

A Novel Method for Removing Cyclic Frequency Modulation in Fourier Space

Curtis Rau^{1,2}, Colin Gordon^{1,2}, David Witalka^{1,2}, and Samuel Akwei-Sekyere^{1,3}

*Department of Mathematics*¹

*Department of Physics and Astronomy*²

*Neuroscience Program*³

Michigan State University

By building on the method established in [5], this paper primarily seeks to introduce the Gustafson algorithm for removing frequency/phase modulation from a signal. The Gustafson algorithm uses the time series data as its input and outputs the original single tone carrier frequency. This method is powerful because it works in the frequency rather than time domain — which easily allows for filtering of the signal. Here, we will focus on the Gustafson algorithm's applications to LIGO and the search for gravitational waves; nevertheless, there are many other possible applications for the algorithm such as radio wave astronomy and communications systems using frequency modulated signals. Since one of the pertinent issues in any detection algorithm is signal adulteration by noise, we shall also introduce a novel approach to deal with correlated noise from a multichannel perspective.

DIVISION OF LABOR

Curtis Rau: Curtis was the most familiar with the topic and therefore performed the role of group leader identifying topics and problems of interest to be researched. He was also in charge of writing sections: "The Gustafson Algorithm", and "The Gustafson Algorithm as a Search Algorithm". He also wrote the section "Implementation of the Gustafson Algorithm in C++" for another class, along with the C++ code. He presented his part in the class presentation.

Colin Gordon: Colin contributed greatly in his research and solving of the phase modulation problem for astronomical signals caused by rotation of the earth using circular orbit approximation. He wrote the introduction to the paper, advised Sam on his section for noise and error analysis, and helped review and edit the paper as a whole. He presented his work in class.

David Witalka: David researched and wrote the sections concerning the Nyquist frequency and Carson's rule. He also added the section pertaining to the speed of the Gustafson Algorithm. David edited and reworded the introduction and helped review and edit the paper as a whole. He also created and organized the slides presented in class.

Samuel Akwei-Sekyere: Sam researched and wrote the section on noise and error analysis, and wrote the results. He helped greatly in rewording problem sections throughout the paper, and greatly expanded the discussion of all topics. Sam played a major role in reviewing and editing others' sections. He presented his work in class. Sam also wrote the Matlab code for the noise removal.

I. INTRODUCTION

General Relativity postulates that information about the spatial distribution of matter in the universe is transmitted via a gravitational field [4]. Information regarding changes in mass distribution of the universe propagate through space-time away from the location of that change at the speed of light. Special events, where the quadrupole moment of the mass distribution changes, emit gravitational radiation [12]. Extremely violent events in the universe can release enough gravitational radiation to be detectable from Earth.

Recently, the Laser Interferometer Gravitational-Wave Observatory (LIGO) obtained the first empirical evidence of gravitational waves. The discovered ripples in space-time by LIGO was a function of the inspiral of two black holes that eventually merged — an event that consumed approximately three solar masses [6]. It is believed that gravitational wave sources are common in the universe and sources of continuous monochromatic (single frequency) gravitational radiation are predicted to exist — at least sources with frequencies that persist over many periods [10]. Neutron stars have also been implicated as one of the prime sources of gravitational waves. These stars are expected to emit single frequency waves over long periods of time. Additionally, it has also been proposed that an appreciable fraction of detectable gravitational wave emitters are binary neutron star systems — two neutron stars orbiting around each other.

A fundamental challenge to efficient detection of gravitational waves is that orbiting binary neutron star systems invoke frequency modulations which reduce the power of the carrier frequency and distributes it to higher and lower harmonics. In essence, the power of the signal is dispersed over a range of side-band frequencies; this phenomenon is due to the Doppler effect arising from the Earth's motion relative to the source of the wave. Since the noise observed in the LIGO data is approximately Gaussian and of low signal-to-noise ratio, the extraction of reliable information about gravitational waves is arduous.

In order to calculate the carrier frequency of the wave, consider the wave as one emitted from a source that is megaparsecs away (potentially in other galaxies). By virtue of its distance, the spherical coordinates will not change significantly and the source will move relative to the sun with nearly constant velocity over the time we collect data. By the same token, the Sun will experience the monochromatic wave produced by the source as truly monochromatic, while its detection on the Earth will be in the form of a frequency modulated signal.

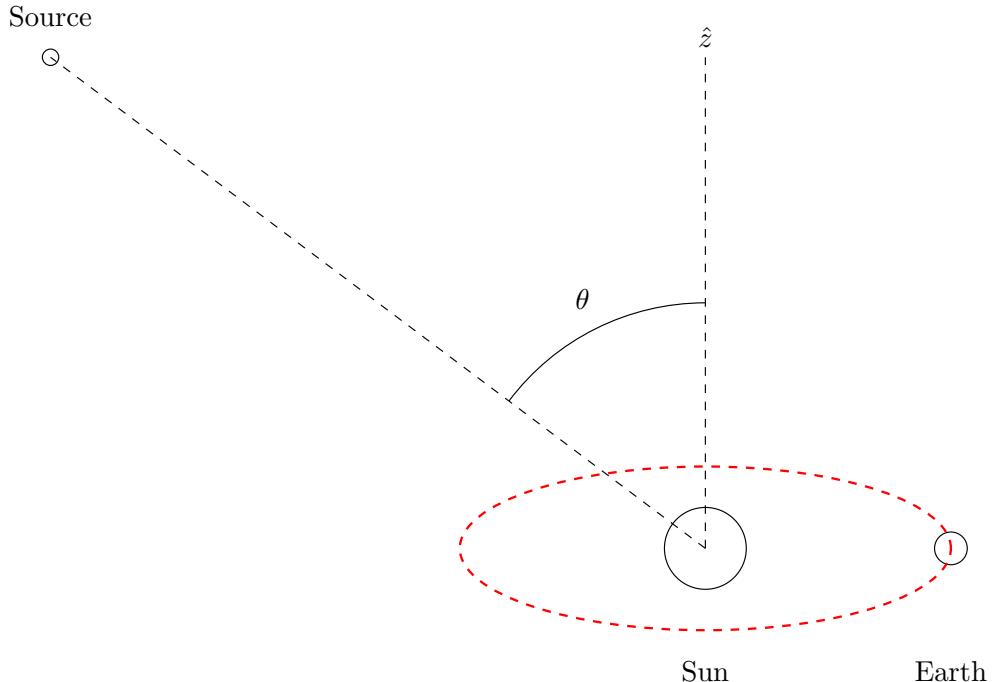


FIG. 1

By assuming that the wave travels at the speed of light, we can capture it as a Lorentz invariant 4-vector, \mathbf{K} with the carrier frequency being the first entry and the second through fourth entries represent the wave number. In order to transform it in the Earth's frame we apply an instantaneous Lorentz boost on the 4-vector; the instantaneous boost takes the direction of the motion of the Earth and the velocity of the Earth in orbit into consideration. In our implementation, the Earth's orbit is captured in the xy plane. The 4-vector is written in the perspective

that assumes the wave is a plane wave which travels towards the Sun on a trajectory that is on the xz plane:

$$\mathbf{K} = \begin{pmatrix} \gamma & -\gamma\beta & 0 & 0 \\ -\gamma\beta & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -\sin(\omega_e t + \phi_e) & \cos(\omega_e t + \phi_e) & 0 \\ 0 & -\cos(\omega_e t + \phi_e) & -\sin(\omega_e t + \phi_e) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_0/c \\ -k \sin \theta \\ 0 \\ -k \cos \theta \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} \gamma(\omega_0/c - \beta k \sin \theta \sin(\omega_e t + \phi_e)) \\ \gamma(-\beta \omega_0/c + k \sin \theta \sin(\omega_e t + \phi_e)) \\ \gamma k \sin \theta \cos(\omega_e t + \phi_e) \\ -k \cos \theta \end{pmatrix}, \quad (2)$$

where θ is the polar angle from the axis perpendicular to the orbit of Earth, γ is the Lorentz factor from special relativity, ω_e is the angular frequency of the Earth's rotation about the Sun, k is the magnitude of the wave number, c is the speed of light and β is the velocity of Earth's orbit divided by the speed of light.

From this formulation we can infer that the instantaneous frequency in the Earth's frame can be given by:

$$\omega(t) = \omega_0 \gamma (1 - \beta \sin \theta \sin(\omega_e t + \phi_e)). \quad (3)$$

Since the phase refers to the time integral of the frequency, the expression for the wave being sought after is the real part of $h(t)$:

$$h(t) = h_0 e^{i(\omega' t + \Gamma \cos(\omega_e t + \phi_e))} \quad (4)$$

$$\omega' = \gamma \omega_0 \quad (5)$$

$$\Gamma = \frac{\gamma \beta \omega_0 \sin \theta}{\omega_e} \quad (6)$$

where h_0 is the amplitude of the wave, Γ is the modulation index, ω_0 is the frequency of the wave to be detected, ϕ_e is the relative phase difference between the Earth's rotation about the Sun and the wave to be detected.

An important note is that Γ is a function of only the azimuthal angle of the source with respect to the normal of the galactic plane. This algorithm will also work perfectly well with any frequency modulated wave produced similar to the above. One likely scenario is that when the compact stars are in a binary system, one or both objects produce waves but the stars rotate around each other causing an analogous phase modulation. Withal, there could be applications for the Gustafson algorithm in areas outside of gravitational waves in areas such as radio wave astronomy and communications using frequency modulated signals. We attempt to keep the discussion of the algorithm and the results as general as possible for this reason.

II. THE GUSTAFSON ALGORITHM

The goal of this paper is to propose a search algorithm that can efficiently detect the presence of frequency modulated waves, while being robust against noise. A brute force approach to accomplish this is to take the inner product of

$$h(t) = h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \quad (7)$$

with the data for different values of ω_0 , ω_1 , ϕ_0 , ϕ_1 , and Γ where ϕ_0 is the phase of the wave to be detected at $t = 0$, with ω_1 and ϕ_1 being generalizations of ω_e and ϕ_e respectively. This is effectively a Fourier transform as a function of the four search parameters:

$$\hat{f}(\omega_0, \omega_1, \phi_1, \Gamma) = \int f(t) e^{-i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} dt. \quad (8)$$

It is expected that this approach results in a something like a delta function, or at least a Dirichlet kernel because of discretization, because of the results in the previous section. The amount of time it will take to perform this computation is $\mathcal{O}(N_t N_{\omega_0} N_{\omega_1} N_{\phi_1} N_{\Gamma})$, where N_t denotes the number of data points, and the other N_x is the number

of steps in x that the search will take. Typically for a Discrete Fourier Transform (DFT) the number of steps in frequency is equal to the number of data points: $N_t = N_{\omega_0} = N_{\omega_1}$ [13] [7, P. 251]. Physically, this infers that the frequency resolution of the DFT is proportional to the duration of the time series over which the DFT is taken. Thus the computation time for the brute force method goes as $\mathcal{O}(N_t^3 N_{\phi_1} N_{\Gamma})$.

The case when the signal is buried deep within non-coherent noise is of particular interest. A conventional method to improve the signal to noise ratio (SNR) is by increasing the number of data points; this assumes the signal is more periodic than noise. The problem with taking more data however is that computational time of the brute force method is proportional to N_t^3 . We will now introduce the Gustafson Algorithm which is roughly a factor of N_t faster than the brute force method.

