A Coherent Three Dimensional FFT Based Search Scheme for Gravitational Waves from Binary Neutron Star Systems

Deanna Emery, Harvard University
Mentor: Dick Gustafson

September 25, 2015

Abstract

Gravitational waves (GW) from binary neutron stars are phase modulated due to a Doppler shift caused by the orbital motion of the stars. This phase modulated signal has an oscillating frequency. Finding the carrier frequency of a phase-modulated signal after applying the Fourier transform is non-trivial. This is due to the modulation factor of the signal, which induces sidebands that vary in magnitude relative to the carrier frequency depending on the size of the modulation index (the extent of the Doppler shift). A GW signal can be extracted from noise using a data search over three parameters: the carrier frequency (ranging from 10-1500 Hz), the modulation frequency (about 0.00001-1 Hz), and the modulation index/amplitude (0-10,000). Key strategies for this search include isolating the modulation factor from the signal and approximating it using the Jacobi-Anger expansion, working in the frequency domain (taking the Fourier Transform only once in the initial step), and utilizing the Convolution Theorem. The program for this search, written in MATLAB, has been able to determine all three parameters from simulated data both with and without noise, though with some limitations that depend on the number of terms used in the Jacobi-Anger expansion and the amount of noise. There are also variations of signals that remain to be addressed.

1 Introduction

Einstein's General Relativity predicts that gravitational wave signals are emitted from binary neutron star systems. Einstein's model for the loss of orbital energy of the neutron stars due to their emission of gravitational waves accurately matches observations. The Laser Interferometer Gravitational Wave Observatory (LIGO) searches for gravitational waves from such sources. Due to the rotation of these binary star systems, the signal is Doppler shifted,

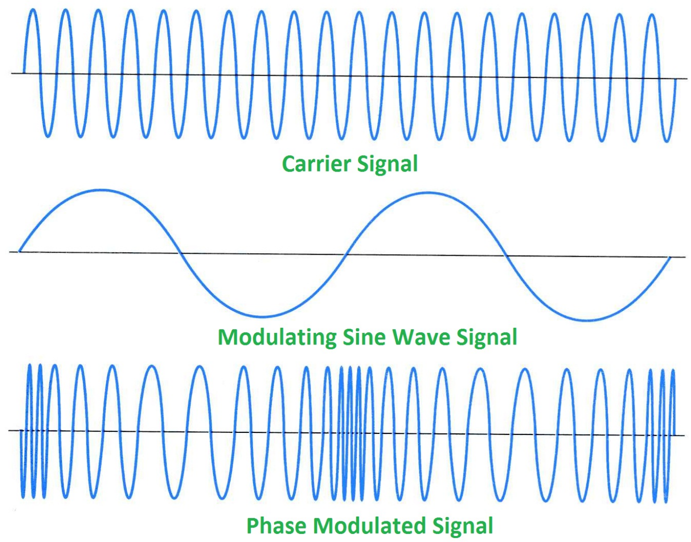


Figure 1: Phase Modulated Signal with Oscillating Frequency
Image Credit:
`http://ironbark.xtelco.com.au/subjects/DC/lectures/7/fig_2010_07_08.jpg`

causing its frequency to oscillate, or to warble. This oscillating frequency can be equivalently written as a modulation in phase. Figure 1 demonstrates an example of a phase modulated signal.

When searching for a gravitational wave in the data collected by LIGO, finding the carrier frequency (the frequency of the source) of a phase modulated signal in noise is not always straightforward. This is due to the sidebands in the Fourier transform, which are induced by the modulation factor of the signal. A phase modulated signal can be extracted from noise using a data search over three parameters which characterize the binary neutron star system: the carrier frequency (the frequency of the gravitational wave), the modulation frequency (the orbital frequency of the neutron stars), and the modulation index/amplitude (the extent of the Doppler shift on the signal).

I have developed a program in MATLAB that utilizes the Fast Fourier Transform (FFT) to perform a three dimensional search for gravitational wave sources from binary neutron stars. Key strategies for this search include isolating the modulation factor from the signal and approximating it using the Jacobi-Anger expansion, working in the frequency domain (taking the Fourier Transform only once in the initial step), and using the Convolution Theorem.

The program has been applied to simulated data that contain phase modulated signals in order to extract the frequency of signal. It was able to determine all three parameters from the simulated data both with and without noise, though its capabilities and accuracy are dependent on the number of terms used from the Jacobi-Anger expansion and the amount of noise in the signal. Additionally, other types of signals were tested on the program with varying results and accuracy.

