation below:

$$\alpha(s) = c + \sum_i \left[\frac{r_i}{s - p_i} + \frac{r_i^*}{s - p_i^*} \right] + \sum_j \frac{r_j}{s - p_j} \tag{16}$$

where the second term denotes the complex term and the third term is the real term. The following set of equations is simply showing how the complex term gets expanded out:

$$\alpha_{complex}(s) = \sum_i \left[\frac{r_i(s - p_i^*) + r_i^*(s - p_i)}{(s - p_i)(s - p_i)^*} \right]$$

$$\alpha_{complex}(s) = \sum_i \left[\frac{r_i(s - p_i^*) + r_i^*(s - p_i)}{(s^2 - 2[p_i s] + p_i^2)} \right] \tag{17}$$

and then for the real term, pair up the real poles:

$$\alpha_{real}(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + ... + \frac{r_j}{s - p_j}$$

$$\alpha_{real}(s) = \frac{r_1(s - p_2) + r_2(s - p_1)}{(s - p_1)(s - p_2)}$$

$$\alpha_{real}(s) = \frac{(r_1 + r_2)s + r_1 p_2 - r_2 p_1}{s^2 - s(p_2 - p_1) + p_1 p_2} \tag{18}$$

Eqns 17 and 18 both contain a second order polynomial in their denominator, which is called the second order stage.

4.6 NonSENS Code

The NonSENS (NON-Stationary Estimation of Noise Subtraction) algorithm is written in Python scripts. The main "nonsens" interface consists of several scripts which contain the codes to perform each step of the subtraction. Fig 6 and 7 show the display of the algorithm's output on an integrated terminal. The iPython command shell is utilized as the interpreter on this terminal prompt since the algorithm is written in Python.

The purpose of utilizing the NonSENS algorithm is to find the optimal parameters that will reduce the maximum amount of noise from the target strain as possible. In order to implement a linear and stationary subtraction using the algorithm, only the noise witnesses should be considered (meaning that the list of modulation witnesses should remain empty). However, the non-stationary noise couplings ought to be considered if one wishes to perform the most optimal noise subtraction, which is what the algorithm takes into account. As shown in eqn.10, the $x_i(t)$ (slow modulation noise) and $s(t)$ (fast noise) are multiplied together. Each of the modulated signal is then coupled with the non-stationary α_i transfer function, which results in having all of those transfer functions summed together.

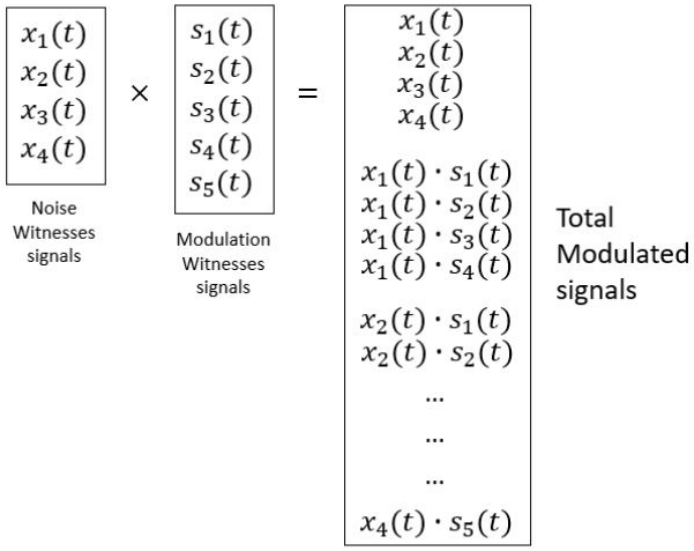

Figure 5: This diagram is an illustration example of how the witness signals are combined in the algorithm in order to form the modulated signals. Essentially, the algorithm takes each noise witness $x_i(t)$ and multiplies it with each modulation witness $s_j(t)$. The total modulated signals in the end are equal to $i + (j \times i)$, where i and j are simply the indices of the witness signals. Each of the modulated signals will then go through a transfer function α_i before being summed together.

The algorithm then implements a filter: the time-domain IIR (infinite impulse response) filter, which as the name implies has an impulse response that is infinite. This is used as opposed to the FIR (finite impulse response). The advantage of utilizing the IIR filter is having a faster processing time and having to use fewer parameters than with the FIR when having to compute a similar filtering operation. The IIR filter also has a recursion feedback aspect which the FIR filter does not have. The FIR filter output can be described in the following equation:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{19}$$

where the $y[n]$ is the output of the filter, $x[n-k]$ is the past input delay, and b_k is is the filter coefficient. The IIR filter output is shown as:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k] \tag{20}$$

where the first term in the equation which represents the sum of past inputs like in Eqn.19 and the second term is the sum of the past outputs with a different filter coefficient a_k.

However, the downside of the IIR filter is the higher complexity of the design when it comes to trying to solve optimization problems. This is due to having the unknown filter coefficient

a_k as shown in Eqn.20. The overarching problem now is that algorithm might not reach an optimal solution every single time. (See Section 4.6.1 regarding the correlation between optimal value and cost function.)

Finally, the sum of all of the filtered signals are then to be subtracted from the original target signal; hence the subtraction is performed!

```
In [13]: %run -i modified_sub_asc.py
Reading target channel (GPS 1242441180 - 1242442380) H1:CAL-DELTAL_EXTERNAL_DQ
    using gwpy
Reading noise witness channels (GPS 1242441180 - 1242442380)
        H1:ASC-DSOFT_P_OUT_DQ
        H1:ASC-DSOFT_P_SM_DQ
        H1:ASC-DSOFT_P_IN1_DQ
        H1:ASC-DHARD_P_SM_DQ
        H1:ASC-DHARD_P_OUT_DQ
        H1:ASC-DHARD_P_IN1_DQ
        H1:ASC-X_TR_B_PIT_OUT_DQ
        H1:ASC-X_TR_B_YAW_OUT_DQ
        H1:ASC-X_TR_A_PIT_OUT_DQ
        H1:ASC-X_TR_A_YAW_OUT_DQ
        H1:SUS-SRM_M3_MASTER_OUT_LL_DQ
        H1:SUS-SRM_M3_MASTER_OUT_UL_DQ
        H1:SUS-SRM_M3_MASTER_OUT_UR_DQ
    using gwpy
Reading modulation witness channels (GPS 1242441180 - 1242442380)

    using gwpy
Preprocessing...
    Normalizing signals to zero mean and unity std
    Resampling all witness signals to 512 Hz
    Build modulated signals
    Applying preconditioning filter
Computing cross spectral density matrices...
    Detecting glitchy segments
        119 good segments / 0 bad segments
    Computing FFTs
    Averaging FFTs to get CSDs
Training model...
step = 0        cost = 1.033274762628
step = 100      cost = 1.905557540165
step = 200      cost = 1.137036420796
step = 300      cost = 0.944441884869
step = 400      cost = 0.965317402949
```