To derive the Gustafson Algorithm we start by assuming the data $f(t)$ takes on the form of a complex frequency modulated wave. Although the recorded data is real, it will be shown later that starting with the complex wave is perfectly valid. First, we demodulate the wave and take the Fourier transform of both sides:

$$h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} = f(t) \quad (9)$$

$$h_0 e^{i\phi_0} e^{i\omega_0 t} = f(t) e^{-i\Gamma \cos(\omega_1 t + \phi_1)} \quad (10)$$

$$h_0 e^{i\phi_0} \mathcal{F}_t [e^{i\omega_0 t}] = \mathcal{F}_t [f(t) e^{-i\Gamma \cos(\omega_1 t + \phi_1)}]. \quad (11)$$

Define $\mathcal{F}_t[f(t)](\omega) = \hat{f}(\omega)$. By invoking the convolution theorem yields,

$$\frac{h_0 e^{i\phi_0}}{\sqrt{2\pi}} \delta(\omega_c - \omega) = \hat{f}(\omega) * \mathcal{F}_t [e^{-i\Gamma \cos(\Omega t + \phi)}]. \quad (12)$$

The Jacobi-Anger Expansion, which comes up often in this paper, has the form [?]:

$$e^{iz \cos(\theta)} = \sum_{n=-\infty}^{\infty} e^{in\pi/2} J_n(z) e^{in\theta}. \quad (13)$$

By invoking the previously mentioned expansion, equation 12 becomes

$$\frac{h_0 e^{i\phi_0}}{\sqrt{2\pi}} \delta(\omega - \omega_0) = \hat{f}(\omega) * \mathcal{F}_t \left[\sum_{n=-\infty}^{\infty} e^{in\pi/2} J_n(-\Gamma) e^{in(\omega_1 t + \phi_1)} \right]. \quad (14)$$

Even order Bessel Functions are even, and odd order Bessel Functions are odd [9]. Explicitly

$$J_n(-z) = (-1)^n J_n(z) \quad (15)$$

By employing this fact we observe that

$$\mathcal{F}_t \left[\sum_{n=-\infty}^{\infty} e^{in\pi/2} J_n(-\Gamma) e^{in(\omega_1 t + \phi_1)} \right] = \sum_{n=-\infty}^{\infty} (-1)^n e^{in\pi/2} e^{in\phi_1} J_n(\Gamma) \mathcal{F}_t [e^{in\omega_1 t}] \quad (16)$$

$$= \frac{1}{\sqrt{2\pi}} \sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \pi/2)} J_n(\Gamma) \delta(\omega - n\omega_1). \quad (17)$$

In the previous step we used the linearity of the Fourier transform $\mathcal{F}_t[c \cdot f(t)] = c \cdot \mathcal{F}_t[f]$. Note that 14 becomes

$$h_0 e^{i\phi_0} \delta(\omega - \omega_0) = \hat{f}(\omega) * \left[\sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \pi/2)} J_n(\Gamma) \delta(\omega - n\omega_1) \right]. \quad (18)$$

By performing the convolution,

$$\hat{f}(\omega) \star \delta(\omega - n\omega_1) = \int \hat{f}(\omega - \tilde{\omega}) \delta(\tilde{\omega} - n\omega_1) d\tilde{\omega} \quad (19)$$

$$= \hat{f}(\omega - n\omega_1) \quad (20)$$

brings us to the almost final answer

$$h_0 e^{i\phi_0} \delta(\omega - \omega_0) = \sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \pi/2)} J_n(\Gamma) \hat{f}(\omega - n\omega_1). \quad (21)$$

For clarity and later convenience let's have $n \rightarrow -n$. Let us employ a property of the Bessel functions: [9]

$$J_{-n}(z) = (-1)^n J_n(z). \quad (22)$$

This results in the Gustafson Algorithm:

$$h_0 e^{i\phi_0} \delta(\omega - \omega_0) = 2 \sum_{n=-\infty}^{\infty} e^{-in(\phi_1 + \pi/2)} J_n(\Gamma) \hat{f}(\omega + n\omega_1) \quad (23)$$

An erroneous factor of two has been added to the algorithm which, as we will see, makes it suitable for use with real data. It is not obvious whether or not this algorithm in its current form can be used on real data because it assumed a complex waveform. This assumption allowed for a clean demodulation, which is marred if one takes the real of this function. This is due to the nonlinearity of the real operator as demonstrated by

$$\Re\{a \cdot b\} \neq \Re\{a\} \cdot \Re\{b\}; \quad a, b \in \mathbb{C}. \quad (24)$$

Naively we try plugging the Fourier Transform of the real waveform (eq. B6) into the Gustafson Algorithm (eq. 23) and see what happens. Perhaps unsurprisingly using either the real or complex waveforms yield the same result. Notice that the Fourier Transform of the real wave looks very similar to the Fourier Transform of the complex wave (see appendices A and B). The difference being that the Fourier Transform of the real is symmetric about $\omega = 0$ up to a phase factor. The vectors corresponding to the positive carrier frequency rotate counterclockwise, whereas the vectors corresponding to the negative carrier frequency rotate clockwise (neglecting the rotation due to ϕ_1 which is the same for both). **This phase factor is the reason why the Gustafson Algorithm only picks out the positive value of ω_0 .** If it is unclear what this means refer to figure (12e).

Now we will try to build an intuitive conceptual model of the Gustafson Algorithm. The complex numbers form a two dimensional vector field over the real numbers. This means the Fourier Transform is a collection of three dimensional vectors. If we plot the vectors in cylindrical coordinates, where the z-axis is frequency, the distance from the z-axis represents the amplitude of that frequency, and the polar angle represents the phase of that frequency component, then we get something like figure (2).

The Gustafson Algorithm rotates these vectors, multiplies them by a constant, and takes the vector sum of them. Apparently this vector sum is zero unless the frequency ω is exactly equal to ω_0 in which case they sum to a vector of infinite length. The unincluded vectors of the real waveform in figure (2) are symmetric about $\omega = 0$ except that they rotate the opposite way around the frequency axis. This asymmetry in phase is why the Gustafson Algorithm is asymmetric. It rotates all vectors the same direction. Apparently the algorithm rotates vectors corresponding to $+\omega_0$ so that they constructively add, whereas it rotates vectors corresponding to $-\omega$ in the wrong direction so they destructively add.

A. The Gustafson Algorithm as a Search Algorithm

It is not explicitly obvious how The Gustafson Algorithm, Eq. 23, can be used as a search algorithm capable of determining the values $h_0, \omega_0, \omega_1, \phi_0, \phi_1, \Gamma$. In this section we seek to make obvious what Eq. 23 can do.

Assume the data takes on the real form of a frequency modulated wave.

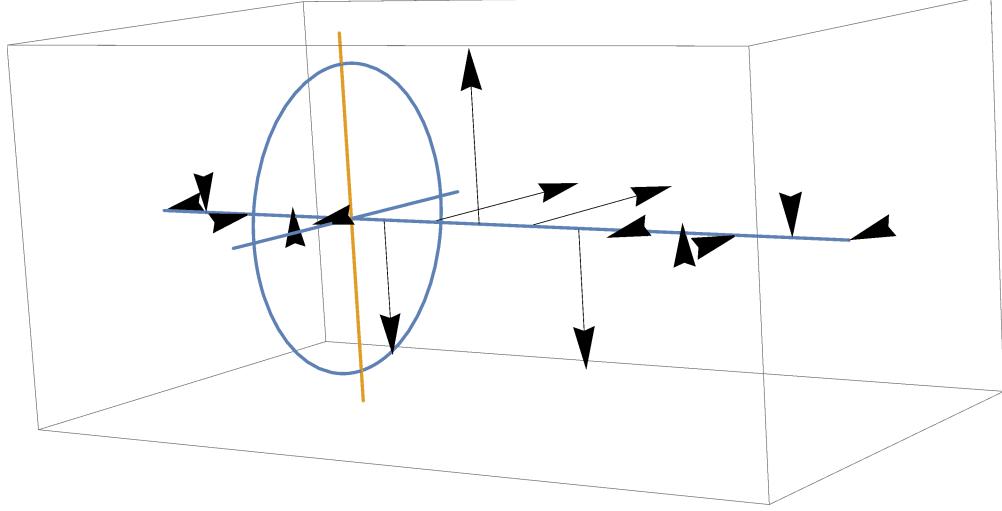


FIG. 2: The 3D representation of the Fourier Transform of the complex waveform.

$$f(t) = \Re \left\{ h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \right\} \quad (25)$$

Initially we do not know the values of $\omega_0, \omega_1, \phi_0, \phi_1, \Gamma$. To determine these values we will try the values $\tilde{\omega}_0, \tilde{\omega}_1, \tilde{\phi}_1, \tilde{\Gamma}$ in the Gustafson Search Algorithm.

$$G_{\tilde{\omega}_0, \tilde{\omega}_1, \tilde{\phi}_1, \tilde{\Gamma}} [\hat{f}(\omega)] = \sum_{n=-\infty}^{\infty} e^{-in(\tilde{\phi}_1 + \pi/2)} J_n(\tilde{\Gamma}) \hat{f}(\omega + n\tilde{\omega}_1) \quad (26)$$

To see what this will yield we substitute the Fourier Transform of the real of the waveform, equation (B6), into the above equation

$$\begin{aligned} G_{\omega_0, \omega_1, \phi_1, \Gamma} [\hat{f}(\omega)] &= h_0 \sum_{m,n=-\infty}^{\infty} e^{i(m\phi_1 - n\tilde{\phi}_1)} J_n(\tilde{\Gamma}) J_m(\Gamma) \\ &\times [e^{i(\phi_0 + (m-n)\pi/2)} \delta(\omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1) + e^{-i(\phi_0 + (m+n)\pi/2)} \delta(\omega + \omega_0 + n\tilde{\omega}_1 - m\omega_1)] \end{aligned} \quad (27)$$

Notice that the erroneous factor of two in front of the Gustafson Algorithm (Eq. 23) has canceled with the factor of one half in front of the Fourier Transform of the real waveform (Eq. B6). This equation is far to complicated to see what is going on, so we will treat it in cases.

We shall treat this in cases.

Case 1: Let us consider the most likely case when $\omega \neq \omega_0$ and $\omega_1 \neq \tilde{\omega}_1$. Let us examine the term $\delta(\omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1)$ in the above equation in detail. This delta function appears in the double sum and is a function of both m and n , so the $n\tilde{\omega}_1 - m\omega_1$ is changing over the sums whereas the value $\omega - \omega_0$ is constant (not a function of m or n). This delta is non-zero only if

$$0 = \omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1 \quad (28)$$

$$\omega_0 - \omega = n\tilde{\omega}_1 - m\omega_1 \quad (29)$$

This can happen twice or never. By the Euclidean Algorithm, the delta will only be non-zero if

$$\gcd(\tilde{\omega}_1, \omega_1) = \omega_0 - \omega \quad (30)$$

If this is false, then the sum is zero. If it is true then call the values for m and n that make it true $\pm m_o$ and $\mp n_o$. So then the sum is

$$G_{\tilde{\omega}_0, \tilde{\omega}_1, \tilde{\phi}_1, \tilde{\Gamma}} [\hat{f}(\omega)] = h_0 e^{i\phi_0} \delta(0) J_n(\tilde{\Gamma}) J_m(\Gamma) \cos(m\phi_1 - n\tilde{\phi}_1) \times \begin{cases} (-1)^{(m-n)/2}, & m+n \text{ is even} \\ i(-1)^{m-n}, & m+n \text{ is odd} \end{cases} \quad (31)$$

Note that the only way this could give us the true amplitude $h_0 e^{i\phi_0}$ is if $\tilde{\Gamma} = \Gamma = 0$, $\tilde{\phi}_1 = \phi_1$, and $m = n = 0$. This is because the Bessel Functions are always less than or equal to one. The only time equality holds is for $J_0(0) = 1$. Moreover if $m = n = 0$ then $\tilde{\omega}_0 = \omega_0$ which violates our assumption. Therefore in this case, when $\omega \neq \omega_0$ and $\omega_1 \neq \tilde{\omega}_1$, the amplitude of the greatest peak will always be less than the true amplitude of the wave.

Case 2: Now consider the case when $\omega = \omega_0$ and $\omega_1 \neq \tilde{\omega}_1$. The term $\delta(\omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1)$ will only be non-zero if

$$\tilde{\omega}_1 = \frac{m}{n} \omega_1 \quad (32)$$

This could happen never, or a countably infinite number of times. If this is false, then the sum is zero.

Case 3: Now consider the case when $\omega \neq \omega_0$ and $\omega_1 = \tilde{\omega}_1$. The term $\delta(\omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1)$ will only be non-zero if

$$\omega - \omega_0 = (n - m)\tilde{\omega}_1 \quad (33)$$

This too can happen twice or never. If this is false then the sum is zero.

Case 4: Now consider the case when $\omega = \omega_0$ and $\omega_1 = \tilde{\omega}_1$. The term $\delta(\omega - \omega_0 + n\tilde{\omega}_1 - m\omega_1)$ will only be non-zero if $m = n$ which will happen an infinite number of times. Thus, the sum becomes

$$\frac{h_0}{2} e^{i\phi_0} \delta(0) \sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \tilde{\phi}_1)} J_n(\tilde{\Gamma}) J_n(\Gamma) \quad (34)$$

Case 4a: Now if we let $\Gamma = \tilde{\Gamma}$ and $\phi_1 = \tilde{\phi}_1$ then the sum becomes

$$\frac{h_0}{2} e^{i\phi_0} \delta(0) \sum_{n=-\infty}^{\infty} J_n^2(\Gamma) = \frac{h_0}{2} e^{i\phi_0} \delta(0) \quad (35)$$

So if we guess every search parameter correctly, $\omega = \omega_0$, $\omega_1 = \tilde{\omega}_1$, $\Gamma = \tilde{\Gamma}$, and $\phi_1 = \tilde{\phi}_1$, then we recover the true amplitude of the wave, as indicated by the Gustafson Algorithm.

