Note on Normalization and Whitening of Gravitational Wave Data

Raghav Girgaonkar *

Department of Physics and Astronomy
The University of Texas Rio Grande Valley

March 2024

1 Introduction

We use the following conventions for the Discrete Fourier and Inverse Fourier Transforms (also adopted by `MATLAB`). The Fast Fourier Fourier transform calculates:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i\frac{2\pi}{N}n} \tag{1}$$

where $\{x_n\}$ is a vector having N samples, and $\{X_N\}$ is it Discrete Fourier Transform. Similarly, the inverse FFT calculates:

$$x_n = \frac{1}{N} \sum_{K=0}^{N-1} X_k \cdot e^{i\frac{2\pi}{N}n} \tag{2}$$

2 Data Whitening

Consider $\overline{d}$ to be a vector of detector data sampled at a sampling frequency of F_s and $\overline{S_n}(f)$ to be the two-sided power spectral density of the data calculated by the Welch's method (such as by using the `pwelch` function in `MATLAB`). By convention, the whitened data is normalized to unit variance [1]. This can be acheived by muliplying a normalization factor of $1/\sqrt{1/\Delta t}$ where Δt is the sampling interval (Check out the whitening function in: gwosc-tutorial/Data_Guide/Guide_Notebook.ipynb). This factor is effectively $1/\sqrt{F_s}$. This factor arises because a function like `pwelch` scales the power spectrum by a factor of $1/F_s$ in order to return a PSD that is in units power per Hertz. This ensures that the PSD returned by such functions is independent of the sampling frequency. Had the function not done this scaling, the PSD would have been returned for the discrete frequency bins that we would use in a Discrete FFT. However, while whitening, we must use the power in the frequency bins being used in the FFT. Thus we must multiply the PSD with F_s. But since in the expression of whitening there is a $\sqrt{\overline{S_n}(f)}$ in the denominator, this factor becomes $1/\sqrt{F_s}$. Thus, the expression for whitening data is:

$$\overline{d}_W = \texttt{IFFT}\left[\frac{1}{\sqrt{F_s}} \times \left(\tilde{d}./\sqrt{\overline{S_n}(f)}\right)\right] \tag{3}$$

Where ./ denotes pointwise division, $\tilde{d}$ is the FFT of $\overline{d}$ as defined in Eqn. 1 and `IFFT` is the Inverse Fourier transform done as defined in Eqn. 2. This convention for whitening must be followed for data and templates alike.

*Email: raghav.girgaonkar@gmail.com

3 Inner Product

Following the whitening normalization factor in Eqn. 3, our definition of inner product between two discrete vectors $\overline{a}$ and $\overline{b}$ now becomes,

$$\langle \overline{a}, \overline{b} \rangle = \frac{1}{N \cdot F_s} \left(\tilde{a}./\sqrt{S_n(f)} \right) \times \left(\tilde{b}./\sqrt{S_n(f)} \right)^{\dagger} \tag{4}$$

Where $\dagger$ denotes taking the complex conjugate. Here each term contributes a $1/\sqrt{F_s}$, and an extra $1/N$ is used because of the IFFT convention of Eqn. 2 in Eqn 3 to form the final normalization term in Eqn. 4. The $1/N$ also arises from the fact that by definition, the matched filtering time-series of $\overline{a}$ and $\overline{b}$ is calculated as filtering a whitened (and normalized) data vector with another whitened (and normalized) data vector,

$$\overline{M}(\overline{a}, \overline{b}) = \texttt{IFFT} \left[\left(\frac{1}{\sqrt{F_s}} \times \tilde{a}./\sqrt{S_n(f)} \right) .\times \left(\frac{1}{\sqrt{F_s}} \times \tilde{b}./\sqrt{S_n(f)} \right)^{\dagger} \right] \tag{5}$$

4 Signal Injection and Normalization

Broadly, there are two ways to inject a simulated GW signal of a certain SNR into strain data: (i) Inject a signal directly into the strain (before any whitening is done) and (ii) Inject a whitened and normalized signal into whitened and normalized strain. We consider both cases, starting with the first.