Figure 6: This shows the first half of the algorithm's output on the iPython interpreter. At the very top of the picture is the "%run -i" which is the command to run a python script.

```
step = 9400     cost = 0.805637329619
step = 9500     cost = 0.788132887089
step = 9600     cost = 0.786126953243
step = 9700     cost = 0.784972109532
step = 9800     cost = 0.784240923142
step = 9900     cost = 0.783878695499
Preprocessing...
    Normalizing signals to zero mean and unity std
    Using pre-computed normalizations
    Resampling all witness signals to 512 Hz
    Build modulated signals
    Applying preconditioning filter
Time domain subtraction (method = serial)
    applying upsampling filter...
    applying antialias bandstop filter...
    Saving to file /home/yuka.lin/test_codes/nonsens-master/examples/Linear/NonS
ENS_test/plots//asc_model1_subtracted_timedomain_H_1242441180_1200_2021_07_24_17
h30m53s.png
```

Figure 7: This is the second half of the algorithm's output, continuation from Fig. 6.

In order to perform a subtraction utilizing the algorithm, a Python script that is run on the terminal which contains all of the user's inputted parameters for the particular subtraction. The following parameters are utilized and adjusted as necessary:

Target channel: is the main strain channel. It is up to the user to input the specified channel's information regarding GPS time, interferometer site, and how the channel would be read into the code.

Noise Witness Channels: are the fast noise witness channels that will be used to subtract with the strain.

Modulation Witness Channels: are the slow noise witness channels that will be used to subtract with the strain. This list should be kept empty in order to perform only a linear subtraction. This was shown in the second term of Eqn. 10, where $H[s(t)]$ is the stationary noise coupled with the fast noise witness channel. (However, as a note, the equation model only considers one fast noise witness channel while the algorithm can compute multiple channels at a time.)

LIGO auxiliary channels are utilized as the channel parameters. These channels monitor the physical behavior of LIGO detectors and are collected as data in the form of a time series.[4]

Preconditioning Filter: applies the preconditioning filter to the modulated signals. For the ASC subtraction, the analog band-pass butter-worth filter was used, which is a variation of the IIR filter. The term "band-pass" is a type of filter that sets a frequency specific range for the filter to pass; anything outside of that range won't be able to pass.

fs: is the sampling frequency, which is the amount of samples in each second (1/time).

n_FFT: is the segment size of the FFT's (Fast Fourier Transform). The FFT essentially computes the DFT (Discrete Fourier Transform) in a more efficient way using an algorithm.

Glitches: are short-duration noise transient signals that can appear as an instrumental artifact in the detectors. [4] The frequency band range and the threshold parameters can be adjusted to detect possible the glitches. If glitches are found in any segments, those will be discarded from the final calculation.

fband: frequency range wanted to minimize the residual noise within that band range.

nSOS: the number of second order stages specified by user. This was explained in the previous section.

The following parameters are specified for the Adam (Adaptive Moment Estimation) algorithm [11] in order to minimize the cost function (see Fig.8): *learning_rate*, *decay_steps*, *decay_rate*, and *nsteps*.

The algorithm outputs the plots shown in the following sections.

4.6.1 Cost Function Plot

A cost function is used to estimate how poorly the model will perform based on the relationship between the independant and dependant variables of a model (i.e. the smaller the cost function is, the better the estimate and vice-versa). Using this logic, it can be concluded for this algorithm the following conditions: 1) if the minimum of the cost function is equal to 1, no noise subtraction is happening, 2) if the minimum of the cost function is greater than 1, noise is being added to the target, 3) if the minimum of the cost function is less than 1, noise is subtracted from the target. In this case, the cost function is the calculated average of the ratio of the subtracted power spectral densities divided by the original power spectral density in that frequency range.

The cost function calculated in the algorithm is defined as the following equation:

$$C(\theta) = \int_{\omega_L}^{\omega_H} \frac{S[r,r](\omega)}{S[h,h](\omega)} d\omega \tag{21}$$

where the $[\omega_L, \omega_H]$ are the selected frequency band range of interest in Hz, $S[r,r]$ is the PSD of the subtracted target signal, and $S[h,h]$ is the PSD of the original target signal.

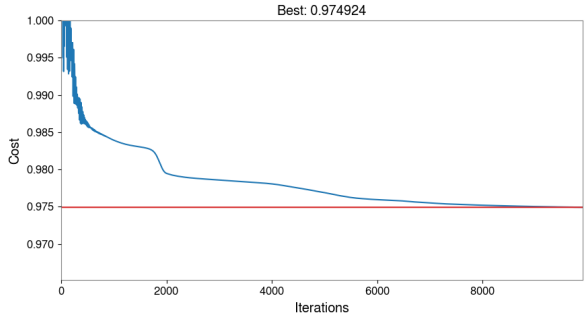

Figure 8: The minimum of the cost function is indicated by the red line. The exact value of this minimum is indicated by the plot title "Best: 0.974924" for this particular model. The optimal value that this algorithm outputs is the minimum value that is calculated by the cost function.

The residual $r(t)$ is the leftover after the subtraction. Ultimately, the goal of the nonlinear subtraction is to minimize the residual and minimize the cost function value – the more optimized the subtraction, the lower cost function and the smaller the residual value should be. This model also helps to solidify that the algorithm's nonlinear subtraction will never perform worse than the linear subtraction.