III. CARSON'S RULE AND THE NYQUIST FREQUENCY

In order to fully reconstruct the carrier frequency we would need to take an infinite sum over the sidebands. As this is impractical and unnecessary as many terms are insignificant, we need to find a way to limit the number of sidebands we are summing over. The first problem is locating the sidebands' characteristic frequency. There are infinite sideband frequencies and eventually they will be higher than the Nyquist frequency which will cause aliasing of the sideband frequency. The Nyquist frequency, ψ_N , is defined as the maximum frequency that can be accurately reconstructed and is equal to half of the sampling frequency, ψ_s .

$$\psi_N = \frac{1}{2} \psi_s \quad (36)$$

Any side band with a frequency of ψ_0 higher than the Nyquist frequency, but less than the sampling frequency, will be reconstructed with a frequency of $(\psi_s - \psi_0)$. That is the frequency will be under-sampled and an alias frequency will be produced. When this occurs the best case scenario is that the alias frequency is unique and does not correspond to the frequency of another lower frequency sideband. In this case we can use anti-aliasing techniques to correct for the aliasing and still obtain information about the amplitude and frequency of the higher side bands. The worst case scenario is that the aliased frequency is too close to another lower frequency sideband and this will cause two problems: a systematic error will be added to the measurement of the lower side band's

amplitude, and we will be unable to determine the necessary information on what the amplitude of the higher frequency would be. To minimize the effect this might have we would like any sidebands with a frequency over the Nyquist frequency to have as low an amplitude as possible. For this we can employ Carson's rule. Carson's rule can be understood to say that almost all (roughly 98 percent) of the power for a frequency-modulated sinusoidal signal is contained within a finite bandwidth B_T , defined by:

$$B_T = 2(\Delta\psi + \psi_m) \quad (37)$$

where $\Delta\psi$ is the peak frequency deviation of the instantaneous frequency $\psi(t)$ from the center carrier frequency ψ_c , and ψ_m is the highest frequency in the modulating signal. In our case the highest frequency of the modulating signal is the carrier frequency, ψ_c . From this calculation we can adjust our sampling frequency to ensure it is large enough in order to minimize aliasing. This also puts a limit on the number of sidebands we need to sum over to just those that fall within the bandwidth. The important thing to take away from this is that it takes an infinite bandwidth to perfectly transmit a phase or frequency modulated wave, regardless of how smooth it is, but in practice only a finite bandwidth is needed for an accurate approximation.

$$B_T = 2(\Delta\psi + \psi_m) \quad (38)$$

where $\Delta\psi$ is the peak frequency deviation of the instantaneous frequency $\psi(t)$ from the center carrier frequency ψ_c , and ψ_m is the highest frequency in the modulating signal. In our case the highest frequency of the modulating signal is the carrier frequency, ψ_c . From this calculation we can adjust our sampling frequency to ensure it is large enough in order to minimize aliasing. This also puts a limit on the number of sidebands we need to sum over to just those that fall within the bandwidth. The important thing to take away from this is that it takes an infinite bandwidth to perfectly transmit a phase or frequency modulated wave, regardless of how smooth it is, but in practice only a finite bandwidth is needed for an accurate approximation.

IV. ERROR ANALYSIS

While the search for monochromatic gravitational waves via the LIGO detector is relatively effortless, the adulteration of signals obtained from the detector by noise adds an appreciable number of complications to the problem. By the same token, techniques by which improved resistance of the LIGO detector to noise can be obtained has been investigated [2] [1].

A. Noise Sources

Since the signal being searched for is relatively stable but of little amplitude, the reliability of data obtained from the detector is related to the degree of noise therein. Among others, quantum, thermal, seismic and Newtonian noise have been implicated as prime drivers of LIGO data adulteration. Since the prime noise contributor is quantum, a summation of all the noise sources are approximately Gaussian. Thus, it suffices to use the properties of Gaussian noise to understand the volatility of the data.

B. Noise Elimination

As mentioned previously, quantum noise — in particular, thermal noise — has been implicated as a limiting factor in the sensitivity of gravitational wave interferometers and hence a prime factor in the degradation of the signal quality [14]. Following from this, it can be inferred that the distribution of the noise is approximately Gaussian.

By low-pass filtering or spectral subtraction, it is possible to derive signals with high SNR; however, these widely-accepted approaches require information which can be difficult to obtain. Since the frequency of the monochromatic wave being searched for is not known, low-pass filtering result in erroneous outputs. Spectral subtraction requires a sample of the noise; nevertheless, due to the low SNR of LIGO data, it is precarious to find segments with pure noise.

In this section, we introduce an alternate approach by which Gaussian noise in data may be reduced. This method utilizes the following principle: the nature of the noise is not known but the nature of the signal is known. Our methodology consists of subtracting estimated noise from the signal via singular value decomposition of intrinsic mode functions and assumes the noise is correlated across different channels.

1. Proposed Denoising Algorithm

In an effort to reduce the noise, we shall consider data from two channels. By the assumption that the noise observed is additive (and correlated), we yield the following:

$$\xi^{(1)}(t) = f^{(1)}(t) + \epsilon^{(1)}(t) \quad (39)$$

$$\xi^{(2)}(t) = f^{(2)}(t) + \epsilon^{(2)}(t) \quad (40)$$

where $\xi^{(n)}(t)$ is the data observed, $f^{(n)}(t)$ is the true signal and $\epsilon^{(n)}(t)$ is the noise obtained from the n-th channel (or gravitational wave interferometer) with $\epsilon^{(n)}(t) \sim \mathcal{N}(0, \sigma^2)$.

First, we disintegrate each observed signal into α intrinsic mode functions via empirical mode decomposition and arrange them into a $2\alpha \times T$ matrix \mathbf{L} :

$$\mathbf{L} = \begin{pmatrix} l_{1,t_1}^{(1)} & l_{1,t_2}^{(1)} & \cdots & l_{1,t_T}^{(1)} \\ l_{2,t_2}^{(1)} & l_{2,t_2}^{(1)} & \cdots & l_{2,t_T}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ l_{\alpha,t_1}^{(1)} & l_{\alpha,t_2}^{(1)} & \cdots & l_{\alpha,t_T}^{(1)} \\ l_{\alpha+1,t_1}^{(2)} & l_{\alpha+1,t_2}^{(2)} & \cdots & l_{\alpha+1,t_T}^{(2)} \\ l_{\alpha+2,t_2}^{(2)} & l_{\alpha+2,t_2}^{(2)} & \cdots & l_{\alpha+2,t_T}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ l_{2\alpha,t_1}^{(2)} & l_{2\alpha,t_2}^{(2)} & \cdots & l_{2\alpha,t_T}^{(2)} \end{pmatrix} \quad (41)$$

where $l_{p,t_q}^{(n)}$ is the $\left[((p-1)\text{mod } \alpha) + 1 \right]$ -th intrinsic mode function at time t_q of $\xi^{(n)}(t)$, and with T being the total number of data points observed.

Thereafter, the fast Fourier transform of each intrinsic mode function in \mathbf{L} is computed to obtain a matrix $\mathcal{F}(\mathbf{L})$. Singular value decomposition of the variance-covariance matrix obtained from $\mathcal{F}(\mathbf{L})$ is then computed to result in the following:

$$\Sigma = \mathbf{U}\mathbf{S}\mathbf{V} \quad (42)$$

where $\Sigma = T^{-1} [\mathcal{F}(\mathbf{L})^\mathbf{T} - \mathbf{1}\mathbf{1}^\mathbf{T} \mathcal{F}(\mathbf{L})^\mathbf{T} T^{-1}]^\dagger [\mathcal{F}(\mathbf{L})^\mathbf{T} - \mathbf{1}\mathbf{1}^\mathbf{T} \mathcal{F}(\mathbf{L})^\mathbf{T} T^{-1}]$ with $(\cdot)^\mathbf{T}$ being the transpose, the conjugate transpose $(\cdot)^\dagger$ and $\mathbf{1}$ a $T \times 1$ column matrix.

The fast Fourier transforms are de-correlated to form a matrix \mathbf{R} by employing the following:

$$\mathbf{R} = \mathbf{U}^\mathbf{T} \mathbf{L} \quad (43)$$

After the decorrelation procedure above, we seek to apply a threshold Δ which dictates the approximate percentage of statistically irrelevant information about the Fourier transform we shall discard. Prior to the thresholding procedure, the eigenvectors that contribute to approximately Δ percent of the variance of the transform by virtue of their respective eigenvalues is estimated using the following procedure:

$$y = \underset{b}{\operatorname{argmin}} \left(\left| \frac{\sum_{j=1}^{(b)} \lambda_j}{\operatorname{trace}\{\mathbf{S}\}} - \frac{\Delta}{100} \right| \right) \quad (44)$$

where y is the index denoting the y^{th} eigenvalue from λ_y with $\lambda = \operatorname{diag}(\mathbf{S})$.

To apply the threshold, the projections with eigenvalues whose indices are less than y are retained and the rest are replaced with zero row vectors:

$$\mathbf{R}_r = \begin{cases} \mathbf{R}_r & , r < y \\ \mathbf{0} & , r \geq y \end{cases}$$

where \mathbf{R}_r the r^{th} row of the matrix \mathbf{R} .

The fast Fourier transforms are then re-correlated to obtain denoised intrinsic mode functions in the following manner:

$$\tilde{\mathbf{L}} = \mathbf{VR} \quad (45)$$

The real component of the inverse fast Fourier transform of each row of $\tilde{\mathbf{L}}$ is then computed to obtain a matrix \mathbf{D} of similar architecture with denoised intrinsic mode functions. Thereafter, the reconstructed signals $\hat{\xi}^{(n)}(t)$ are obtained as follows:

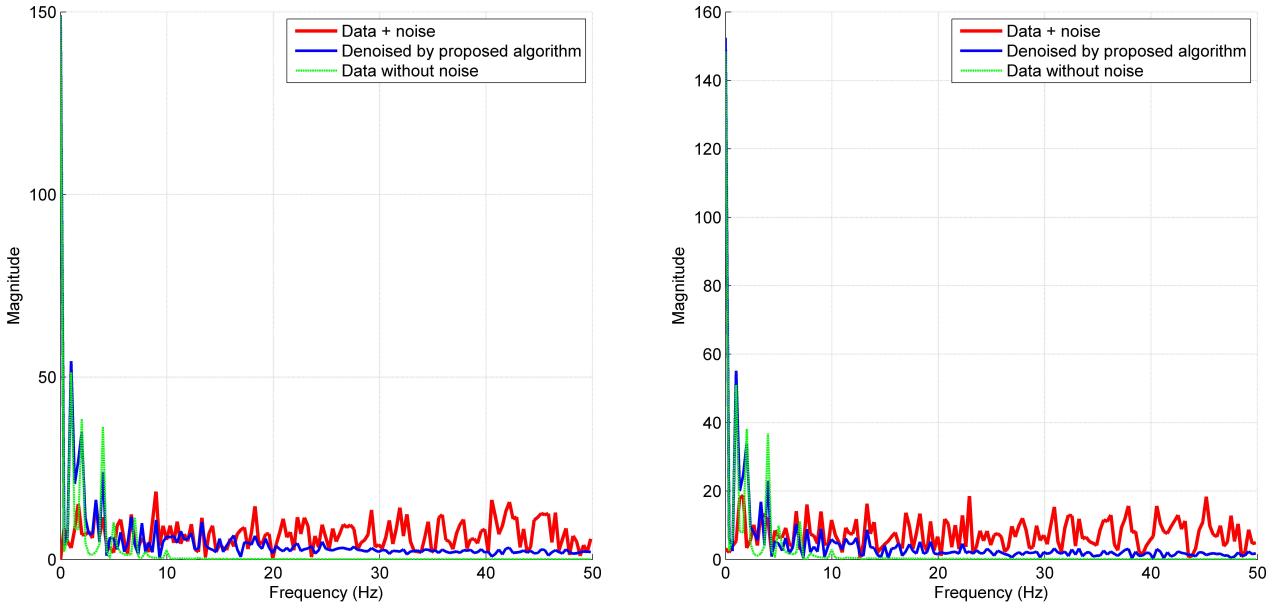
$$\hat{\xi}^{(n)}(t_q) = \sum_{u=(n-1)\alpha+1}^{n\alpha} \mathbf{d}_{u,q} \quad (46)$$

where $\mathbf{d}_{u,q}$ is the value at the u^{th} row and q^{th} column of \mathbf{D} .