2 Methods

2.1 The Signal

The expression for a warbling (phase modulated) signal, assuming circular motion, is given by:

$$d(t) = e^{i(\omega t - \Gamma \cos \Omega t)} \tag{1}$$

Where ω is the carrier frequency, Ω is the warble frequency (the frequency of orbit of the binary stars), and Γ is the modulation index, or warble amplitude. A phase modulated signal contains sidebands which occur at the frequencies $\omega \pm n\Omega$ (for integer n). This is made evident when the expression for the signal is rewritten using the Jacob-Anger expansion. The Jacobi-Anger expansion is given by:

$$e^{i\Gamma \cos \Omega t} = \sum_{-\infty}^{\infty} i^n J_n(\Gamma) e^{in\Omega t} \tag{2}$$

$$\approx J_0(\Gamma) + iJ_1(\Gamma)[e^{i\Omega t} + e^{-i\Omega t}] - J_2(\Gamma)[e^{i2\Omega t} + e^{-i2\Omega t}] + \cdots \tag{3}$$

Substituting this expansion into our expression for a phase modulated signal:

$$d(t) = e^{i(\omega t + \Gamma \cos \Omega t)} \tag{4}$$

$$= e^{i\omega t} e^{i\Gamma \cos \Omega t} \tag{5}$$

$$\approx e^{i\omega t}[J_0(\Gamma) + iJ_1(\Gamma)(e^{i\Omega t} + e^{-i\Omega t})] \tag{6}$$

$$\approx J_0(\Gamma)e^{i\omega t} + J_1(\Gamma)e^{i(\omega+\Omega)t} + J_1(\Gamma)e^{i(\omega-\Omega)t} \tag{7}$$

From here, it can be seen that a Fourier transform of a warbling signal given by equation 7, will show sidebands on either side of the carrier frequency. Figure 2 shows an example of the FFT of a signal with $\omega = 100$ Hz, $\Omega = 10$ Hz, and $\Gamma = 10$. The sidebands are located at multiples of 10 Hz away from the carrier frequency on either side. Also note that the size of the sidebands with respect to the size of the peak located at the carrier frequency is dependent on the value of Γ. In the case where Γ is greater than one, the magnitude of the sidebands is larger than that of the carrier.

2.2 The Theory of the Program

Recall the expression for a warbling signal:

$$d(t) = e^{i(\omega t - \Gamma \cos \Omega t)} \tag{8}$$

Which can be factored and reordered to get:

$$e^{i\Gamma \cos \Omega t} d(t) = e^{i\omega t} \tag{9}$$

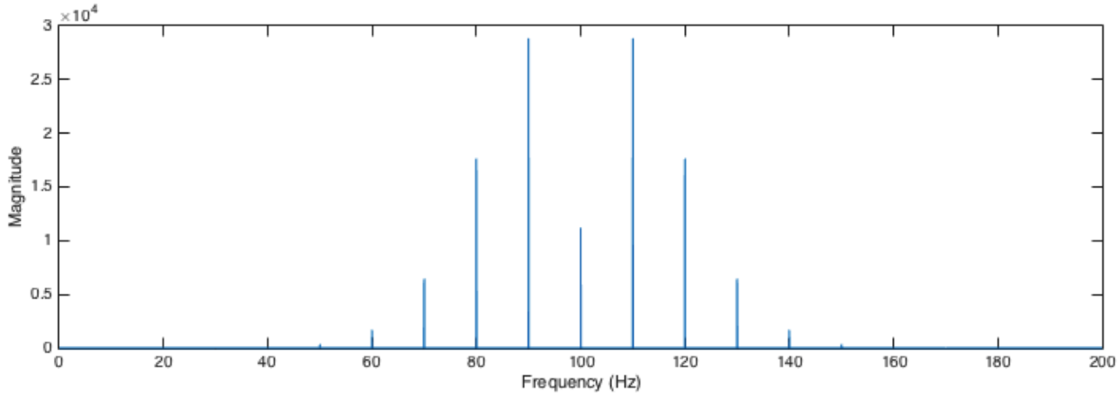


Figure 2: The FFT of a signal with $\omega = 100$ and $\Omega = 10$. Sidebands are located at multiples of Ω away from ω on either side.

We then apply the Fourier Transform on both sides, so that we are now in the frequency domain:

$$\mathcal{F}\left(e^{i\Gamma\cos\Omega t}d(t)\right) = \mathcal{F}(e^{i\omega t}) \tag{10}$$

From here, we can use the convolution theorem, which states that for any two functions $f(t)$ and $g(t)$, the Fourier transform of the product of these functions can be given as:

$$\mathcal{F}\left(f(t)g(t)\right) = \mathcal{F}\left(f(t)\right) * \mathcal{F}\left(g(t)\right) \tag{11}$$

Applying the convolution theorem to the expression for our signal, we find that:

$$\mathcal{F}(e^{i\Gamma\cos\Omega t}) * \mathcal{F}(d(t)) = \delta(\omega' - \omega) \tag{12}$$