The first assumption begins with only the linear case:

$$r(t) = h(t) - \epsilon_L(t) \tag{22}$$

where $\epsilon_L(t)$ is the optimal linear subtraction that is performed. Therefore, the value of the cost $C(\theta_L)$ should be optimal. Now, the next case shows the non-linear subtraction being implemented.

$$r(t) = h(t) - \epsilon_L(t) - \epsilon_{NL}(t) \tag{23}$$

Now there are two scenarios that can be made: the first one is that the cost $C(\theta_{NL})$ is equal to the $C(\theta_L)$ because there was no non-linear contribution to the subtraction; therefore, the only subtraction that was done in the entire model was the optimized linear part. The second scenario is that $C(\theta_{NL}) < C(\theta_L)$ since the modulated coupling had some contribution, which is then subtracted with the linear stationary coupling.

4.6.2 Subtraction Plot

The amplitude spectral density (ASD) is calculated as the square root of the power spectral density (PSD). The PSD shows an estimation of how the power is distributed as a function of frequency. This estimation is calculated using Welch's method. [12] Welch's method essentially breaks a signal into segments called "time windows" in order to take the FFT of each piece and then averaging them all together to create the PSD. This tends to help with calculations of stationary-like signals and make the PSD's smoother as oppose to simply just taking a single FFT of a full signal.

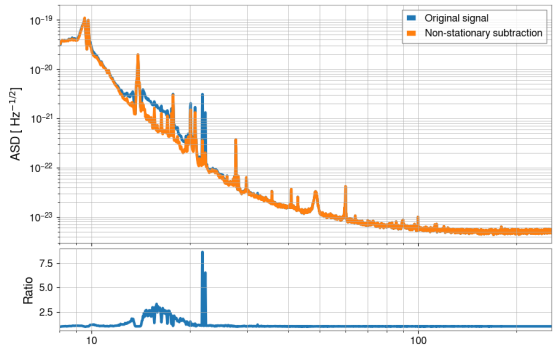

Figure 9: This plot depicts the ASD of the original target signal (which in this case is the "DCS-CALIB_STRAIN_CLEAN_SUB60HZ_C01" strain) and the subtracted version of that same target signal in order to show the amount of modification that was done in this subtraction. The ratio plot at the bottom is graphed as the original target signal divided by the subtracted target signal — therefore, if the ratio is shown to be above one as is depicted in this example, then it is further indication that a subtraction is being performed as opposed to an addition of noise. In the example plot above, it is shown that the noise was reduced by a factor of a little over 2.5 between the 15-18 Hz frequency range.

4.6.3 Contribution Plot

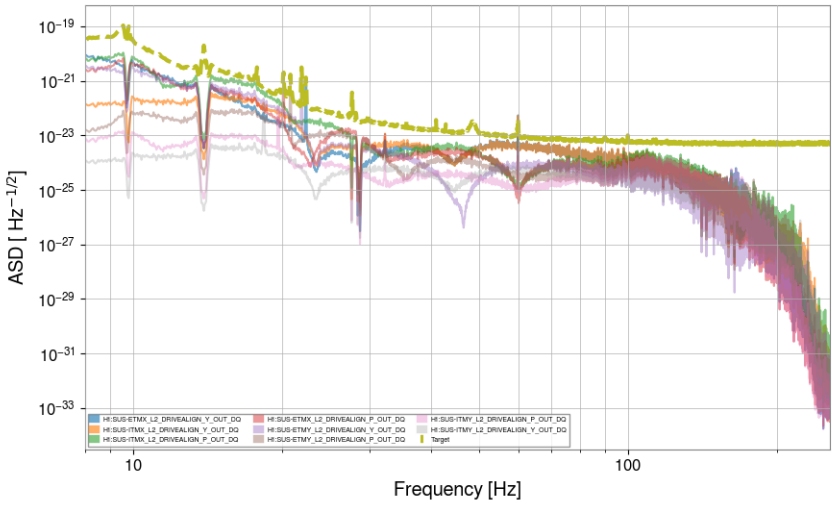

Figure 10: This plot shows the amplitude spectral density (ASD) of the original target channel and the noise witness channels. This helps to compare whether or not the target and the witness channels have similar power content at the frequency ranges of interest.

4.6.4 Alpha Plot

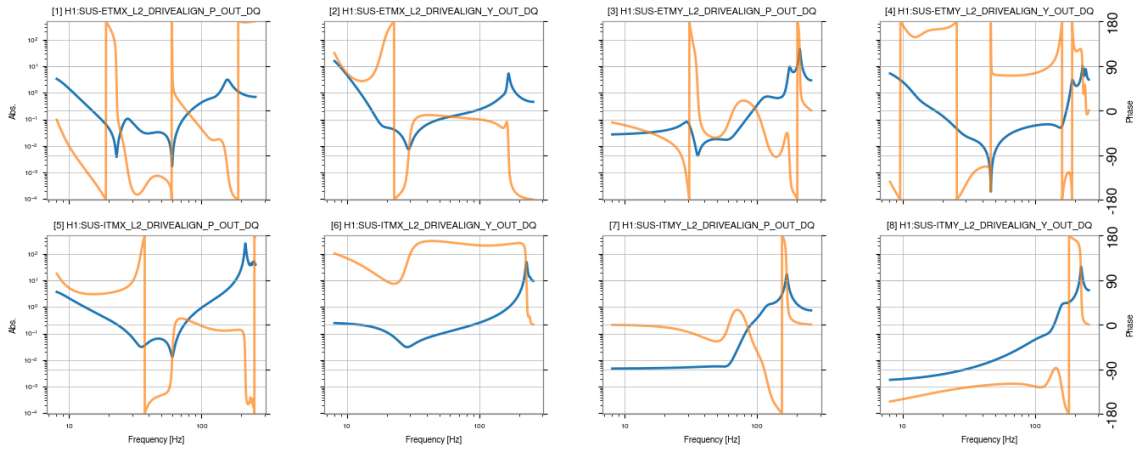

Figure 11: The plots shown are the transfer functions for each of the modulated signals. The orange lines are the phase values and the blue lines are the magnitude values of the transfer functions. In this case, there are eight plots since only eight noise witnesses and no modulation witnesses were used for this run. See Fig. 5 for how the modulated signals are combined from the noise and modulation witness signals.