2. Discussion

In contrast with denoising techniques such as low-pass filtering, this approach does not require the knowledge of the highest frequency of the signal being recorded. However, the knowledge of how much variance of the data to keep is integral to successful denoising — this is usually unknown and techniques have been proposed to solve this non-trivial problem [].

To demonstrate the potential of this procedure in removing correlated Gaussian noise, two frequency modulated signals that are phase shifted from each other are adulterated with correlated Gaussian noise. The original signals are then reconstructed using the proposed denoising algorithm with a threshold of 99 percent — that is, eigenvectors that account for closest to (but, less than) 99 percent of the variation of the data obtained from the set of multichannel intrinsic mode functions are retained. In Figure 8a, it can be easily observed that the algorithm extinguished an appreciable amount of the high frequency component that was not previously present in the data and selectively reduced the magnitude of the Fourier transform of the noise components with low frequency. Figure 8b was no different in terms of its denoising performance. It is worth noting that both results do have some discrepancy — and their discrepancies are not the same. This feature is due to the fact that, although the noise sources are correlated, the noises are different.



(a) Fourier transform of the proposed denoising approach for channel 1. (b) Fourier transform of the proposed denoising approach for channel 2.

FIG. 3: Output of the proposed denoising approach on two channels with signals of similar frequency modulation, but shifted in time. This procedure assumes the signal being recorded is from the same source and the noise observed in each channel are correlated.

A fundamental inquiry that needs a resolution is about how the proposed algorithm responds to different SNRs. To address this, the deviations of the Fourier transforms of reconstructed signals as a function of a myriad of SNRs were computed from the Fourier transform of the original signal. These deviation scores were measured as the mean-square errors between the reconstructed signals and the original signal. The threshold employed in this investigation was 99 percent. From Figure 9, it can be noted that the performance of the proposed algorithm increases with an increasing SNR; however, this increase in performance is limited. Since the proposed approach disregards parts of the data that are statistically irrelevant (with respect to the threshold chosen), it is plausible th

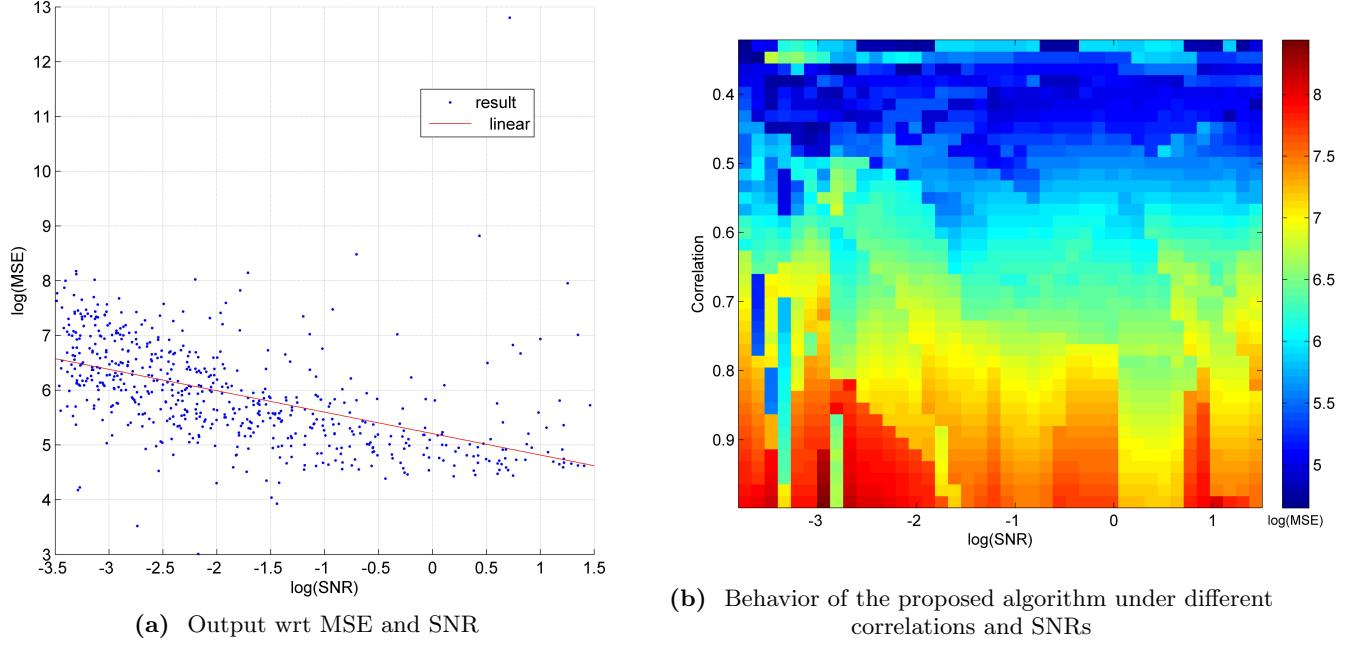


FIG. 4: Behavior of outputs

V. IMPLEMENTING THE GUSTAFSON ALGORITHM IN C++

A. Calculating Bessel Functions

Known problems with the algorithm include calculating large order Bessel functions. The Bessel functions of natural number order are given by

$$J_n(z) = \sum_{m=0}^{\infty} \frac{(-1)^n}{m!(m+n)!} \left(\frac{z}{2}\right)^{2m+n}. \quad (47)$$

Hence, calculating the n -th Bessel function involves calculating factorials greater than $n!$. The highest n for which we can store $n!$ in an unsigned long integer type is 20:

$$2^{64} - 1 = 18,446,744,073,709,551,615 \quad (48)$$

$$< 21! = 51,090,942,171,709,440,000. \quad (49)$$

If we are willing to sacrifice some precision to truncation, we can use a double precision floating point integer which allows values upto $1.797,693,134,862,315,7 \cdot 10^{308}$

$$170! \approx 7.25 \cdot 10^{306} < 1.797,693,134,862,315,7 \cdot 10^{308} < 171! \approx 1.24 \cdot 10^{309} \quad (50)$$

This is not acceptable however, because $|J_n(z)| \leq 1$. Therefore, when performing the sum we need precision all the way down to at the very least 0.1, which would require a mantissa with roughly 308 digits. This is not only

impractical, but it is by far one of the slowest ways one could calculate values of the Bessel functions. This whole business of wrestling with factorials can be avoided by calculating the Bessel Functions using Bessel's Equation 51 [7].

$$x^2 J_n''(x) + x J_n'(x) + (x^2 - n^2) J_n(x) = 0 \quad (51)$$

The standard prescription for numerically solving differential equations is first to discretize the independent variable $x \rightarrow x_i = x_{min} + i \cdot \delta x$, and then employ the limit definitions of the derivatives. We will use the three point definitions because their error goes as $\mathcal{O}(\delta x^2)$ rather than using the two point definition with an error that goes as $\mathcal{O}(\delta x)$ [?]. Further, because there are only one order of Bessel functions in Bessel's equation, the order is implied by the n that appears, so we will drop the subscript of n , $J_n \rightarrow J$, and adopt the notation $J(x_i) = J_i$.

$$J'_i = \frac{J_{i+1} - J_{i-1}}{2\delta x} \quad (52)$$

$$J''_i = \frac{J_{i-1} - 2J_i + J_{i+1}}{\delta x^2}. \quad (53)$$

Therefore the discretized form of Bessel's equation is

$$x_i^2 \frac{J_{i-1} - 2J_i + J_{i+1}}{\delta x^2} + x \frac{J_{i+1} - J_{i-1}}{2\delta x} + (x^2 - n^2) J_i = 0 \quad (54)$$

$$\left(\frac{x_i^2}{\delta x^2} - \frac{x_i}{2\delta x} \right) J_{i-1} + \left(\left(1 - \frac{2}{\delta x^2} \right) x_i^2 - n^2 \right) J_i + \left(\frac{x_i^2}{\delta x^2} + \frac{x_i}{2\delta x} \right) J_{i+1} = 0 \quad (55)$$

where we have collected like terms of J_i .

For convenience we define coefficients to simplify the above equation.

$$a_i = \frac{x_i^2}{\delta x^2} - \frac{x_i}{2\delta x} \quad (56)$$

$$b_i = \left(1 - \frac{2}{\delta x^2} \right) x_i^2 - n^2 \quad (57)$$

$$c_i = \frac{x_i^2}{\delta x^2} + \frac{x_i}{2\delta x} \quad (58)$$

These coefficients can be simplified by using the definition for x_i , and taking $x_{min} = 0$.

$$a_i = i \left(i - \frac{1}{2} \right) \quad (59)$$

$$b_i = i^2 (\delta x^2 - 2) - n^2 \quad (60)$$

$$c_i = i \left(i + \frac{1}{2} \right) \quad (61)$$

In effect, the discretized Bessel's equation with simplified coefficients is

$$a_i J_{i-1} + b_i J_i + c_i J_{i+1} = 0 \quad (62)$$

By writing out the system of equations explicitly, we have:

$$b_1 J_1 + c_1 J_2 = -a_1 J_0 \quad (63)$$

$$a_2 J_1 + b_2 J_2 + c_2 J_3 = 0 \quad (64)$$

$$a_3 J_2 + b_3 J_3 + c_3 J_4 = 0 \quad (65)$$

$$\vdots \quad (66)$$

$$a_{n-1} J_{n-2} + b_{n-1} J_{n-1} + c_{n-1} J_n = 0 \quad (67)$$

$$a_n J_{n-1} + b_n J_n = -c_n J_{n+1} \quad (68)$$

This suggests Bessel's equation can be written as a matrix equation:

$$\begin{pmatrix} b_1 & c_1 & & \\ a_2 & b_2 & c_2 & \\ a_3 & b_3 & c_3 & \\ \vdots & \vdots & & \\ a_{n-1} & b_{n-1} & c_{n-1} & \\ & a_n & b_n & \end{pmatrix} \begin{pmatrix} J_1 \\ J_2 \\ J_3 \\ \vdots \\ J_{n-1} \\ J_n \end{pmatrix} = \begin{pmatrix} -a_1 J_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -c_n J_{n+1} \end{pmatrix}. \quad (69)$$

This is a tridiagonal matrix system that characterizes a one dimensional differential equation. It can be easily solved using the standard Thomas Algorithm. To program this in C++ we need all the arrays to be indexed from 0. By virtue of this indexing system, we yield:

$$\begin{pmatrix} B_0 & C_0 & & \\ A_1 & B_1 & C_1 & \\ A_2 & B_2 & C_2 & \\ \vdots & \vdots & & \\ A_{N-4} & B_{N-4} & C_{N-4} & \\ & A_{N-3} & B_{N-3} & \end{pmatrix} \begin{pmatrix} J_1 \\ J_2 \\ J_3 \\ \vdots \\ J_{N-3} \\ J_{N-2} \end{pmatrix} = \begin{pmatrix} -A_0 J_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -C_{N-3} J_{N-1} \end{pmatrix} \quad (70)$$

$$A_i = a_{i+1} = (i + 1) \left(i + \frac{1}{2} \right) \quad (71)$$

$$B_i = b_{i+1} = (i + 1)^2 (\delta x^2 - 2) - n^2 \quad (72)$$

$$C_i = c_{i+1} = (i + 1) \left(i + \frac{3}{2} \right) \quad (73)$$

1. The Thomas Algorithm

The Thomas Algorithm is a lightning fast way to solve tridiagonal systems like equation 70. The algorithm consists of two steps. First there is a forward substitution which eliminates the A_i 's, and modifies the B_i 's and the vector on the right which we will call the source vector S_i ; $\forall i \in [0, N - 3]$.

$$R_i = R_i - \frac{A_i}{B_{i-1}} R_{i-1}; \quad \forall i \in [1, N - 3] \quad (74)$$

$$B_i \rightarrow B_i - \frac{A_i}{B_{i-1}} C_{i-1}; \quad \forall i \in [1, N - 3] \quad (75)$$

$$S_i \rightarrow S_i - \frac{A_i}{B_{i-1}} S_{i-1}; \quad \forall i \in [1, N - 3] \quad (76)$$

Then there is a backwards substitution eliminating the C_i 's, again modifying the S_i 's. Finally we solve for J_i .

$$R_i = R_i - \frac{C_i}{B_{i+1}} R_{i+1}; \quad \forall i \in [N - 4, 0] \quad (77)$$

$$S_i \rightarrow S_i - \frac{C_i}{B_{i+1}} S_{i+1}; \quad \forall i \in [N - 4, 0] \quad (78)$$

$$J_i = \frac{S_{i-1}}{B_{i-1}}; \quad \forall i \in [N - 2, 1] \quad (79)$$

The time it takes to perform the Thomas Algorithm goes as N [].