We can further reduce this by using the Jacobi-Anger expansion to approximate the modulation factor of the signal, which enables the Fourier Transform to be done by hand.

$$\mathcal{F}\left(J_0(\Gamma) + iJ_1(\Gamma)[e^{i\Omega t} + e^{-i\Omega t}] + \cdots\right) * \mathcal{F}(d(t)) = \delta(\omega' - \omega) \tag{13}$$

$$[J_0(\Gamma) + iJ_1(\Gamma)\delta(\omega' - \Omega) + iJ_1(\Gamma)\delta(\omega' - \Omega) + \cdots] * \mathcal{F}(d(t)) = \delta(\omega' - \omega) \tag{14}$$

Now, if we imagine that $d(t)$ represents a data set of some sampled signal with unknown parameters, ω, Ω, and Γ, we can use the FFT and a search over Ω and Γ to find the carrier frequency of the gravitational wave. From Equation 14, we can see that the Fourier transform of the data convolved with the Fourier transform of the modulation factor (given the correct values of ω, Ω, and Γ) should output a single delta function with a peak located at the carrier frequency.

2.3 The Program

The program applies the concepts discussed in the previous section and creates a graphical image of the expected peak; this peak determines the correct values of the search over the three parameters, namely the carrier frequency, the modulation frequency, and the modulation index.

The MATLAB program that performs this search samples a signal with known parameters with the objective of extracting the expected values of these parameters. The first step in the program is to take the FFT of the data set. An approximation is then made for the Fourier Transform of the warbling factor of the signal using the Jacobi-Anger expansion and its resulting sidebands (note that the Jacobi-Anger expansion is dependent on the unknown variables Γ and Ω). Next, searching over Γ and Ω in increments, this approximation is convolved with the FFT of the data using a brute force method. Since we expect to find a peak located at the correct values of the three parameters, the program searches through the results of the convolutions for each increment of the search and finds the maximum value. It reports the values of Γ, Ω and ω associated with this maximum. It then generates a three dimensional plot for the specific value of Γ that attains this maximum, and plots the steps in Ω on the x-axis, the steps in ω on the y-axis, and magnitude on the z-axis.

Additionally, there have been some variations of the program created that work for specific cases, such as when there is more than one phase modulated signal or when there is both a phase modulated signal and a regular, non-modulated signal.

2.4 The Intervals of the Seach Parameters

The intervals of search were determined for each parameter in order to decrease the time spent running the search and to tailor the program to the specific case of gravitational wave detection at LIGO. A full search over the determined intervals for the three parameters has not yet been done due to limitations of computational power and time.

LIGO is sensitive to gravitational waves within the frequency range of 50 Hz to 1500 Hz. Ideally, we would want to take steps of 0.001 Hz, making the total number of steps made in the ω search equal to approximately one million.

The modulation frequency is the orbital frequency of the binary neutron stars. Searching through a database of binary systems revealed that these stars could have orbital frequencies ranging from 0.1 Hz to 10^{-6} Hz. A more constrained search interval or more efficient step size is necessary since this currently would require 900,000 total steps in the search.

To find the search interval for the modulation index, an expression had to be derived based on the Doppler shift of the gravitational wave. I assumed simple circular motion (which is fairly accurate for binary neutron star orbits) and that the orbit lies in the plane of sight. The expression for the motion of the neutron stars along the line of sight is given to be:

$$y(t) = R \sin \Omega t \tag{15}$$

Where R is the radius of the orbit and Ω is the orbital frequency. This changing distance along the line of sight can then be treated as a changing phase for a gravitational wave that is being emitted from the neutron stars. The expression for a gravitational wave with frequency ω that is emitted from a binary neutron star system would then become:

$$x(t) = \cos\left(\omega(t + \frac{y(t)}{c}\right) \tag{16}$$

$$= \cos\left(\omega t + \frac{\omega R}{c}\sin(\Omega t)\right) \tag{17}$$

Where c is the speed of light. Since we are looking for an expression of the form $x(t) = \cos(\omega t + \Gamma \sin \Omega t)$, we can conclude that:

$$\Gamma = \frac{\omega R}{c} \tag{18}$$

Then, using Kepler's third law of planetary motion, an expression for R was determined that was only dependent on the total mass and the orbital frequency of the binary system.

$$R = \left(\frac{GM_{tot}}{\Omega^2}\right)^{1/3} \tag{19}$$

We then have an expression for Γ that is dependent only on the carrier frequency, ω, the total mass of the system, M_{tot}, and the orbital frequency, Ω.

$$\Gamma = \frac{\omega}{c}\left(\frac{GM_{tot}}{\Omega^2}\right)^{1/2} \tag{20}$$

To find the interval of the search over modulation index, we can use the bounds of the intervals for the carrier frequency and modulation frequency, and the knowledge that the vast majority of binary neutron stars have masses of $1.4M_\odot$ and none have been observed with masses larger than $2M_\odot$. These constraints give Γ a large range from 0.1 to values greater than 500,000. Larger steps in Γ seem to be promising, though this must be investigated in greater depth.