4.6.5 Time-Domain Plot

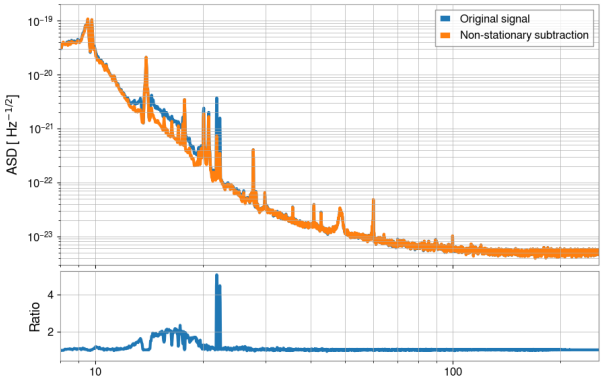

Figure 12: This is the same plot as Fig. 9, but showing the subtraction with the time-domain PSD signals. The results from algorithm was converted from the computed frequency domain transfer function into the time-domain IIR filter. [1] The resulting time-domain PSD is what is plotted in the figure.

4.6.6 Spectrogram Plots

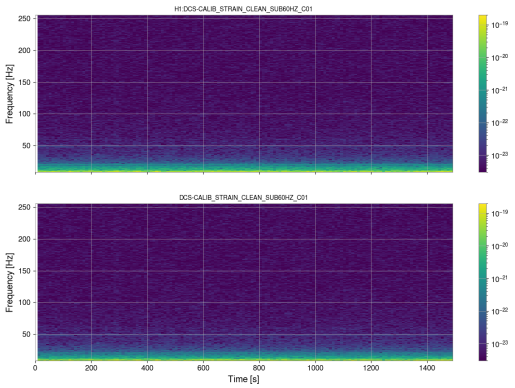

Figure 13: A spectrogram shows an estimate of the PSD on shorter intervals on a frequency vs time scale in order to see how the PSD changes with time.

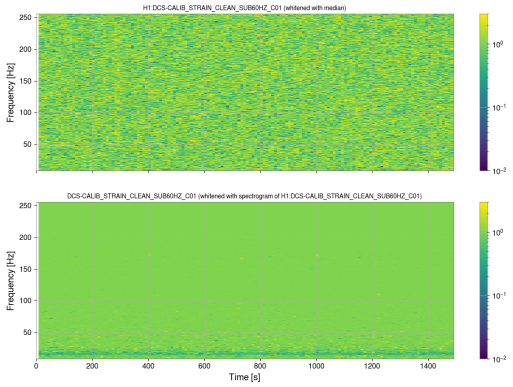

Figure 14: The same spectrogram as Fig.13 but whitened. Whitening is normalizing the data at each frequency to make it appear more uniform in order to be able to check for any excess power at certain points in the frequency ranges. As shown in this figure, the power seems evenly distributed.

A more in-depth summary of the algorithm code can be found at this website: `https://wiki.ligo.org/CSWG/Algorithm`.

The algorithm Python code can be found here: `https://git.ligo.org/gabriele-vajente/nonsens/-/tree/master`.[8]

4.7 Approach

The first step in the project would be to subtract as much noise from the "target" strain that is linearly correlated with the fast noise witness auxiliary channels. Since there are a number of active LIGO channels to choose from, the practical strategy would be to calculate whether any linear coherence exists in the lower frequency regime between the channels in order to establish if linear subtraction would be viable. If no coherence exists between channels, than there is no linear subtraction that can be performed.

The next step then would be to remove the part of the noise that is non-linearly correlated. However, finding the linear coherence between the channels is not a set determination of whether or not a non-linear subtraction would be possible. At this time, though, it would serve as a starting point to select certain set of channels that might have the possibility of subtracting the non-linear noise from.

For more information and details regarding this project, please see this link which documents more information about it: `https://wiki.ligo.org/CSWG/NonLinearNoiseSub_LF`.

Specific daily logs are kept at this link: `https://wiki.ligo.org/CSWG/WeeklySummaries`.

5 Progress

5.1 NonSENS Algorithm

The initial weeks prior to the project was dedicated to understanding how to utilize the NonSENS algorithm code. The existing example code for the subtraction of the ASC arms was used as the starting template for this project.[8] This is a good starting point because some of the ASC signals were previously used as a fast witness to subtract noise between 10 and 30 Hz, which is the lowest range that the subtraction algorithm had performed up to date.

5.2 Spectrograms

Another important focus in the first week was also producing the spectrogram of the H1:CAL-DELTAL_EXTERNAL_DQ strain. The purpose of making the spectrograms was to visually observe if there were stationary or non-stationary behavior in each of the spectrum bands that are contained within the strain. If there are some prominent stationary behavior found, then next approach would be to find channels that are linearly coherent at the general frequency range of this spectrum. Linear coherence describes the correlation between the two stationary signals in the frequency domain. This was the first step in being able to narrow down specific auxiliary channels to utilize that will be able to perform a noise subtraction for this lower frequency range.

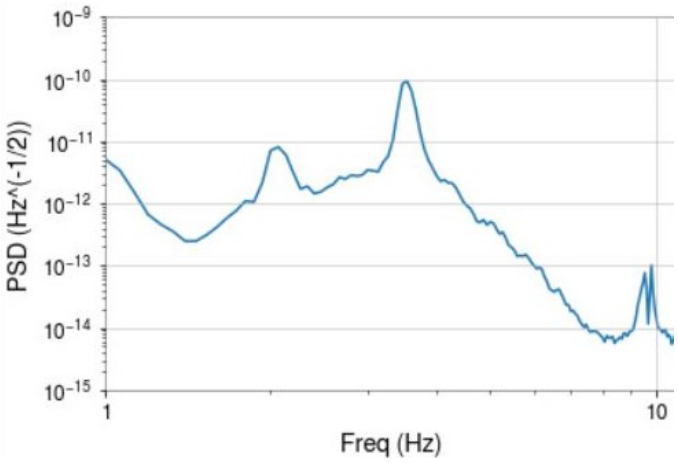

Figure 15: This plot shows the power spectral density (PSD) versus the frequency of the H1:CAL-DELTAL_EXTERNAL_DQ strain signal. As shown in this plot, the PSD peaks occur at the close to the same frequencies as each of peaks in FIG. 16 below.