2. Boundary Conditions

To solve this second-order differential equation we will need two boundary conditions. The boundary condition at $x = 0$ is simple:

$$J_n(0) = \begin{cases} 1 & : n = 0 \\ 0 & : n \neq 0 \end{cases}$$

For the other boundary condition we *could* use the asymptotic form of the Bessel functions. Theorem 5.1 in [7] states:

For each $n \in N$ there is a constant $C_n \in R$ such that, if $x \geq 1$, then

$$\left| J_n(x) - \sqrt{\frac{2}{\pi x}} \cos \left(x - \frac{\pi}{4}(2n+1) \right) \right| \leq \frac{C_n}{x^{3/2}}. \quad (80)$$

This is an asymptotic expansion, so it is only valid for large values of x . It turns out the accuracy of the solution to Bessel's Equation relies heavily on the accuracy of the endpoint at $x \neq 0$. Figures ?? and ?? illustrate this point quite well. The conclusion is that the asymptotic form of the Bessel functions is not accurate enough for even fairly large values of z . We can also do

$$J_n(z) = \frac{1}{\pi} \int_0^\pi \cos [z \sin \theta - n\theta] d\theta. \quad (81)$$

Which is computed using a Riemann sum:

$$J_n(z) \approx \frac{1}{N} \sum_{i=0}^{N+1} \cos [z \sin(i \cdot dx) - n \cdot i \cdot dx]; \quad dx = \frac{\pi}{N} \quad (82)$$

The function being integrated becomes more oscillatory as n increases. Numerical analysis shows that n gives the number of zeros the function has on the interval $(0, \pi)$ for $z = 0$. For $n = 0$ the number of zeros is given by $\lfloor \frac{2z}{\pi} \rfloor$. We want a way to ensure the calculation is accurate regardless of how oscillatory it is, but we also want it to be as fast as possible. So, we use these facts regarding the zeros to ensure there are roughly the same number of integration points between zeros. By taking the number of (evenly spaced) integration points to be

$$N = 100 \cdot (n + 1) \cdot \left\lfloor \frac{2z}{\pi} + 1 \right\rfloor, \quad (83)$$

we achieved excellent results (accurate consistently to 16 decimal places) for a wide range of z and n . The $+1$'s are to ensure accurate results when z or n are zero. The algorithm can have problems when J is extremely small. Then the function can be off by many orders of magnitude, but because it is essentially zero, it does not seem to effect our results when used as a boundary condition for solving Bessel's Differential Equation.

3. Calculating Bessel Functions: Results

The ultimate benchmark for success in calculating Bessel Functions in this manner (using the Thomas Algorithm to solve Bessel's Differential Equation) is to compare it to the standard way of calculating the value at each point using the Riemann Sum (as defined in equation 82 section V A 2). So how much faster is it? This depends on the values of n and z because we use a dynamic number of integration points as given by eq. 83. Table I gives the speed increase for various parameters.

Parameters	Point by Point Using Integral	Thomas Algo. Solving Bessel's Equation	Speed Increase
$n = 3$ $N = 500$ $z_{max} = 20.0$	0.051786 s	0.00013800 s	375
$n = 14$ $N = 50,000$ $z_{max} = 100.0$	113.33 s	0.003937 s	28,785

TABLE I: N is the number of points returned, n is the order of the Bessel function, and the points calculated were for $z \in [0, z_{max}]$

Figure 6 provides three examples of Bessel Functions that were calculated using the Thomas Algorithm to solve Bessel's Equation.

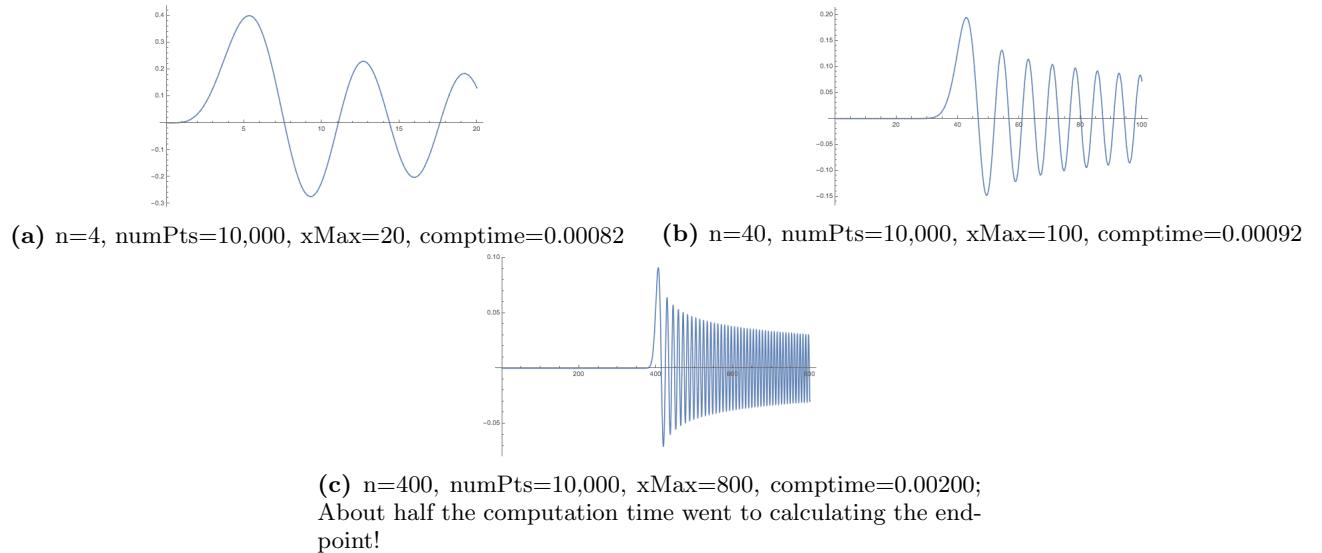


FIG. 6

In Figure ??, it can be observed that the standard approach and the suggested approach are in agreement except at $n = 0, 85$. This is a small subset of the 400,000 data points generated in 0.33 seconds (on 5 year old a laptop). It solved the Bessel differential equation for 4000 steps in $\Gamma \in (0, 100)$, then saved the output, and moved on to solve Bessel's equation for the next value of n until all values of Γ and n were solved for.

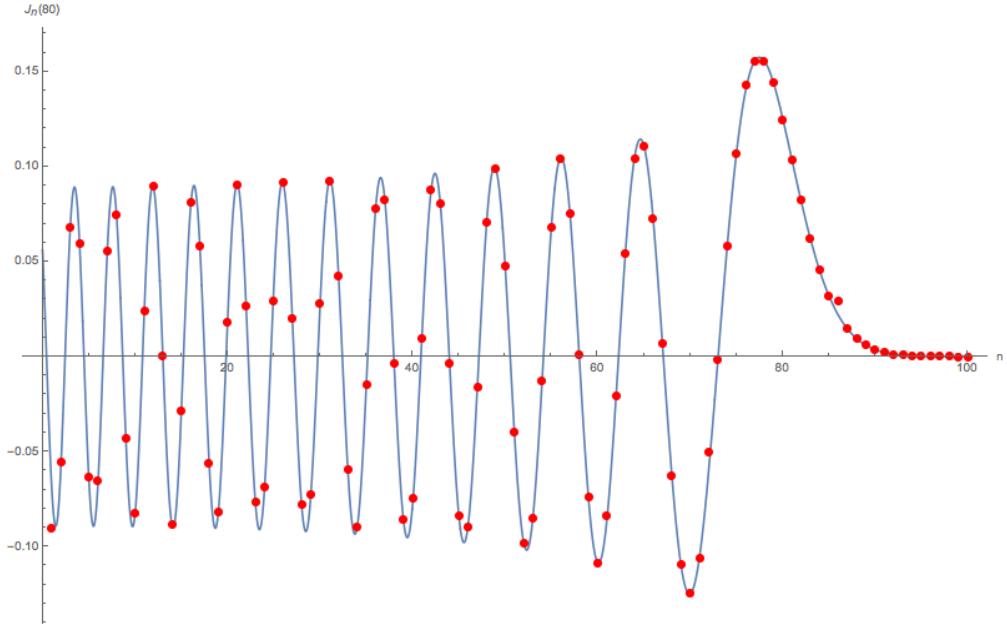


FIG. 7: This plot is a plot of $J_n(80)$ for values of $n \in (0, 100)$. The blue curve was generated by the BesselJ function of Mathematica. The red dots were calculated by our code. The two sets of data are in excellent agreement at except perhaps $n = 0, 85$ in which case there is a slight disagreement. This is a small selection of the 400,000 data points my code generated in 0.33 seconds (on 5 year old a laptop). It solved the Bessel differential equation for 4000 steps in $\Gamma \in (0, 100)$, then saves the output, and moved on to solve Bessel's Equation for the next value of n until all values of Γ and n were solved for.

VI. CARSON'S RULE AND THE NYQUIST FREQUENCY

In order too fully reconstruct the carrier frequency we would need to take an infinite sum over the sidebands. As this is impractical and unnecessary as many terms are insignificant, we need to find a way to limit the number of sidebands we are summing over. The first problem is locating the sidebands' characteristic frequency. There are infinite sideband frequencies and eventually they will be higher than the Nyquist frequency which will cause aliasing of the sideband frequency. The Nyquist frequency, ψ_N , is defined as the maximum frequency that can be accurately reconstructed and is equal to half of the sampling frequency, ψ_s .

$$\psi_N = \frac{1}{2}\psi_s \quad (84)$$

Any side band with a frequency of ψ_0 higher than the Nyquist frequency, but less than the sampling frequency, will be reconstructed with a frequency of $(\psi_s - \psi_0)$. That is the frequency will be under-sampled and an alias frequency will be produced. When this occurs the best case scenario is that the alias frequency is unique and does not correspond to the frequency of another lower frequency sideband. In this case we can use anti-aliasing techniques to correct for the aliasing and still obtain information about the amplitude and frequency of the higher side bands. The worst case scenario is that the aliased frequency is too close to another lower frequency sideband and this will cause two problems: a systematic error will be added to the measurement of the lower side band's amplitude, and we will be unable to determine the necessary information on what the amplitude of the higher frequency would be. To minimize the effect this might have, we would like any sidebands with a frequency over the Nyquist frequency to have as low an amplitude as possible. For this we can employ Carson's rule. Carson's rule can be understood to say that almost all (roughly 98 percent) of the power for a frequency-modulated sinusoidal signal is contained within a finite bandwidth B_T , defined by:

$$B_T = 2(\Delta\psi + \psi_m) \quad (85)$$

where $\Delta\psi$ is the peak frequency deviation of the instantaneous frequency $\psi(t)$ from the center carrier frequency ψ_c , and ψ_m is the highest frequency in the modulating signal. In our case the highest frequency of the modulating

signal is the carrier frequency, ψ_c . From this calculation we can adjust our sampling frequency to ensure it is large enough in order to minimize aliasing. This also puts a limit on the number of sidebands we need to sum over to just those that fall within the bandwidth. The important thing to take away from this is that it takes an infinite bandwidth to perfectly transmit a phase or frequency modulated wave, regardless of how smooth it is, but in practice only a finite bandwidth is needed for an accurate approximation.

VII. ERROR ANALYSIS

While the search for monochromatic gravitational waves obtained via the LIGO detector is relatively effortless, the adulteration of signals obtained from the detector by noise adds an appreciable number of complications to the problem. By the same token, techniques by which improved resistance of the LIGO detector to noise can be obtained has been investigated vigorously [2] [1].

A. Noise Sources

Since the signal being searched for is relatively stable but of little amplitude, the reliability of data obtained from the detector is related to the degree of noise therein. Among others, quantum, thermal, seismic and Newtonian noise have been implicated as prime drivers of LIGO data adulteration [14]. Since the prime noise contributor is quantum, a summation of all the noise sources are approximately Gaussian. Thus, it suffices to use the properties of Gaussian noise to understand the volatility of the data.

B. Noise Elimination

As mentioned previously, quantum noise — in particular, thermal noise — has been implicated as a limiting factor in the sensitivity of gravitational wave interferometers and hence a prime factor in the degradation of the signal quality; this noise source is generally modeled as Gaussian [14].