3 Results

3.1 Program Capabilities

The program was able to attain a peak located at the correct carrier frequency and modulation frequency for the correct value of the modulation index (Γ). The program automatically finds and displays the values for the three parameters associated with the peak. Figure 3 shows an example of the three-dimentional plot with a peak located at the values of the three parameters. The simulated signal in this plot was given a carrier frequency of 100 Hz, a modulation frequency of 0.05 Hz, and a modulation index of 10.

The peak is easily noticeable and lies in only one frequency bin. However, there are baseline features along the bottom of the peak that are not yet fully understood, though

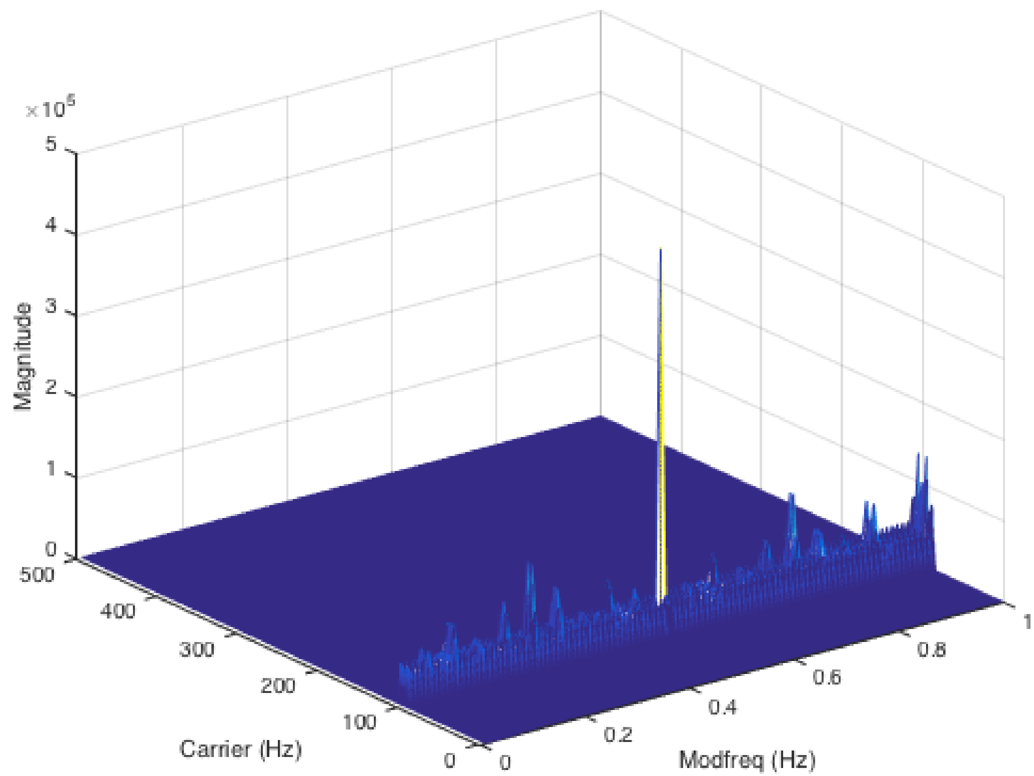


Figure 3: Three-dimensional plot of the peak associated with the correct values of the three parameters.

they are likely a result of the discreteness of the signal and also the brute force convolution picking up the sidebands and reporting false peaks at search values that are multiples of the modulation frequency. The size of these baseline features can vary with the modulation index of the signal and especially with the number of sidebands used from the Jacobi-Anger expansion. This is demonstrated in Figure 4, which shows the three-dimensional plot for a large modulation index and for varying numbers of sidebands. In this example, the carrier frequency is 100 Hz, the modulation frequency is 0.48 Hz, and the modulation index is 1000.

It is evident from Figure 4 that the quality (or narrowness in bins) of the peak in the plot improves significantly when the number of sidebands (terms from the Jacobi-Anger expansion) used in the convolution is increased. For this specific signal, using any fewer than 18 sidebands resulted in an incorrect output of the carrier frequency, however, using any more than 200 sidebands made very little change in the quality of the peak in the plot. It is important to note that generating the arrays that represent the Fourier transform of the modulation factor in the signal (the sidebands) takes a considerable amount of time since they must be regenerated for every value in the Ω and Γ search. Further investigation must be done to optimize the number of sidebands generated for efficiency and accuracy.