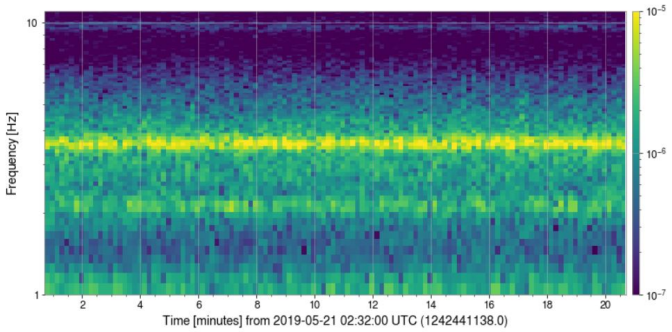

Figure 16: In observing the spectrogram of the H1:CAL-DELTAL_EXTERNAL_DQ strain produced, there appears to be prominent peaks near in the $\sim 2.1 - 2.4$ Hz, $\sim 3.5 - 3.8$ Hz, and $\sim 9.5 - 9.9$ Hz frequency ranges. The following figures display the same spectrograms that are zoomed in on these particular frequency ranges as shown in the graphs below these graphs which can be done by adjusting the FFT lengths (increase gets better frequency resolution) and the stride (increase gets better time resolution).

5.2.1 Spectrogram 2.1 - 2.4 Hz

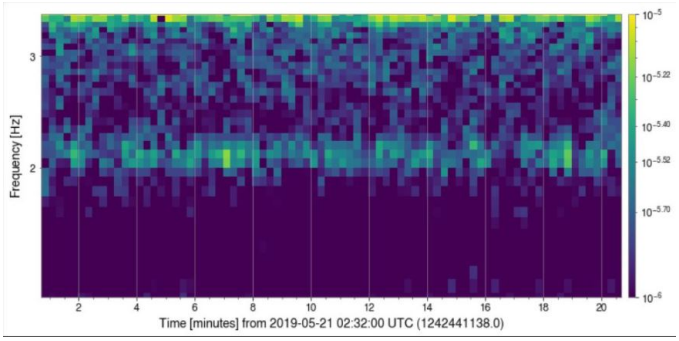

Figure 17: This part of the spectrogram shows that there is a peak between 2.1-2.4 Hz. This particular peak is displaying a lot of non-stationary behavior, though there is some stationary as well.

5.2.2 Spectrogram 3.5 - 3.8 Hz

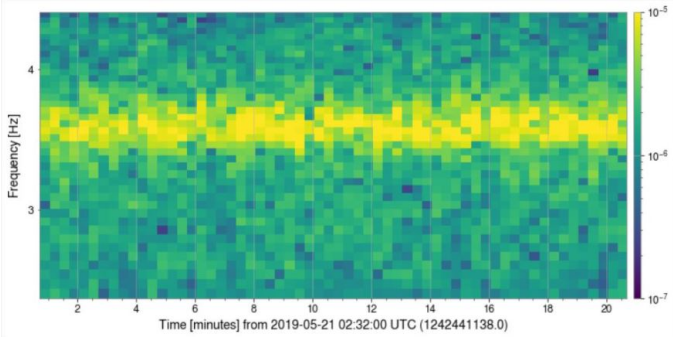

Figure 18: The most prominent spectral feature appears on this plot between 3.5-3.8 Hz. There appears to be a mostly stationary behavior on this spectrum peak.

5.2.3 Spectrogram 9.5 - 9.9 Hz

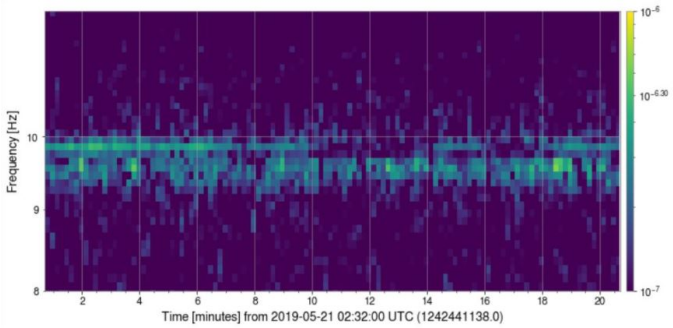

Figure 19: This peak appears in the 9.5 - 9.9 range on the spectrogram. The non-stationary behavior is more apparent than the other peaks.

5.3 Coherence

Linear coherence describes the correlation between the two signals in the frequency domain. The coherence between two signals is calculated from a scale from 0 to 1. The closer the coherence value is to 1, the higher the coherence between the two signals.

If a target signal and noise witness signals have any linear coherence with each other, the linear subtraction on the target channel should eliminate the coherence between the subtracted target and the noise channels.

Frequencies: (Hz)	2.06	2.09	2.12	2.16	2.19	2.22	2.25	2.28	2.31	2.34	2.38	2.41
PEM-EX_ADC_0_19_OUT_DQ	0.65	0.49	0.44	0.43	0.51	0.40	-	0.22	-	0.32	0.32	
ASC-X_TR_A_YAW_OUT_DQ	0.31	0.30	0.33	0.25	-	0.22	0.24	0.23	0.28	-	-	-
ASC-Y_TR_B_NSUM_OUT_DQ	0.21	0.30	-	-	0.27	0.25	0.28	0.29	0.25	-	-	-
ASC-DSOFT_P_OUT_DQ	-	-	-	0.26	0.30	0.26	0.24	0.20	0.27	0.24	-	-
ASC-DSOFT_P_SM_DQ	-	-	-	0.26	0.30	0.26	0.24	0.20	0.27	0.24	-	-
ASC-DHARD_P_OUT_DQ	-	-	0.28	0.33	0.36	0.25	-	-	-	0.25	0.42	0.50
ASC-DHARD_P_SM_DQ	-	-	0.28	0.33	0.36	0.25	-	-	-	0.25	0.42	0.50
SUS-ETMX_R0_DAMP_Y_IN1_DQ	0.30	0.31	0.31	0.25	0.24	0.31	0.25	0.19	0.27	-	-	-
SUS-ITMX_M0_DAMP_Y_IN1_DQ	-	0.23	0.26	0.25	-	0.33	0.32	0.26	0.40	0.30	-	-
SUS-ETMX_M0_DAMP_Y_IN1_DQ	0.28	0.26	-	-	0.22	0.30	0.28	0.27	0.37	0.24	-	-

Table 1: Top channels with the best coherence with H1:CAL-DELTAL_EXTERNAL_DQ at 2.0-2.4 Hz. Each of the coherence was calculated for each of the channels at different frequencies.