4.1 Injecting Signal directly into strain

Consider $\overline{s}$ to be an un-normalized simulated GW signal vector at arbitrary SNR and injected into unwhitened detector strain $\overline{d}$. The first step would be to normalize this to a preset SNR ρ. This can be achieved by first normalizing $\overline{s}$ to unit SNR using the inner product definition given in Eqn. 4. This leads to a signal ($\overline{s}'$) normalized to an SNR ρ,

$$\overline{s}' = \frac{\rho \cdot \overline{s}}{\sqrt{\frac{1}{N \cdot F_s} \left(\frac{\tilde{s}}{\sqrt{S_n(f)}} \right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}} \right)^{\dagger}}} \tag{6}$$

The normalization factor is now

$$N_f = \frac{\sqrt{F_s} \times \rho}{\sqrt{\frac{1}{N} \left(\frac{\tilde{s}}{\sqrt{S_n(f)}} \right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}} \right)^{\dagger}}} \tag{7}$$

$$\Rightarrow \overline{s}' = N_f \times \overline{s} \tag{8}$$

The signal $\overline{s}'$ can now be directly injected into the unwhitened detector strain $\overline{d}$. The matched filtering timeseries for $\overline{s}'$ and $\overline{d}$ can be calculated using the expression in Eqn. 5.

4.2 Injecting whitened signal into whitened strain

Consider that we have already whitened and normalized detector strain $\overline{d}$ into $\overline{d}_W$ as done in Eqn. 3. Also consider as similar to Section 4.1, we start with $\overline{s}$ to be an un-normalized simulated GW signal vector at arbitrary SNR. Thus, we start with,

$$\overline{d}_W = \texttt{IFFT} \left[\frac{1}{\sqrt{F_s}} \times \left(\tilde{d}./\sqrt{S_n(f)} \right) \right]$$

We can now first normalize the signal as we did in Eqn. 8 using the expression in Eqn. 7. Thus we get,

$$\overline{s}' = N_f \times \overline{s}$$
$$N_f = \frac{\sqrt{F_s} \times \rho}{\sqrt{\frac{1}{N}\left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right)^{\dagger}}}$$

Note that now, we need to whiten and normalize $\overline{s}'$ before we can inject it in $\overline{d}_W$. This whitening and normalizing should be done the same way as given in Eqn. 3. I.e,

$$\overline{s}'_W = \texttt{IFFT}\left[\frac{1}{\sqrt{F_s}} \times \left(\tilde{s}'./\sqrt{S_n(f)}\right)\right] \tag{9}$$

This can now be re-written as,

$$\overline{s}'_W = \texttt{IFFT}\left[\frac{N_f}{\sqrt{F_s}} \times \left(\tilde{s}./\sqrt{S_n(f)}\right)\right] \tag{10}$$
$$= \texttt{IFFT}\left[\frac{\rho}{\sqrt{\frac{1}{N}\left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right)^{\dagger}}} \times \left(\tilde{s}./\sqrt{S_n(f)}\right)\right] \tag{11}$$

Comparing Eqn. 11 with Eqn. 6, we see that the inner product needed for normalizing a whitened signal is different. The expression for this inner product is now,

$$\langle \overline{a}, \overline{b} \rangle_W = \frac{1}{N}\left(\tilde{a}./\sqrt{S_n(f)}\right) \times \left(\tilde{b}./\sqrt{S_n(f)}\right)^{\dagger} \tag{12}$$

From Eqn. 11 we also then have a new normalization factor,

$$N_{fW} = \frac{\rho}{\sqrt{\frac{1}{N}\left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right)^{\dagger}}} \tag{13}$$
$$\Rightarrow \overline{s}'_W = \texttt{IFFT}\left[N_{fW} \times \left(\tilde{s}./\sqrt{S_n(f)}\right)\right] \tag{14}$$

From Eqn. 14 we can now trace the steps of injecting $\overline{s}'_W$ into $\overline{d}'_W$:

1. Create a un-normalized signal vector $\overline{s}$ at arbitrary SNR.
2. Create $\tilde{s}' = \tilde{s}./\sqrt{S_n(f)}$
3. Calculate N_{fW} as,
$$N_{fW} = \frac{\rho}{\sqrt{\frac{1}{N}(\tilde{s}') \times (\tilde{s}')^{\dagger}}}$$
4. Create $\overline{s}'_W = \texttt{IFFT}\left[N_{fW} \times \tilde{s}'\right]$

This $\overline{s}'_W$ can now be injected into $\overline{d}_W$.