3.2 Adding Noise

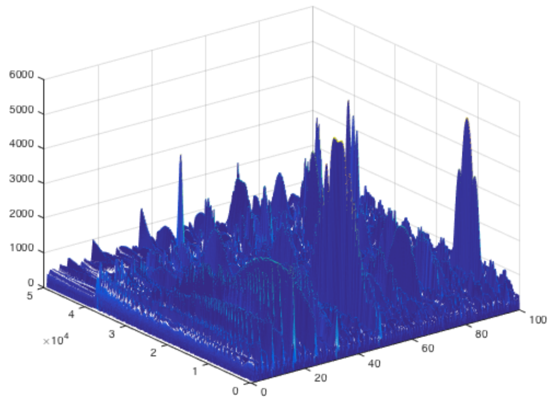
The program was additionally tested on signals with noise of varying amounts. It is still able to determine the correct values for the parameters, though as the amount of noise is increased, more terms from the Jacobi-Anger expansion are needed. Furthermore, there is a limit to the amount of noise that can be added to the signal, after which the program can no longer identify the correct parameter values.

The following three plots in Figure 5 were generated by the program when it was applied to a signal with a carrier frequency of 100 Hz, a modulation frequency equal to 0.5 Hz, and a modulation index of 10. Figure 5(a) shows a plot of the signal of the wave in the top panel, and the signal with noise (equal to 50*randn in MATLAB) added in the bottom panel. Figure 5(b) is a graph of the Fast Fourier Transform of the noisy signal. The carrier frequency can be seen as the tallest peak located at 100 Hz, though the magnitude of this peak is not significantly larger than the magnitude of the noise. Figure 5(c) is the final product of the program, which demonstrates an evident peak located at the expected values for the carrier and the modulation frequency.

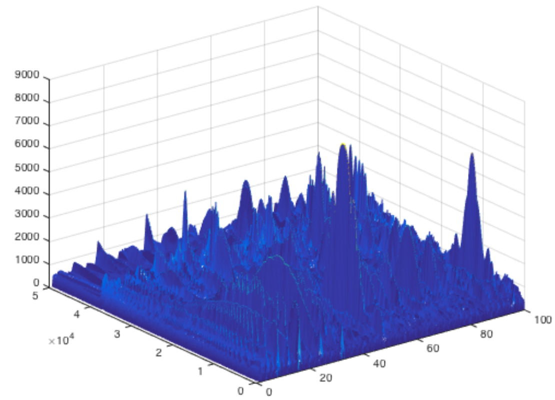
4 Conclusions

The program was capable of correctly determining the expected values for the three parameters, though with some limitations.These limitations are dependent on the size of the data being processed, N, the number of sidebands used in the convolution, and the number of steps in the search over the three parameters.

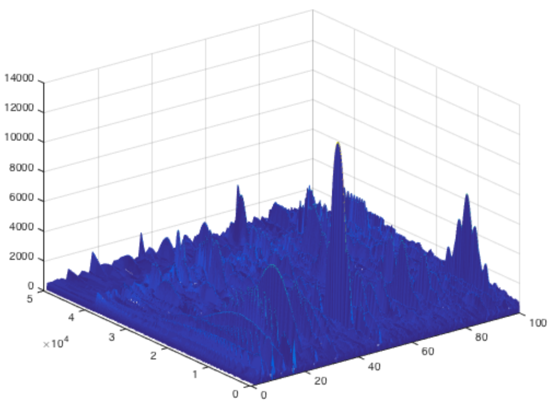
Further investigation on the applicability of the program in an all-sky search must be conducted. Using the same strategy to account for the motion of the neutron stars, the