Frequencies: (Hz)	3.50	3.53	3.56	3.59	3.62	3.66	3.69	3.72	3.75	3.78	3.81
LSC-SRCL_IN1_DQ	0.65	0.65	0.57	0.61	0.59	0.50	0.35	0.27	0.40	0.33	0.32
SUS-SRM_M1_NOISEMON_RT_OUT_DQ	0.65	0.66	0.58	0.61	0.58	0.50	0.34	-	0.40	0.32	0.31
SUS-SRM_M1_NOISEMON_LF_OUT_DQ	-	0.66	0.59	0.61	-	0.50	0-	0.27	0.40	0.33	0.30
SUS-SRM_M3_MASTER_OUT_LL_DQ	0.66	0.68	0.61	0.63	0.60	0.49	0.34	0.27	-	-	-
SUS-SRM_M3_NOISEMON_LL_OUT_DQ	0.66	0.68	0.61	0.63	0.60	0.49	-	0.27	-	-	-

Table 2: Top channels with the best coherence with H1:CAL-DELTAL_EXTERNAL_DQ at 3.5-3.8 Hz.

5.4 Picking Channels

In order to aid even further in narrowing down a list of possible auxiliary channels to use as a fast noise witness, an algorithm called Bruco (Brute Force Coherence) [13] was utilized to create a list of specific channels that have some linear coherence with the H1:CAL-DELTAL_EXTERNAL_DQ strain. More specifically, this algorithm is able to output the top twenty channels that have the highest coherence with the strain relevant at each frequency. The following link is the most updated list that was generated by Dr. Gabriele Vajente to utilize for this project: `https://ldas-jobs.ligo.caltech.edu/~gabriele.vajente/bruco_lf_2021_06_22/`.

With the list of top linear coherences, it was easier to pick the best coherent channels in a systematic manner. Each of the channels that were picked at the corresponding frequency ranges listed in Tables I and II had the most relevance to the overall range instead of just at an individual frequency according to the generated list.

However, it was discovered that while certain auxiliary channels appeared to have coherence with the target, they were not a direct source of the noise coupling. For example, certain channels that are used for control systems purposes do not contain GW signals at all. Other channels are repeated channels such as the ASC-DSOFT_P_OUT_DQ and ASC-DSOFT_P_SM_DQ channels, which means that one of the channels should be omitted when doing the subtraction.

5.5 Linear Subtraction

A series of linear subtractions was performed based on which channels had the best coherence with the target channel. The channels that was shown to have the largest subtraction were combined into a single list of noise witnesses and a full subtraction was performed for that.

Best Noise Witness Channels:
ASC-DSOFT_P_IN1_DQ
ASC-DHARD_P_IN1_DQ
ASC-X_TR_B_PIT_OUT_DQ
ASC-X_TR_B_YAW_OUT_DQ
ASC-X_TR_A_PIT_OUT_DQ
ASC-X_TR_A_YAW_OUT_DQ
SUS-SRM_M3_ISCINF_L_IN1_DQ
SUS-SRM_M3_ISCINF_P_IN1_DQ
SUS-SRM_M3_ISCINF_Y_IN1_DQ
SUS-SRM_M3_WIT_P_DQ
SUS-SRM_M3_WIT_Y_DQ
SUS-SRM_M3_WIT_L_DQ
SUS-ZM2_M1_VOLTMON_UL_OUT_DQ
SUS-ZM2_M1_VOLTMON_LL_OUT_DQ
SUS-ZM2_M1_VOLTMON_LR_OUT_DQ

Table 3: A list of the best channels for linear subtraction with H1:CAL-DELTAL_EXTERNAL_DQ. These channels were chosen from the list of the top linear coherence. Each channel was then linearly subtracted from the target individually to check for the contribution before being appended to this list.

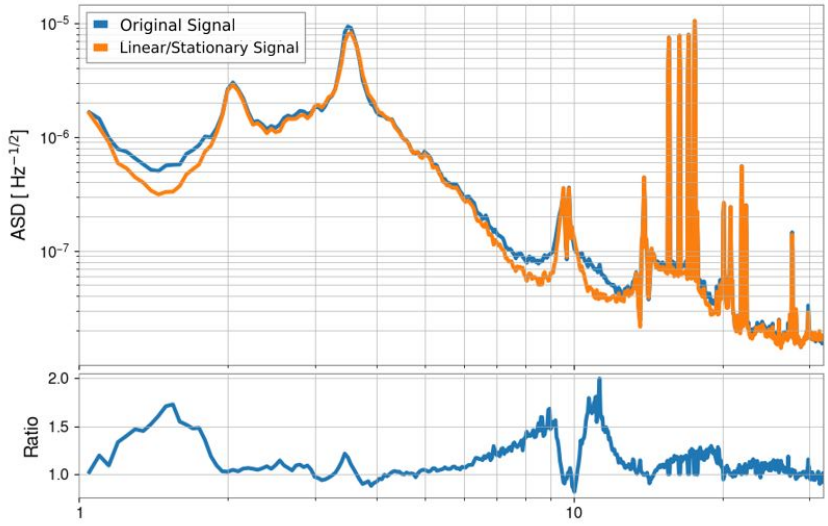

Figure 20: This plot is the best linear and stationary subtraction subtraction, using the list of noise witness channels that are given Table 5.5. The ratio plot on the bottom gives a clearer indication of the amount of noise that is being subtracted (any time the ratio goes above 1.0 indicates subtraction while any time the ratio goes below 1.0 indicates addition instead).