The matched filtering time series using this set of whitened and normalized vectors is now defined as,

$$\overline{M}_W(\overline{d}_W, \overline{s}'_W) = \texttt{IFFT}\left[\tilde{d}_W. \times (\tilde{s}'_W)^{\dagger}\right] \tag{15}$$

which is consistent with the definition in Eqn. 5.

5 Code Implementation and Optimization

When creating a pipeline to perform matched filtering on detector data taken from a source such as the Gravitational Wave Open Science Center (GWOSC) (https://gwosc.org), one generally starts off with calibrated strain data $\overline{d}$. Considering that we want to find the matched filtering timeseries of $\overline{d}$ with a template $\overline{s}$, we can use Eqn. 5 to do so. This is the case when the pipeline is designed in a way that the detector and template vectors need not be whitened and normalized as it is taken care of in the expression for matched filtering. However, pipelines generally whiten and normalize the detector data separately before any matched filtering is done.

Therefore, we first create a whitened and normalized strain vector $\overline{d}_W$ using Eqn. 3.

$$\overline{d}_W = \texttt{IFFT}\left[\frac{1}{\sqrt{F_s}} \times \left(\tilde{d}./\sqrt{S_n(f)}\right)\right]$$

The matched filtering of $\overline{d}_W$ with respect to a template $\overline{s}$ made with certain parameters can then be done optimally as described below.

Matched filtering can be done effectively in the Fourier domain, so one can design the pipeline workflow such that one need not perform an IFFT operation until the final step of creating the matched filtering timeseries. Starting from the whitened and normalized $\overline{d}_W$, we can further create a vector $\tilde{d}'_W$ which we can refer to as the FFT of the *double whitened* data vector. This is defined as,

$$\tilde{d}'_W = \tilde{d}_W./\sqrt{S_n(f)} \tag{16}$$

This saves us the step of whitening our templates every time matched filtering is done. Note that this is possible because $S_n(f)$ is a real valued vector with no imaginary components. Since we have already whitened and normalized our data, the factor we would use to normalize our templates to unit SNR ($\rho = 1$) would be the one given in Eqn. 13. Taking a closer look at this factor, we first note that the FFT of any signal waveform can be expressed in the form of,

$$\tilde{s} = A(f)e^{-i\Psi(f)}$$

Where $A(f)$ is the frequency magnitude term which in the case of the 2PN waveform is $\propto f^{-7/6}$. We can substitute this expression in Eqn. 13, which gives us,

$$N_{fW} = \frac{1}{\sqrt{\frac{1}{N}\left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right) \times \left(\frac{\tilde{s}}{\sqrt{S_n(f)}}\right)^{\dagger}}} \tag{17}$$

$$= \frac{1}{\sqrt{\frac{1}{N}\left(\frac{A(f)e^{-i\Psi(f)}}{\sqrt{S_n(f)}}\right) \times \left(\frac{A(f)e^{-i\Psi(f)}}{\sqrt{S_n(f)}}\right)^{\dagger}}} \tag{18}$$

$$= \frac{1}{\sqrt{\frac{1}{N}\left(\frac{A^2(f)(e^{-i\Psi(f)} \times e^{i\Psi(f)})}{S_n(f)}\right)}} \tag{19}$$

$$\Rightarrow N_{fW} = \frac{1}{\sqrt{\frac{1}{N}\left(\frac{A^2(f)}{S_n(f)}\right)}} \tag{20}$$

From Eqn. 20 we see that the normalization factor does not depend on the parameters of the signal (such as the masses or the chirp times) but rather only on the data properties such as the PSD and the number of data samples. It also depends on the freqeuncy magnitude term $A(f)$. Hence, for optimization purposes, we can pre-calculate this factor for every data segment $\overline{d}$ that we analyze.