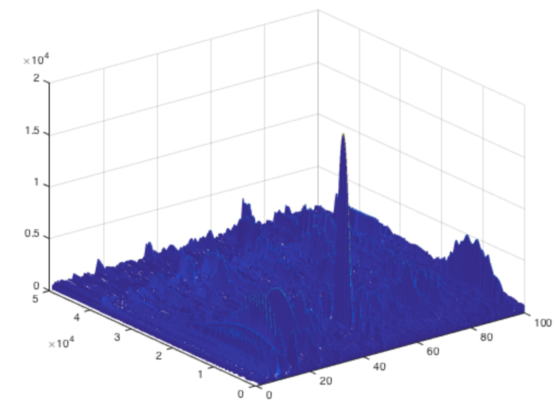
(a) 17 Sidebands



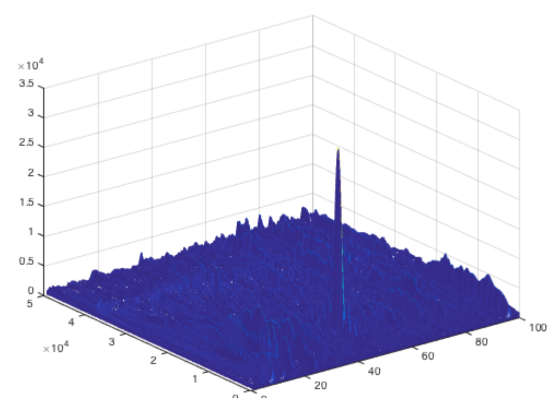
(b) 25 Sidebands



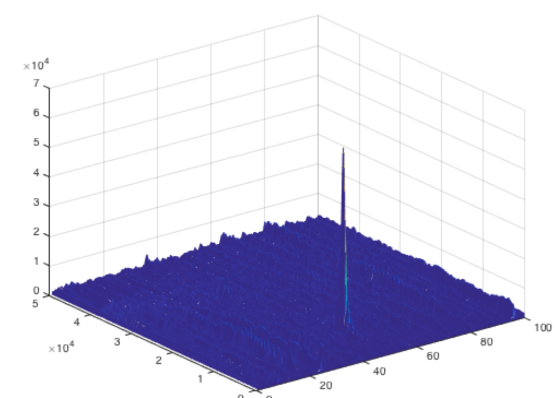
(c) 40 Sidebands



(d) 60 Sidebands

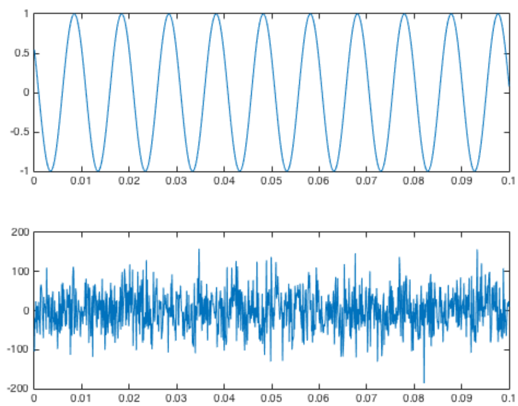


(e) 100 Sidebands

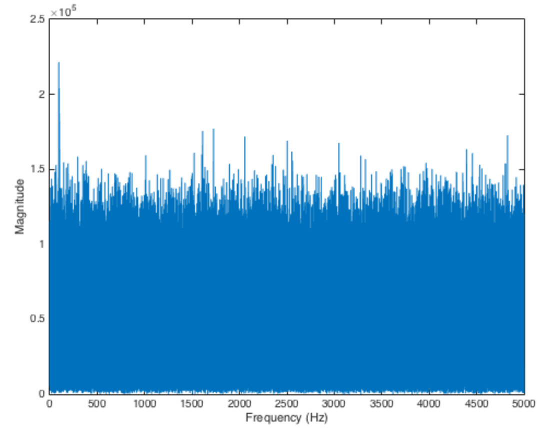


(f) 200 Sidebands

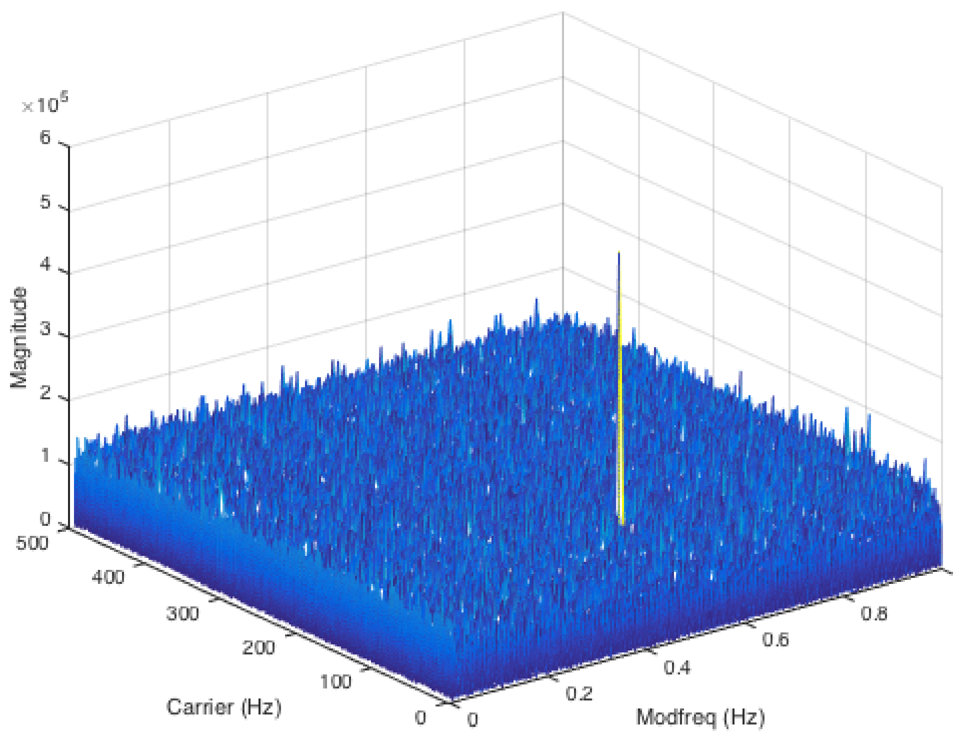
Figure 4: Three-dimensional plot for a signal with $\omega = 100$ Hz, $\Omega = 0.48$ Hz, and $\Gamma = 1000$. Each plot demonstrates the results of the program for a different number of sidebands used in the convolution. Note that the values along the axes of these plots are equal to 100 times the values of ω (y-axis) and Ω (x-axis).