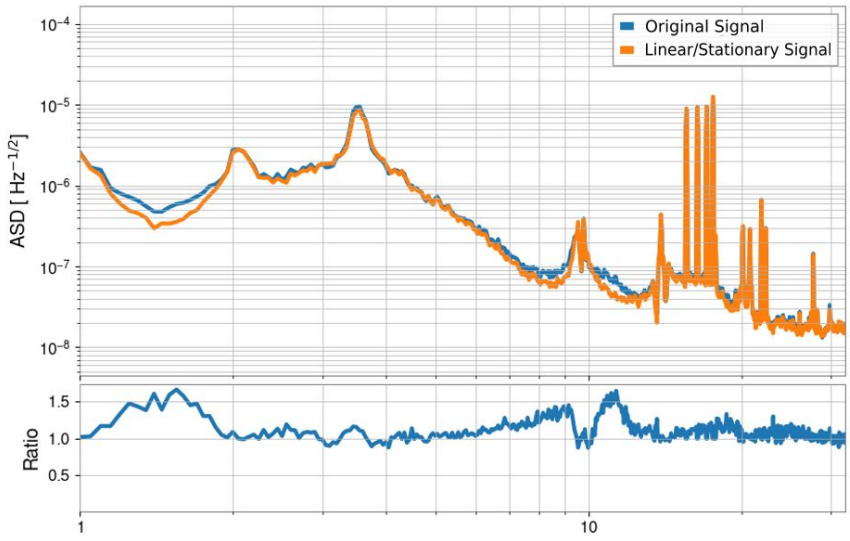

Figure 21: This plot is the time domain version of the plot in Fig. 22.

5.6 Non-Linear Subtraction

In order to perform the non-linear subtraction with the NonSENS algorithm, the same parameters from the linear subtraction can be used along with an added parameter now: the list of modulation witness channels. Choosing a set of modulation channels is harder than select-

ing fast noise channels because there is no systematic way of selecting such auxiliary channels. Therefore, as a starting point, it is logical to pull from previous successful tests which was done with the 10-30 Hz subtraction on the "DCS-CALIB_STRAIN_CLEAN_SUB60HZ_C01" strain, which was shown in Fig. 9.

Modulation Witness Channels:
ASC-INP1_P_INMON
ASC-INP1_Y_INMON
ASC-MICH_P_INMON
ASC-MICH_Y_INMON
ASC-PRC1_P_INMON
ASC-PRC1_Y_INMON
ASC-PRC2_P_INMON
ASC-PRC2_Y_INMON
ASC-SRC1_P_INMON
ASC-SRC1_Y_INMON
ASC-SRC2_P_INMON
ASC-SRC2_Y_INMON
ASC-DHARD_P_INMON
ASC-DHARD_Y_INMON
ASC-CHARD_P_INMON
ASC-CHARD_Y_INMON
ASC-DSOFT_P_INMON
ASC-DSOFT_Y_INMON
ASC-CSOFT_P_INMON
ASC-CSOFT_Y_INMON

Table 4: A list of the modulation channels for non-linear subtraction with H1:CAL-DELTAL_EXTERNAL_DQ. These channels were originally used successfully for subtraction between 10-30 Hz, which was the lowest subtraction performed so far. The ASC signals comes from the part of the detector system that helps to "align" the suspended test masses as a way to counteract seismic disturbances. [14] Noise couplings in the "lower" frequency regime are usually related to seismic motion. This is why for the modulation witness and noise witness auxiliary channels, ASC and SUS channels were considered to be reasonable starting points for this project.

The non-linear subtraction poses another challenge since the algorithm does not always converge for every run. The cost function, as mentioned before, helps to indicate how good or bad the model in each run performs. The following figures show two different trials utilizing the same parameters in both runs to exemplify this.

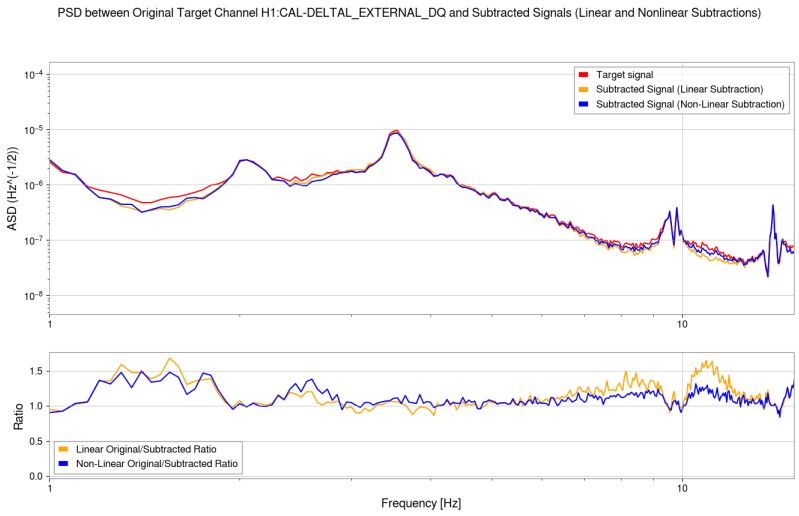

Figure 22: The top portion of this plot shows the subtraction, using the list of noise witness channels and the modulation witness channels that are given in Table 5.5 and 5.6. The linear subtraction, indicated with the orange line, is plotted along side with the non-linear subtraction, indicated by the blue line, in order to show how much improvement in the subtraction there there between the two. Intuitively, the non-linear subtraction should perform either the same or better than the linear subtraction. However, the algorithm does not always guarantee converge for the non-linear subtraction; in other words, the algorithm demonstrates its stochastic nature when trying to obtain the optimal solution. The bottom portion of the plot shows the ratios between each of the subtracted signals to the original signal, which makes it easier to visually see how much the subtractions have improved/worsened. From the range of about $2 - 4$ Hz, there is slight improvement by the non-linear subtraction; however, over the 6 Hz range, the non-linear subtraction actually appears to have gotten worse. This indicates that the algorithm might not have converged well in this run at that frequency range. However, the next plot below will show a different run in which the same range improves instead.