By low-pass filtering or spectral subtraction, it is possible to derive signals with high SNR; however, these widely-accepted approaches require information which can be difficult to obtain. Since the frequency of the monochromatic wave being searched for is not known, low-pass filtering may result in erroneous outputs. Spectral subtraction requires a sample of the noise; nevertheless, due to the low SNR of LIGO data, it is precarious to search for segments with pure noise.

In this section, we introduce an alternate approach by which Gaussian noise in data may be reduced. Our methodology consists of subtracting estimated noise from the signal via singular value decomposition of the Fourier transform of intrinsic mode functions (see appendix) and assumes the noise is correlated across different channels. A significant part of this approach is motivated by [11] and [3]. It is worth noting that because correlated Gaussian noise is prevalent, this approach may be used for a host of other signals.

1. Proposed Denoising Algorithm

In an effort to reduce the noise, we shall consider data from two channels. By the assumption that the noise observed is additive (and correlated), we yield the following:

$$\xi^{(1)}(t) = f^{(1)}(t) + \epsilon^{(1)}(t) \quad (86)$$

$$\xi^{(2)}(t) = f^{(2)}(t) + \epsilon^{(2)}(t) \quad (87)$$

where $\xi^{(n)}(t)$ is the data observed, $f^{(n)}(t)$ is the true signal and $\epsilon^{(n)}(t)$ is the noise obtained from the n-th channel (or gravitational wave interferometer) with $\epsilon^{(n)}(t) \sim \mathcal{N}(0, \sigma^2)$.

First, we disintegrate each observed signal into α intrinsic mode functions via empirical mode decomposition (see appendix) and arrange them into a $2\alpha \times T$ matrix \mathbf{L} :

$$\mathbf{L} = \begin{pmatrix} l_{1,t_1}^{(1)} & l_{1,t_2}^{(1)} & \dots & l_{1,t_T}^{(1)} \\ l_{2,t_2}^{(1)} & l_{2,t_2}^{(1)} & \dots & l_{2,t_T}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ l_{\alpha,t_1}^{(1)} & l_{\alpha,t_2}^{(1)} & \dots & l_{\alpha,t_T}^{(1)} \\ l_{\alpha+1,t_1}^{(2)} & l_{\alpha+1,t_2}^{(2)} & \dots & l_{\alpha+1,t_T}^{(2)} \\ l_{\alpha+2,t_2}^{(2)} & l_{\alpha+2,t_2}^{(2)} & \dots & l_{\alpha+2,t_T}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ l_{2\alpha,t_1}^{(2)} & l_{2\alpha,t_2}^{(2)} & \dots & l_{2\alpha,t_T}^{(2)} \end{pmatrix} \quad (88)$$

where $l_{p,t_q}^{(n)}$ is the $\left[\left((p-1) \bmod \alpha \right) + 1 \right]$ -th intrinsic mode function at time t_q of $\xi^{(n)}(t)$, and with T being the total number of data points observed.

Thereafter, the fast Fourier transform of each intrinsic mode function in \mathbf{L} is computed to obtain a matrix $\mathcal{F}(\mathbf{L})$. Singular value decomposition of the variance-covariance matrix Σ obtained from $\mathcal{F}(\mathbf{L})$ is then computed to result in the following:

$$\Sigma = \mathbf{U}\mathbf{S}\mathbf{V} \quad (89)$$

where $\Sigma = T^{-1} [\mathcal{F}(\mathbf{L})^T - \mathbf{1}\mathbf{1}^T \mathcal{F}(\mathbf{L})^T T^{-1}]^\dagger [\mathcal{F}(\mathbf{L})^T - \mathbf{1}\mathbf{1}^T \mathcal{F}(\mathbf{L})^T T^{-1}]$ with $(\cdot)^T$ being the transpose, the conjugate transpose $(\cdot)^\dagger$ and $\mathbf{1}$ a $T \times 1$ column matrix.

The fast Fourier transforms are then decorrelated to form a matrix \mathbf{R} by employing the following:

$$\mathbf{R} = \mathbf{U}^T \mathbf{L}. \quad (90)$$

After the decorrelation procedure above, we seek to apply a threshold Δ which dictates the approximate percentage of statistically irrelevant information about the Fourier transform we shall discard. Prior to the thresholding procedure, the eigenvectors that contribute to approximately Δ percent of the variance of the transform by virtue of their respective eigenvalues is estimated using the following formulation:

$$y = \operatorname{argmin}_b \left(\left| \frac{\sum_{j=1}^{(b)} \lambda_j}{\operatorname{trace}\{\mathbf{S}\}} - \frac{\Delta}{100} \right| \right) \quad (91)$$

where y is the index denoting the y^{th} eigenvalue from λ_y with $\lambda = \operatorname{diag}(\mathbf{S})$. It is essential to note that the eigenvalues are ordered from the highest to the lowest; the eigenvectors are also ordered by the position of their corresponding eigenvalues.

To apply the threshold, the projections with eigenvalues whose indices are less than y are retained and the rest are replaced with zero row vectors:

$$\mathbf{R}_r = \begin{cases} \mathbf{R}_r & , r < y \\ \mathbf{0} & , r \geq y \end{cases}$$

where \mathbf{R}_r the r^{th} row of the matrix \mathbf{R} . This preserves the vectors that explain closest to, but less than, Δ percent of the total variance.

The fast Fourier transforms are then recorrelated to obtain denoised Fourier transforms of intrinsic mode functions in the following manner:

$$\tilde{\mathbf{L}} = \mathbf{V}\mathbf{R}. \quad (92)$$

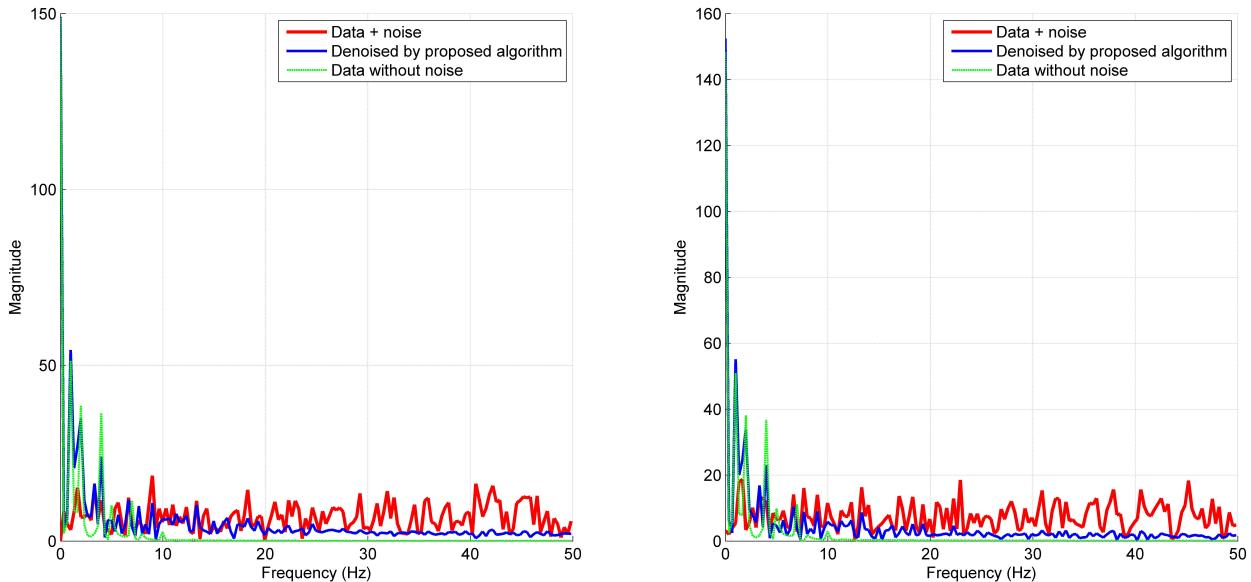
The real component of the inverse fast Fourier transform of each row of $\tilde{\mathbf{L}}$ is then computed to obtain a matrix \mathbf{D} of similar architecture with denoised intrinsic mode functions. Thereafter, the reconstructed signals $\hat{\xi}^{(n)}(t)$ are obtained as follows:

$$\hat{\xi}^{(n)}(t_q) = \sum_{u=(n-1)\alpha+1}^{n\alpha} \mathbf{d}_{u,q} \quad (93)$$

where $\mathbf{d}_{u,q}$ is the value at the u^{th} row and q^{th} column of \mathbf{D} .

2. Discussion

To demonstrate the potential of this procedure in removing correlated Gaussian noise, two frequency modulated signals that are phase shifted from each other are adulterated with correlated Gaussian noise. The original signals are then reconstructed using the proposed denoising algorithm with a threshold of 99 percent — that is, eigenvectors that account for closest to (but, less than) 99 percent of the variation of the data obtained from the set of multichannel intrinsic mode functions are retained. In Figure 8a, it can be easily observed that the algorithm extinguished an appreciable amount of the high frequency component that was not previously present in the data and selectively reduced the magnitude of the Fourier transform of the noise components with low frequency. Figure 8b was no different in terms of its denoising performance. It is worth noting that both results do have some discrepancy — and their discrepancies are not the same. This feature is due to the fact that, although the noise sources are correlated, the noises are different.



(a) Fourier transform of the proposed denoising approach for channel 1.
(b) Fourier transform of the proposed denoising approach for channel 2.

FIG. 8: Output of the proposed denoising approach on two channels with signals of similar frequency modulation, but shifted in time. This procedure assumes the signal being recorded is from the same source and the noise observed in each channel are correlated.

A fundamental question that needs a resolution is about how the proposed algorithm responds to different SNRs. To address this, the deviations of the Fourier transforms of reconstructed signals as a function of a myriad of SNRs were computed from the Fourier transform of the original signal. These deviation scores were measured as the mean squared errors between the reconstructed signals and the original signal. The threshold employed in this investigation was 99 percent. From Figure 9, it can be noted that the performance of the proposed algorithm increases with an increasing SNR; however, this increase in performance is limited. Since the proposed approach disregards parts of the data that are statistically irrelevant (with respect to the threshold chosen), it is plausible that parts of the original data may have been erased. This suggests that an adaptive threshold that depends on the structure of the data and the structure of the noise to be extinguished may be most appropriate for this algorithm.

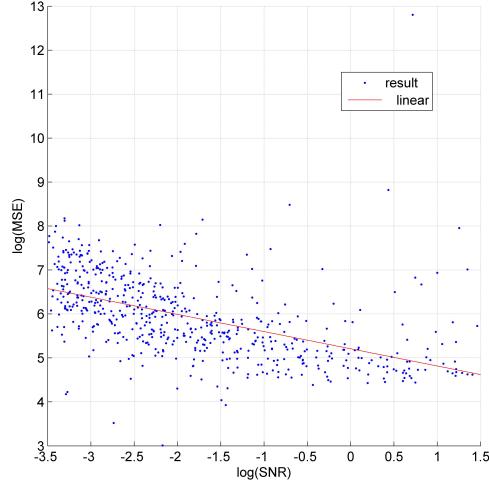
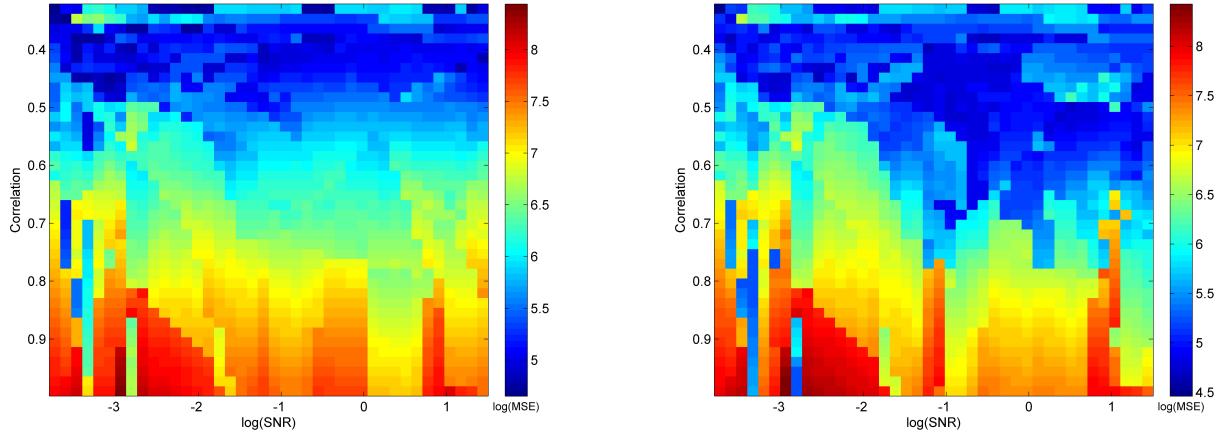


FIG. 9: The algorithm performs better with increasing SNR; however, this performance plateaus at high SNRs. The correlation coefficient $r^2 = -0.6543$ with $p < 0.05$.