(a) The signal: the top frame is the signal without noise, and the bottom frame is the signal with noise added to it



(b) The FFT of the signal with noise



(c) The three-dimensional plot for a signal with $\omega = 100$ Hz, $\Omega = 0.48$ Hz, and $\Gamma = 1000$.

Figure 5: These three plots were the results of the program applied to a noisy signal (50*randn in MATLAB) with $\omega = 100$ Hz, $\Omega = 0.5$ Hz, and $\Gamma = 10$.

motion of the Earth with respect to the sky is known and can be factored out of the signal. Once again, the FFT and the convolution theroem can be applied to return an array with the motion of the Earth accounted for. From this point, the sky will be stationary with respect to the Earth, and the program can be applied again to determine the three parameter values. There are likely additional applications of this program that should be further investigated.

5 Acknowledgements

I would like to thank my mentor, Dr. Dick Gustafson, for his immeasurable contributions and guidance throughout the process of this project. I would also like to thank all of the LIGO scientists and fellow SURF students for their help and kindness over the course of this summer. Finally, thank you to the NSF and to the Caltech SURF program for making this possible.

6 Apendix

The code for the program is the following:

```
%% CREATING SAMPLED SIGNAL

carrier_frequency = 100;            % Specify carrier frequency (Hz)
modulation_frequency = 0.5;         % Specify modulation frequency (Hz)
Gamma = 10;                         % Specify modulation index
sigma = 0;                          % Specify amount of noise (sigma)

w = 2*pi*carrier_frequency;          % Carrier angular frequency
Omega = 2*pi*modulation_frequency;   % Modulation angular frequency

fs = 10000;                         % Sampling frequency (number of ...
    samples per second)
T = 100;                            % Sample time
N = fs*T;                           % Length of data array/sample
t = (0:N-1)/fs;


% Generating the signal data without noise
x = cos(w*t-Gamma*sin(Omega*t + pi/2));                  % Phase-modulated ...
    function
%x = cos(w*t-Gamma*cos(Omega*t)) + 5*cos(2*pi*80.5*t);  % Phase-modulated ...
    and non-modulated functions
%x = 5*cos(2*pi*80.5*t);                                 % Non-modulated ...
    sine function


% Adding noise to the signal
d = x + sigma*randn(1, N);
```

```
% Plot of signal without noise
figure(1)
subplot(2,1,1)
plot(t,x)
xlim([0,.1])

% Plot of signal with noise
subplot(2,1,2)
plot(t,d)
xlim([0,.1])


%% TAKING THE FFT

X = fft(d);                          % DFT
f = (0:N/2)/T;                       % Frequency range from 0 Hz to the ...
    Nyquist frequency

X_mag = abs(X);                      % Magnitude of FFT
X_magNy = X_mag(1:N/2+1);

% Plot magnitude of FFT
figure(2)
plot(f,X_magNy)
xlabel('Frequency (Hz)')
ylabel('Magnitude')



%% DEFINING INTERVALS OF SEARCH

g_increment = 1;                         % Increment/size gamma steps
g_start = 5;                             % Gamma search limits
g_end = 15;
g_interval = g_end - g_start;            % Must be divisible by g_increment
g_size = g_interval/g_increment;         % Number of gamma steps

w_interval = 500;                        % Carrier search limits (Hz)
j_interval = w_interval*T;               % Array indices of search
j_size = j_interval;                     % Number of carrier steps

Omega_interval= 1;                       % Modulation search limit (Hz)
k_interval = Omega_interval*T;           % Array indices of search
k_size = k_interval;                     % Number of Omega steps



% Which bessel function to extend the expansion to (J_besselterms)