We can now use this factor to normalize our templates to form the FFT of our quadrature templates. I.e,

$$\tilde{q} = N_{fW} \times \tilde{s} \tag{21}$$

Taking a close look at the expression for matched filtering for this case (Eqn. 15) we can re-write this as follows,

$$\overline{M}_W(\tilde{d}'_W, \tilde{q}) = \texttt{IFFT}\left[\tilde{d}'_W. \times (\tilde{q})^\dagger\right] \tag{22}$$

Expanding the expression for $\tilde{q}$, this then becomes,

$$\overline{M}_W(\tilde{d}'_W, \tilde{q}) = \texttt{IFFT}\left[\tilde{d}'_W. \times (N_{fW} \times \tilde{s})^\dagger\right] \tag{23}$$

$$= \texttt{IFFT}\left[\tilde{d}'_W. \times (N_{fW} \times A(f)e^{-i\Psi(f)})^\dagger\right] \tag{24}$$

$$= \texttt{IFFT}\left[(A(f) \times \tilde{d}'_W). \times (N_{fW} \times e^{-i\Psi(f)})^\dagger\right] \tag{25}$$

We now have Eqn. 25 as the optimized version of matched filtering.

To summarize the steps described in this section, an optimized implementation of matched filtering detector data $\overline{d}$ having estimated two-sided PSD $S_n(f)$ with a signal template $\overline{s}$ of certain mass or chirp time parameters can be done as follows:

1. Create whitened and normalized $\overline{d}_W$ as per Eqn. 3.
2. Double whiten $\overline{d}_W$ to create $\tilde{d}'_W$ as per Eqn. 16.
3. Create normalization factor N_{fW} as per Eqn. 20.
4. Multiply $\tilde{d}'_W$ with the frequency magnitude term $A(f)$, i.e $\tilde{d}'_{WO} = A(f) \times \tilde{d}'_W$. This can be referred to as `fftdatabyPSD`.
5. Calculate only the phase Fourier term of the template $\tilde{s}$ and normalize to SNR 1 by multiplying by the factor N_{fW}.
6. The matched filtering time-series can then be calculated by,

$$\overline{M}_W = \texttt{IFFT}\left[(\tilde{d}'_{WO}). \times (N_{fW} \times e^{-i\Psi(f)})^\dagger\right] \tag{26}$$

7. For two quadrature templates, we can find two matched filtering time-series

$$\overline{M}_{W1} = \texttt{IFFT}\left[(\tilde{d}'_{WO}). \times (N_{fW} \times e^{-i\Psi(f)})^\dagger\right] \tag{27}$$

$$\overline{M}_{W2} = \texttt{IFFT}\left[(\tilde{d}'_{WO}). \times (N_{fW} \times e^{-i(\Psi(f)+\pi/2)})^\dagger\right] \tag{28}$$

8. The final time-series can then be obtained by,

$$\overline{M}_{WF} = \sqrt{\overline{M}_{W1}^2 + \overline{M}_{W2}^2} \tag{29}$$

The peak value of $\overline{M}_{WF}$ will be the estimated SNR for the template $\overline{s}$.

Note: One could potentially also pre-multiply the normalization factor N_{fW} to create $\tilde{d}'_{WO} = N_{fW} \times A(f) \times \tilde{d}'_W$ which would then require only the phase Fourier term in the matched filtering expression.

The above optimization steps are employed in the PSO-based matched filtering pipeline present in https://github.com/RaghavGirgaonkar/Accelerated-Network-Analysis.

References

[1] B. P. e. a. Abbott, “A guide to ligo–virgo detector noise and extraction of transient gravitational-wave signals,” *Classical and Quantum Gravity*, vol. 37, p. 055002, Feb. 2020.