Since a fundamental assumption the proposed algorithm makes is that the noise sources are correlated, it is important to investigate how the correlation affects its performance. To this end, a procedure similar to that of Figure 9 was implemented; nevertheless, a second independent variable was added — the correlation between the noise sources. The performance for denoising in each channel was not analyzed jointly. From Figure 10a, it can be noticed — as insinuated before — that the algorithm’s performance plateaus at high SNRs and progressively worsens as the same threshold is more likely to remove parts of the original data. Interestingly, the proposed approach works well with noise sources of tolerably high correlation; however, it fails to produce quality results when the correlation is really high.



(a) Response of the algorithm with changing noise correlations and SNRs for channel 1. (b) Response of the algorithm with changing noise correlations and SNRs for channel 2.

FIG. 10: Behavior of the proposed algorithm with respect to differing SNRs and noise correlations. This figure provides an insight to the performance of the proposed denoising algorithm (via the mean squared error) under the outlined conditions.

The potential of this approach has been outlined above. With a few modifications, this denoising algorithm may improve its capability and, to some extent (depending on the structure of the data and the noise), could be modified to deal with uncorrelated noise. In the spirit of demonstrating the robustness of the Gustafson algorithm against noise, the results from the search algorithm will be analyzed with no attempt to remove noise.

VIII. RESULTS AND DISCUSSION

Since frequency modulated signals may sometimes look like noise in the time domain (see appendix), it is imperative to ascertain that the Gustafson algorithm provides accurate information about the carrier frequency. For this purpose, the Gustafson algorithm was designed to be implemented in Fourier space.

By observing Figures 11a and 11b, we can see that noise added to the frequency modulated signal makes it almost impossible to visually characterize. In practice, most frequency modulated signals encountered — especially those of astronomical sources and signal susceptible to environmental noise — are of this form. A simple approach to identifying relevant features of the frequency modulated signal adulterated with noise will be to search for patterns therein. However, since the noise is sampled randomly it is nearly impossible to extract patterns without some form of transformation. The most obvious transform to employ that allows us access the frequencies that compose the noisy signal is its Fourier transform.

In Figures 11c and 11d the Fourier transform of the frequency modulated signal and that of the signal adulterated with noise are shown respectively. In the space of the Fourier transform, the frequencies that compose the original signal can be clearly observed. However, this task gets challenging when observing the Fourier transform of the noisy signal. Note that although a high degree of noise has been added, major peaks can still be seen. This is a function of the SNR; with a high enough noise amplitude, the Fourier transform will fail in providing useful information. This suggests that although the Fourier transform is a powerful technique when it comes to ignoring noise within a continuous signal, it is limited by the power of the noise.

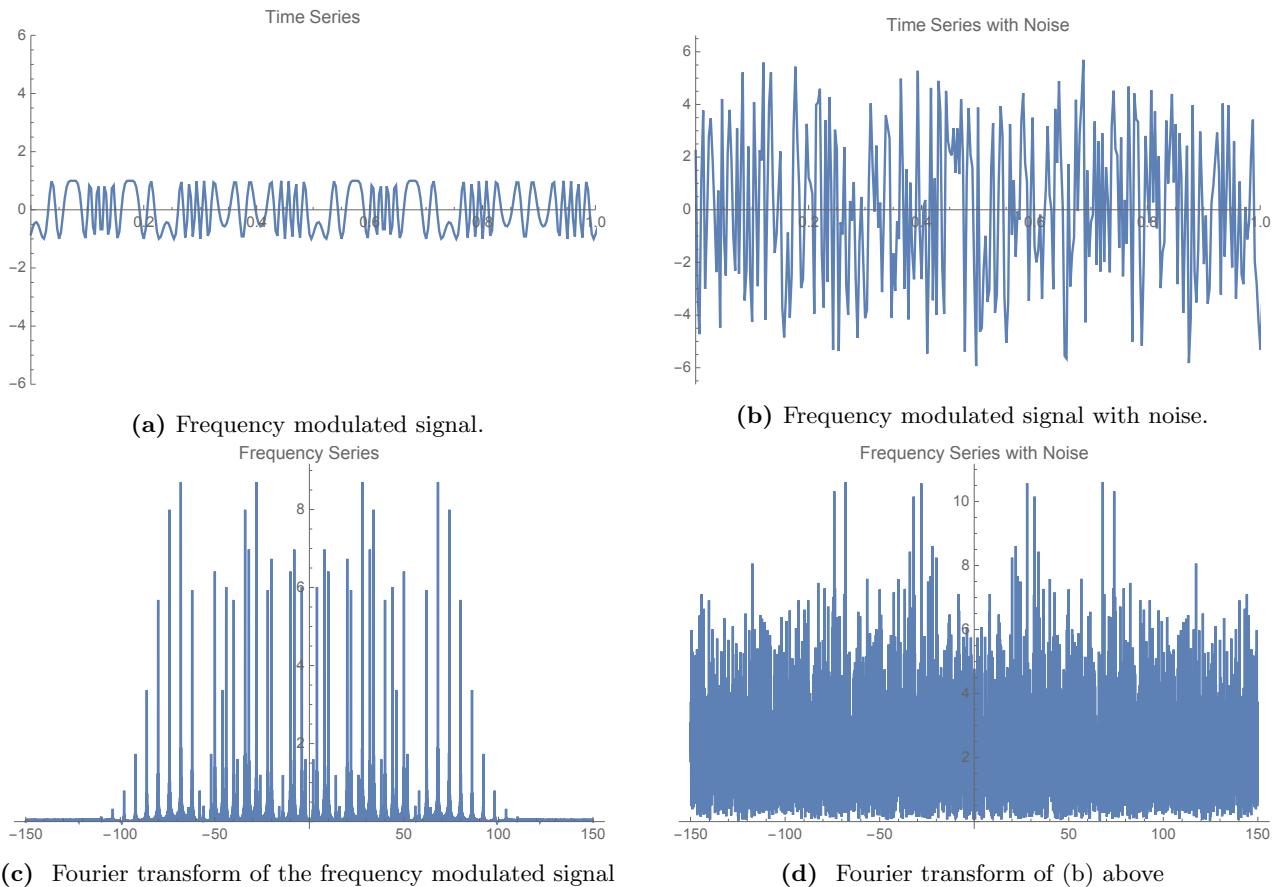


FIG. 11: Frequency modulated signal and its noisy version in the time series and Fourier space. The sampling rate for these traces was 300Hz. The carrier frequency was 20Hz, while the modulation frequency was 6Hz. It is worth noting that these represent 10 seconds of recording with a modulation index of 10.

To test the robustness of the Gustafson algorithm against noise, we searched for the carrier frequency, modulation frequency, modulation index and the recipient's angular frequency for they noisy signal provided in Figure 11b. From Figure 12a we can see that the Gustafson algorithm appropriately separated out the signal (carrier frequency) from the noise even though it was noise obvious in the corresponding Fourier transform. It is worth noting that the search results in a transform that is asymmetric in frequency; this is a function of the fact that the Gustafson algorithm

includes a rotation in Fourier space that is opposite to that of the Fourier transform. With Figure 12b, it can be observed that the Gustafson algorithm accurately obtained modulation frequency of the noisy signal. Since the search for the carrier frequency and modulation frequency is sensitive, it is important to use small steps; this will aid in preventing one from accidentally skipping over the correct solution. Like the previous results, the Gustafson algorithm was able to extract the correct recipient's angular frequency (Fig. 12c) and the modulation index (Fig. 12d) of the frequency modulated signal. Since the search for the modulation index and recipient's angular frequency is relatively smooth, one may take tolerably larger step sizes and still converge to the right solution.

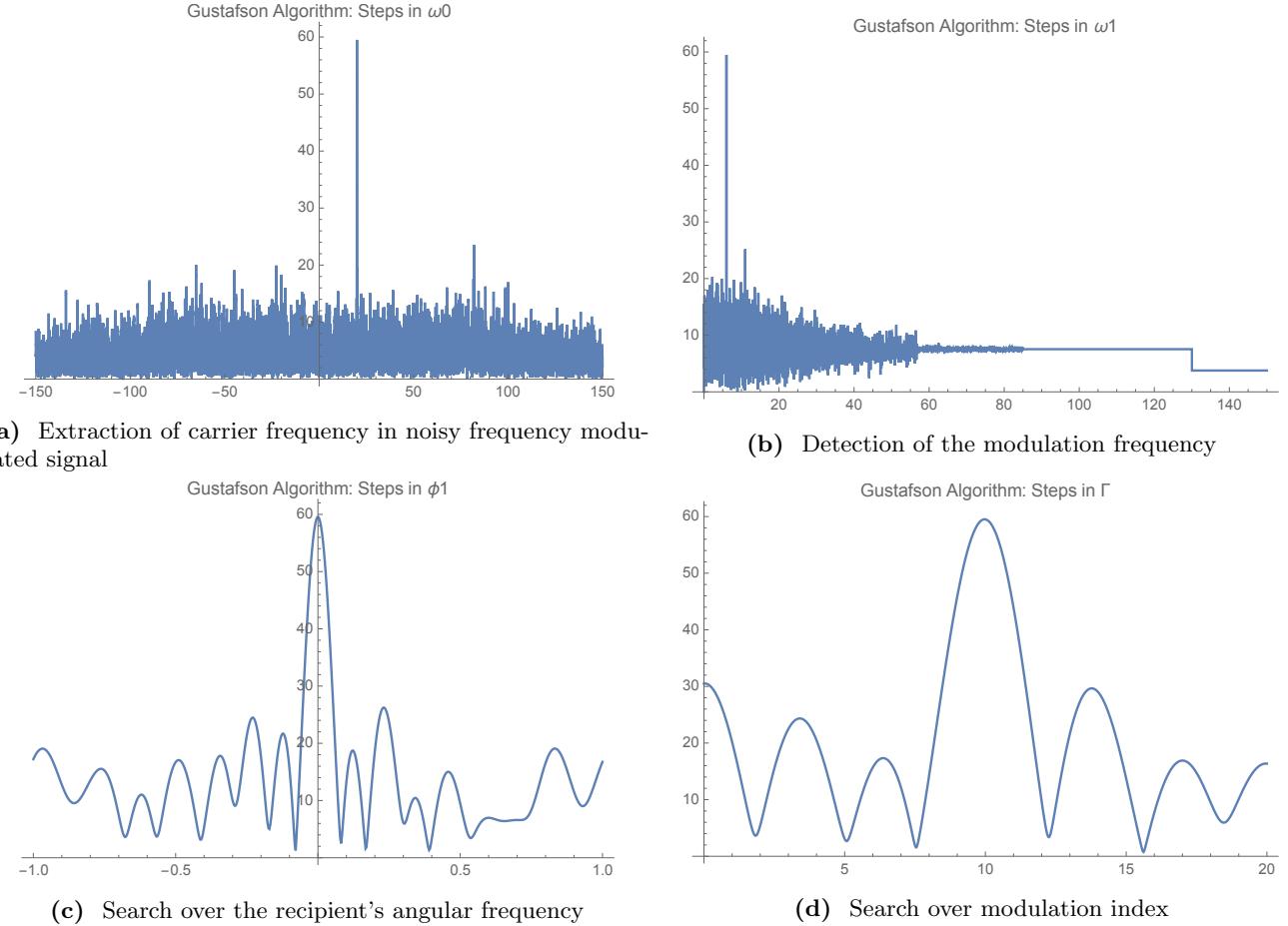


FIG. 12: Application of the Gustafson algorithm to the noisy frequency modulated signal illustrated in Fig. 11b. Sampling Rate = 300Hz, Carrier = 20Hz, Mod Frequency = 6Hz, Mod Index = 10, Time Series Duration = 10s, SNR = 0.1

IX. CONCLUSION

In this paper, the Gustafson algorithm has been introduced as well as a novel approach to noise reduction. Although both algorithms presented have been shown to be effective, they may require substantial modifications to be useful in practice. One major draw back of the Gustafson algorithm is the selection of a range of parameters to search over. An investigation of how one can introduce learning procedures such as stochastic gradient descent might be able to provide a steady solution to this issue.