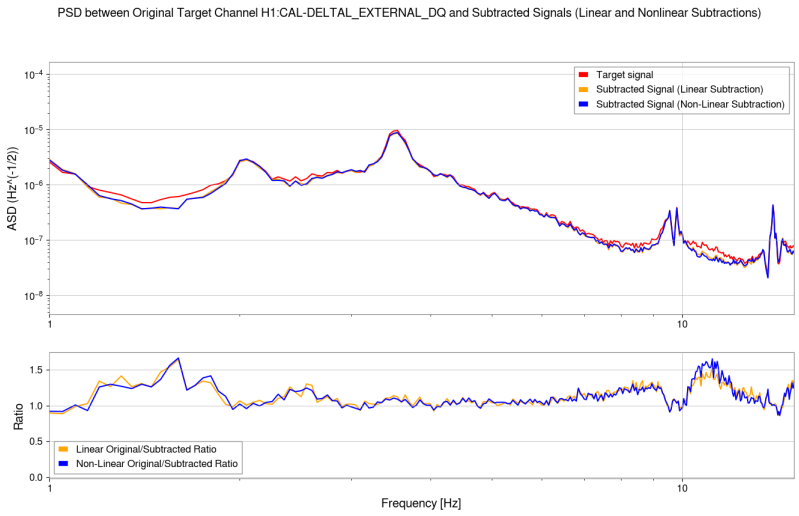

Figure 23: This subtraction was performed with the exact same parameters and channels that was done in the previous plot, Fig. 22. However, there are apparent differences in this run than that last. Between the $2 - 10$ Hz range, the subtractions of the non-linear and linear seem virtually similar - in that there is not really an improvement with the non-linear subtraction. However, there is a slight notable improvement with the non-linear in the $10 - 11$ Hz range, which the previous run had the opposite effect.

6 Conclusion

The objective of this project was to determine how effective a non-linear subtraction could be on the CAL-DELTAL_EXTERNAL_DQ calibrated strain below 10 Hz. This was done by selecting the appropriate auxiliary channels that are relevant to the strain between the 1-10 Hz frequency regime. The motivation was to contribute to the study of the search for GW memory from core-collapse supernovae, in which the low-frequency GW emission is theoretically found between the 10^{-5} Hz and 50 Hz range. [2] Previous experiments utilizing the NonSENS algorithm have found significant improvement with doing subtractions over the 10 Hz range; however, none was ever done below.

This project proved that there are some relevant noise improvements below 10 Hz utilizing certain SUS and ASC auxiliary channels. However, the results proved that it may not be completely viable to subtract a large amount at such a low range due to the capabilities of current LIGO detectors. The sensitivity of the detectors would need to be increased significantly in order to capture the GW emission from a core-collapse supernovae below 10 Hz. Therefore, it is not practical to attempt further subtraction nor run the algorithm through all of the O3 data if the results show a lack of improvement in the noise mitigation to be effective at this time.

7 Acknowledgements

I would like to sincerely thank my LIGO SURF mentor Dr. Gabriele Vajente for his guidance and support with this project, as well as the Caltech LIGO SURF Program and the National Science Funding (NSF) for funding this project over the summer. I would also like to thank Dr. Alan Weinstein, Derek Davis, and the other faculty members for their support and insights during this program.

References

[1] G. Vajente et al. "Machine-learning nonstationary noise out of gravitational-wave detectors". In: *Physical Review D* 101.4 (Feb. 2020), pp. 1–3, 6. ISSN: 2470-0029. DOI: 10.1103/physrevd.101.042003.

[2] Colter Richardson et al. *Modeling Core-Collapse Supernovae Gravitational-Wave Memory in Laser Interferometric Data*. 2021. arXiv: 2109.01582 [astro-ph.HE].

[3] B. P. Abbott et al [LIGO Scientific Collaboration]. "LIGO: The Laser Interferometer Gravitational-Wave Observatory". In: *Reports on Progress in Physics* 72.7 (2009), p. 6.

[4] B P Abbott et al. "A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals". In: *Classical and Quantum Gravity* 37.5 (Feb. 2020), p. 055002. ISSN: 1361-6382. DOI: 10.1088/1361-6382/ab685e. URL: http://dx.doi.org/10.1088/1361-6382/ab685e.

[5] Marek Szczepanczyk et al. "Detecting and Reconstructing Gravitational Waves From the Next Galactic Core-Collapse Supernova in the Advanced Detector Era". In: (2021), pp. 2–3. DOI: https://arxiv.org/pdf/2104.06462.pdf.

[6] D. V. Martynov et al. "Sensitivity of the Advanced LIGO detectors at the beginning of gravitational wave astronomy". In: *Physical Review D* 93.11 (June 2016). ISSN: 2470-0029. DOI: 10.1103/physrevd.93.112004. URL: http://dx.doi.org/10.1103/PhysRevD.93.112004.

[7] Derek Davis et al. "Improving the sensitivity of Advanced LIGO using noise subtraction". In: *Classical and Quantum Gravity* 36.5 (Feb. 2019), p. 055011. ISSN: 1361-6382. DOI: 10.1088/1361-6382/ab01c5. URL: http://dx.doi.org/10.1088/1361-6382/ab01c5.

[8] G. Vajente. *NonSENS: NON-Stationary Estimation of Noise Subtraction*. 2021. URL: https://git.ligo.org/gabriele-vajente/nonsens.

[9] A D Viets et al. "Reconstructing the calibrated strain signal in the Advanced LIGO detectors". In: *Classical and Quantum Gravity* 35.9 (Apr. 2018), p. 095015. ISSN: 1361-6382. DOI: 10.1088/1361-6382/aab658. URL: http://dx.doi.org/10.1088/1361-6382/aab658.

[10] G. Vajente. *Non-stationary noise subtraction at low frequency*. 2021. URL: https://dcc.ligo.org/DocDB/0174/G2100346/001/JIS_NoiseSubtraction.pdf.

[11] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[12] P. Welch. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[13] G. Vajente. *bruco: Brute Force Coherence*. 2021. URL: https://git.ligo.org/gabriele-vajente/bruco.

[14] Tomislav Andric and Jan Harms. "Lightsaber: A Simulator of the Angular Sensing and Control System in LIGO". In: *Galaxies* 9.3 (Sept. 2021), p. 61. ISSN: 2075-4434. DOI: 10.3390/galaxies9030061. URL: http://dx.doi.org/10.3390/galaxies9030061.