```

```
%besselterms = 20;                          % Set number of terms for all gamma
besselterms = g_start+5;                    % Increasing number of terms with ...
    gamma



% Add zeros to both ends of array so convolution doesn't fall off
padding = k_interval*(besselterms+g_end);     % Distance of search (amount ...
    needed for padding)
X = padarray(X,[0,padding]);


%% PREDEFINING ARRAYS TO BE USED LATER

BesselExpansionCosine = zeros(1,besselterms);
BesselExpansionSine = zeros(1,besselterms);

X_SB = zeros(1,besselterms);

Max_Conv = zeros(j_size,k_size);
Max_Conv_mag = 0;
Conv = zeros(j_size,k_size);                % Pre-defining arrays that will be ...
    used later
Product = zeros(j_size,k_size);
Mag = zeros(1,g_size);
modfreq_all = zeros(1,g_size);
carrier_all = zeros(1,g_size);
answer = zeros(1,g_size);

%% SEARCHING OVER GAMMA, CARRIER, AND MODFREQ


tic      % Begin timing


% Search over gamma
for G = 1:g_size



    g = g_increment*G + g_start;


    % Calculating bessel functions
    b0 = besselj(0,g);


    % Jacobi-Anger expansion for COSINE
    for b = 1:besselterms
        BesselExpansionCosine(b) = -1i*((1i)^b)*besselj(b,g);
    end
```

```
%     % Jacobi-Anger expansion for SINE
%     for b = 1:besselterms
%         BesselExpansionSine(b) = -1i*besselj(b,g);
%     end

    % Search over w (carrier)
    for j = (1:j_interval) + padding
        xj = X(j);

        % Search over W (modulation frequency)
        for k = 1:k_interval

            % Sidebands
            %COSINE expansion
            for n = 1:besselterms
                X_SB(n) = X(j + n*k) + X(j - n*k);
            end
            C = BesselExpansionCosine .* X_SB;
            % Calculating convolution
            Conv_cos((j-padding),k) = xj*b0 + sum(C);

%             %SINE expansion
%             for n = 1:2:besselterms
%                X_SB(n) = X(j + n*k) - X(j - n*k);
%             end
%
%             for n = 2:2:besselterms
%                 X_SB(n) = X(j + n*k) + X(j - n*k);
%             end
%
%             C = BesselExpansionSine .* X_SB;
%
%             % Calculating convolution
%             Conv_sin((j-padding),k) = xj*b0 + sum(C);

        end
```

```
    end


    Conv_mag = abs(Conv_cos);               % Finding magnitude of convolution


    % Extract values at where the expected solution lies for analysis purposes
    answer(G) = Conv_mag(carrier_frequency*T,modulation_frequency*T);


    % Finding candidates for peaks (maximum values)
    [M,I] = max(Conv_mag,[],1);          % Finding max values within each ...
        column (representing steps in modfreq) of convolution matrix
    variation = size(unique(I));         % Number of different values of ...
        modfreq that contain a peak
    [mag, indx] = max(M);                % Finding max row value (carrier)
    Mag(G) = mag;                        % Save max value in an array for ...
        each step of gamma
    modfreq_all(G) = indx/T;             % Array with corresponding W for ...
        max w
    [¬, num] = max(Conv_mag(:,indx));    % Finding w associated with max value
    carrier_all(G) = f(num);             % Array of max w for each gamma


    [¬, m] = max( [mag, Max_Conv_mag] ); % Comparing the max values of ...
        this search in gamma to the previous largest one

    if m == 1

        Max_Conv = Conv_mag;         % If this search was larger, save this ...
            one instead of the last largest one
        Max_Conv_mag = mag;

    end


    % Increasing number of terms used for the next search
    besselterms = besselterms + 1;


    % Report current value of gamma in search
    g


end


% Report time it took to do the full search for all three parameters
toc

```

```
%% FINDING MAX AND PLOTTING


[¬, gval] = max(Mag);                         % Finding gamma with largest ...
    magnitude from convolution search

gamma = g_increment*gval + g_start            % Report gamma value (convert ...
    from array position to value)
carrier = carrier_all(gval)                   % Carrier value where max ...
    magnitude is achieved
modfreq = modfreq_all(gval)                   % Modulation frequency value ...
    where max is achieved

x = (1:k_size)/T;        % Scaling axis with correct values for carrier and ...
    modfreq
y = (1:j_size)/T;
figure(3)
mesh(x,y,Max_Conv)       % Creating 3D plot and naming axes
xlabel('Modfreq (Hz)')
ylabel('Carrier (Hz)')
zlabel('Magnitude')
rotate3d on              % Allowing rotation of 3D plot (click and drag to ...
    rotate))

efficiency = max(Mag) - min(Mag)     % Difference between minimum and ...
    maximum magnitudes in gamma search (if small, may not be accurate enough)


% Plot of maximum magnitudes for each search of gamma
x = (1:g_size) + g_start; % Scaling axis with correct values for gamma
figure(4)
stem(x,Mag)
xlabel('Gamma')
ylabel('Magnitude')


% Sound to indicate end of program
load gong;
sound(y)
```

7 References

Discussions with Dr. Richard Gustafson (University of Michigan)

The Scientist and Engineer's Guide to Digital Signal Processing, Dr. Steven W. Smith (can be accessed at `www.dspguide.com`)