Appendix A: The Fourier Transform of The Complex Function

The results this and the following section are quite surprising. Looking at the instantaneous frequency of a frequency modulated wave, it takes on every value between $\omega_0 \pm \omega_1$. There are an uncountable infinite number of frequencies in this band, but essentially every frequency in this band does not appear in the Fourier Transform! In

fact, there are only two or three of these frequencies present in the Fourier Transform. Moreover the bandwidth for a frequency modulated wave is infinite. A countably infinite number of frequencies called *sidebands* are evenly spaced at integer multiples of ω_1 from the *carrier frequency* ω_0 . Interestingly for particular values of the *modulation index* Γ it is possible for the carrier frequency to not be present in the Fourier Transform! This occurs at the zeros of the zeroth Bessel Function.

$$\mathcal{F}_t \left[h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \right] = h_0 e^{i\phi_0} \mathcal{F}_t \left[e^{i\omega_0 t} e^{i\Gamma \cos(\omega_1 t + \phi_1)} \right] \quad (\text{A1})$$

$$= h_0 e^{i\phi_0} \mathcal{F}_t \left[e^{i\omega_0 t} \right] * \mathcal{F}_t \left[e^{i\Gamma \cos(\omega_1 t + \phi_1)} \right] \quad (\text{A2})$$

$$= 2\pi h_0 e^{i\phi_0} \delta(\omega - \omega_0) * \mathcal{F}_t \left[\sum_{n=-\infty}^{\infty} e^{in\pi/2} J_n(\Gamma) e^{in(\omega_1 t + \phi_1)} \right] \quad (\text{A3})$$

$$= 2\pi h_0 e^{i\phi_0} \delta(\omega - \omega_0) * \sum_{n=-\infty}^{\infty} e^{in\pi/2} e^{in\phi_1} J_n(\Gamma) F_t \left[e^{in\omega_1 t} \right] \quad (\text{A4})$$

$$= 2\pi h_0 e^{i\phi_0} \delta(\omega - \omega_0) * \sum_{n=-\infty}^{\infty} e^{in(\phi_1 + \pi/2)} J_n(\Gamma) \delta(\omega - n\omega_1) \quad (\text{A5})$$

$$= 2\pi h_0 e^{i\phi_0} \sum_{n=-\infty}^{\infty} e^{in(\phi_1 + \pi/2)} J_n(\Gamma) \delta(\omega - \omega_0) * \delta(\omega - n\omega_1) \quad (\text{A6})$$

$$= 2\pi h_0 e^{i\phi_0} \sum_{n=-\infty}^{\infty} e^{in(\phi_1 + \pi/2)} J_n(\Gamma) \delta(\omega - \omega_0 - n\omega_1) \quad (\text{A7})$$

In this derivation we start by using the linearity of the Fourier Transform, then we invoke the Convolution Theorem, use the Jacobi-Anger Expansion, again use the linearity of the Fourier Transform, and the last few steps are simple Dirac Delta Function manipulations.

Appendix B: The Fourier Transform of The Real-Valued Function

$$\mathcal{F}_t \left[\Re \left\{ h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \right\} \right] = \frac{1}{2} \mathcal{F}_t \left[h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} + h_0 e^{-i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \right] \quad (\text{B1})$$

$$= \frac{h_0}{2} \left[e^{i\phi_0} \mathcal{F}_t \left[e^{i(\omega_0 t + \Gamma \cos(\omega_1 t + \phi_1))} \right] + e^{-i\phi_0} \mathcal{F}_t \left[e^{i(-\omega_0 t - \Gamma \cos(\omega_1 t + \phi_1))} \right] \right] \quad (\text{B2})$$

From the previous section we see that

$$\mathcal{F}_t \left[e^{i(\omega_0 t + \Gamma \cos(\omega_1 t + \phi_1))} \right] = \sum_{n=-\infty}^{\infty} e^{in(\phi_1 + \pi/2)} J_n(\Gamma) \delta(\omega - \omega_0 - n\omega_1) \quad (\text{B3})$$

and by replacing ω_0 with $-\omega_0$ and Γ with $-\Gamma$, and using the fact that $J_n(-\Gamma) = (-1)^n J_n(\Gamma)$ we also have

$$\mathcal{F}_t \left[e^{i(-\omega_0 t - \Gamma \cos(\omega_1 t + \phi_1))} \right] = \sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \pi/2)} J_n(\Gamma) \delta(\omega + \omega_0 - n\omega_1) \quad (\text{B4})$$

so the Fourier transform of the real part of our frequency/phase modulated signal is

$$\begin{aligned}
\mathcal{F}_t \left[\Re \left\{ h_0 e^{i(\omega_0 t + \phi_0 + \Gamma \cos(\omega_1 t + \phi_1))} \right\} \right] &= \frac{1}{2} h_0 \left[e^{i\phi_0} \sum_{n=-\infty}^{\infty} e^{in(\phi_1 + \pi/2)} J_n(\Gamma) \delta(\omega - \omega_0 - n\omega_1) \right. \\
&\quad \left. + e^{-i\phi_0} \sum_{n=-\infty}^{\infty} e^{in(\phi_1 - \pi/2)} J_n(\Gamma) \delta(\omega + \omega_0 - n\omega_1) \right] \\
&= \frac{1}{2} h_0 \sum_{n=-\infty}^{\infty} e^{in\phi_1} J_n(\Gamma) \left[e^{i(\phi_0 + n\pi/2)} \delta(\omega - \omega_0 - n\omega_1) \right. \\
&\quad \left. + e^{-i(\phi_0 + n\pi/2)} \delta(\omega + \omega_0 - n\omega_1) \right]
\end{aligned} \tag{B5}$$

$$\hat{f}(\omega) = \frac{1}{2} h_0 \sum_{n=-\infty}^{\infty} e^{in\phi_1} J_n(\Gamma) \left[e^{i(\phi_0 + n\pi/2)} \delta(\omega - \omega_0 - n\omega_1) + e^{-i(\phi_0 + n\pi/2)} \delta(\omega + \omega_0 - n\omega_1) \right] \tag{B6}$$

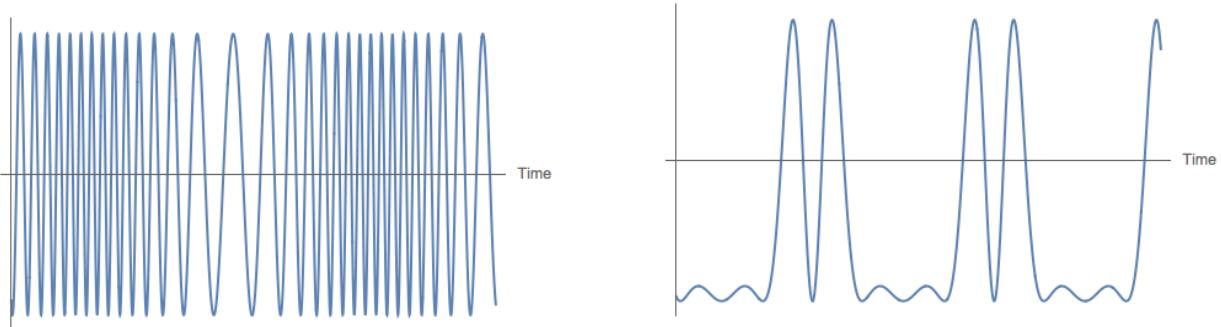
||||| HEAD Notice that the Fourier Transform of the real wave looks very similar to the Fourier Transform of the complex wave. The difference being that the Fourier Transform of the real is symmetric about $\omega = 0$ up to a phase factor. The vectors corresponding to the positive carrier frequency rotate counterclockwise, whereas the vectors corresponding to the negative carrier frequency rotate clockwise (neglecting the rotation due to ϕ_1 which is the same for both). **This phase factor is the reason why the Gustafson Algorithm only picks out the positive value of ω_0 .** If it is unclear what this means refer to figure (12e). ====== ::::: master

Appendix C: Intrinsic Mode Functions

Intrinsic mode functions are functions that are obtained by decomposing a signal. In general, intrinsic mode functions add up to the signal that was decomposed and are usually extracted by empirical mode decomposition [8].

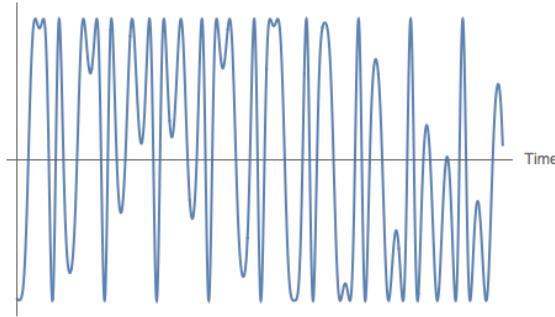
Classically, the extraction of intrinsic mode functions also yields a component named the residue — which is the back-bone of the signal. In this section, we shall assume the residue is one of the intrinsic mode functions for simplicity. Suppose we want to disintegrate a signal $f(t)$ into k intrinsic mode functions. We first find the local maxima and minima of $f(t)$. Thereafter, we interpolate the local maxima to create an upper envelope and interpolate the local minima to create a lower envelope via cubic splines. The mean of these envelopes is computed and subtracted from the input signal to result in the first intrinsic mode function. This intrinsic mode function is then subtracted from the input signal to result in the residue. This approach is repeated $k - 1$ times in total to obtain k intrinsic mode functions (where the last intrinsic mode function is the residue).

Appendix D: Selected Waveforms



(a) When the carrier frequency is much higher than the modulation frequency the frequency modulated wave is "nice".
 $\Gamma = 9.88, \omega_0 = 3.62, \omega_1 = 0.21, \phi_1 = 3.14$

(b) When the carrier frequency is a multiple of the modulation frequency one can get exotic waveforms. One example of the exotic signals cyclic frequency modulation can produce.
 $\Gamma = 2.88, \omega_0 = 1.5, \omega_1 = 0.75, \phi_1 = 0$



(c) When the carrier and modulation frequencies are almost equal the resulting FM wave is not so nice. This is a particularly nasty waveform that could be mistaken as noise by looking at it.
 $\Gamma = 3, \omega_0 = 0.42, \omega_1 = 0.46, \phi_1 = 0.1$

- [1] Junaid Aasi, J Abadie, BP Abbott, Richard Abbott, TD Abbott, MR Abernathy, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Enhanced sensitivity of the ligo gravitational wave detector by using squeezed states of light. *Nature Photonics*, 7(8):613–619, 2013.
- [2] A Abramovici, W Althouse, J Camp, D Durante, JA Giaime, A Gillespie, S Kawamura, A Kuhnert, T Lyons, FJ Raab, et al. Improved sensitivity in a gravitational wave interferometer and implications for ligo. *Physics Letters A*, 218(3):157–163, 1996.
- [3] Mina Aminghafari, Nathalie Cheze, and Jean-Michel Poggi. Multivariate denoising using wavelets and principal component analysis. *Computational Statistics & Data Analysis*, 50(9):2381–2398, 2006.
- [4] Albert Einstein. *On gravitational waves*. Springer, 1990.
- [5] Deanna Emery. A coherent three dimensional fft based search scheme for gravitational waves from binary neutron star systems. *LIGO DCC*, (T1500307-v2), September 2015.
- [6] B.P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration). Observation of gravitational waves from a binary black hole merger. *Physical Review Letters*, 116(061102):1–16, 12 February 2016.
- [7] Gerald B. Folland. *Fourier Analysis and its Applications*. Number ISBN 0-534-17097-3. Brooks and Cole Publishing Company, 1992.
- [8] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 903–995. The Royal Society, 1998.
- [9] Martin Kreh. Bessel functions. Project for the Penn State - Göttingen Summer School on Number Theory.
- [10] Nadja S Magalhães, Warren W Johnson, Carlos Frajula, and Odylio D Aguiar. Determination of astrophysical pa-

- rameters from the spherical gravitational wave detector data. *Monthly Notices of the Royal Astronomical Society*, 274(3):670–678, 1995.
- [11] Karim G Oweiss and David J Anderson. Noise reduction in multichannel neural recordings using a new array wavelet denoising algorithm. *Neurocomputing*, 38:1687–1693, 2001.
 - [12] Eric Poisson. Gravitational waves from inspiraling compact binaries: The quadrupole-moment term. *Physical Review D*, 57(8):5287, 1998.
 - [13] Charles Rader. Discrete fourier transforms when the number of data samples is prime. *Proceedings of the IEEE*, 56(6):1107–1108, 1968.
 - [14] Peter R. Saulson. *Fundamentals of Interferometric Gravitational Wave Detectors*. World Scientific Publishing Company, March 31, 1994.