Assignment 1: The Hunt for Red October

William Ojemann
AMATH 482
January 27th, 2021

Abstract

This report outlines the efforts to use an analysis of a new acoustic emitting technology contained by submarines to precisely locate a submarine across a 24-hour period. The performed analysis utilizes a suite of filtering and frequency isolation techniques to filter out noise in acoustic recordings of the water surrounding the submarine. The result of the application of these tools is the identification of the submarine's acoustic emission frequency as well as the location of the submarine at each time point a sample was taken. With the resulting locations are then used by a P-8 Poseidon to track the sub.

Introduction and Overview

The field of signal processing and analysis has a myriad of diverse applications, but they all rely on analyzing signals in both time, space, and frequency. The foundation of these signal analyses lies in the application of the Fourier transform, which allows for the segmentation and analysis of the frequencies of signals measured in time or space. The Fourier transform and its inverse have a variety of applications much like the broader field of signal processing as it can be used for frequency visualization, frequency filtering, and dimensionality reduction among other uses.

The goal of this project is to analyze underwater sound recordings of the general area of a submarine in order to locate and track the sub through space and time and to identify the frequency of its radar's acoustic emissions. The solution presented in this report was reached through a combination of Fourier analysis and manipulation of the data in the frequency space. The signal processing and filtering steps that were applied to this relevant issue are highly translatable to other real-world problems but require a background in Fourier transforms as well as applying filters in the frequency domain. This report will cover these mathematical fundamentals as well as an overview of their implementation in MATLAB.

Theoretical Background

Equations and theory are adapted from lecture notes. As was previously mentioned, the theoretical foundation for signal analysis lies in Fourier harmonic analysis. The Fourier transform, a method for decomposing a function into its frequencies, is based on the work by Jean-Baptiste Joseph Fourier where he first created something called a Fourier series (1).

$$f(x) = a_0 + \sum_{\mathrm{k}=1}^{\infty} \left(a_{\mathrm{k}} \cos\frac{\mathrm{k}\pi x}{L} + b_{\mathrm{k}} \sin\frac{\mathrm{k}\pi x}{L}\right) \quad \text{eq.1}$$
$$x \in [-\mathrm{L}, \mathrm{L}]$$

The Fourier series is a way to represent a non-discrete function, f(x), that is periodic on the range $[-\mathrm{L},\mathrm{L}]$ as the infinite sum of cosine and sine functions with discrete frequencies, k, ranging from 1 to infinity. This decomposition is based on Euler's theorem, which shows how the oscillatory nature of a complex exponential can be represented as harmonic functions (2).

$$e^{i\theta} = cos(\theta) + isin(\theta) \qquad \text{eq. 2}$$

While the function f(x) cannot be discrete, the Fourier series sums discrete frequencies and the contribution of each frequency to the function is dictated by the constants a_0, a_k, and b_k, which are found by equations 2, 3, and 4.

$$a_k = \frac{1}{L}\int_{-L}^{L} f(x)cos\left(\frac{\pi kx}{L}\right)dx \qquad \text{eq. 3}$$

$$b_k = \frac{1}{L}\int_{-L}^{L} f(x)sin\left(\frac{\pi kx}{L}\right)dx \qquad \text{eq. 4}$$

$$a_0 = \frac{1}{2L}\int_{-L}^{L} f(x)dx \qquad \text{eq. 5}$$

While the function f(x) must be periodic on the boundary $[-\mathrm{L}, \mathrm{L}]$, the Fourier series can still be used to decompose a discontinuous function as discontinuities become averaged by their left and right limits through the partial sums of continuous sine functions.

The Fourier series is useful for analyzing the frequency decomposition of a function within a limited boundary, but the Fourier transform presents an alternative as the range of the function approaches infinite time or space. The Fourier transform (FT) (6) is a projection that transforms a signal from the time or space domain, x, into the frequency domain, k. The inverse FT (7) is very similar to the FT except it does the opposite and projects data from the frequency domain back into time or space.

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)e^{-ikx}dx \qquad \text{eq. 6}$$

$$f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \hat{f}\,(k)e^{ikx}dk \qquad \text{eq. 7}$$

The FT takes a function, f(x) that spans infinite time or space and gives the magnitude of the frequencies that make up the continuous and infinite function, f(k) (**figure 1**). This allows us to analyze a signal across infinite time or space and frequency, and the Fourier series lets us analyze a more practical periodic function bounded on a range $[-\mathrm{L}, \mathrm{L}]$, but both require a continuous function f(x). In many situations the signal being investigated is discretized through sampling and incompatible with the traditional continuous FT. In these cases, we turn to the discrete Fourier transform (DFT) (8) and its complement, the inverse DFT (9).

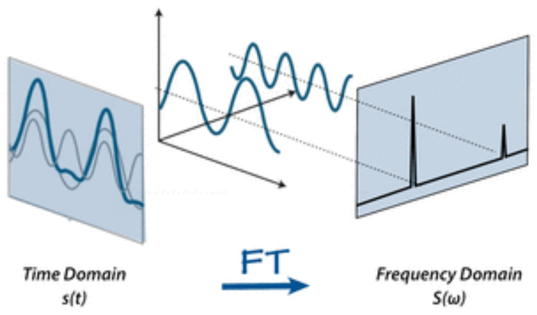


Figure 1: *A Fourier transform projects between frequency and space/time [1].*

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{2\pi ikn}{N}} \qquad \text{eq. 8}$$

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{\frac{2\pi ikn}{N}} \qquad \text{eq. 9}$$

The DFT takes N equally spaced samples of the signal periodic on the range [-L, L], x, and projects the data into the frequency domain, where it portrays N frequency components, k. Because the number of frequencies produced by the transform is limited by the number of samples, a faster sampling rate is necessary to accurately capture data with higher frequency oscillations. To avoid under sampling and misrepresenting the data, a phenomenon called aliasing, it is best practice to sample at two times the fastest frequency in the data. Since the formulation of the original DFT, a novel method for calculating the frequency decomposition of a signal has been developed called the fast Fourier transform (FFT). The FFT and its inverse (iFFT) are the same as the DFT and its inverse for the scope of this paper except that it has a runtime of $O(NlogN)$ as opposed to $O(N^2)$ for the DFT.

While the FT is fundamental to frequency decomposition, much of the analysis performed in this report relies on the understanding of a Gaussian, or normal, distribution or function. The Gaussian distribution describes a probabilistic distribution with a mean of zero and a shape called a bell curve. White noise, a common barrier to signal analysis, applies to the whole frequency spectrum and is drawn from a normal distribution of complex or real values. Besides characterizing noise, multiplying a Gaussian function (10) by a signal in frequency space has the effect of filtering out frequency components that are not under the bell curve. Using this principle, a Gaussian filter (**figure 3, right**) can be applied to any frequency in the spectrum by shifting the Gaussian curve and filtering out any noise frequencies. While the Gaussian filter is traditionally represented in two dimensions, it can also be in three or more dimensions as the dimensionality of the signal increases.

$$G(\mathrm{k}) = e^{-\tau(\mathrm{k}-\mathrm{k}_0)^2} \qquad \text{eq. 10}$$

Algorithm Implementation and Development

Achieving either of our goals – tracking the sub and identifying its acoustic emission - starts with formatting the data. The given recording data has 64 Fourier modes, or samples, and can be reshaped into a three-dimensional 64x64x64 dimension array for each of the 49 time points taken over a 24-hour period (**figure 2**). The three dimensions must be scaled to reflect the spatial and frequency domains instead of matrix indices. The x, y, and z spatial dimensions were created by scaling the 64 samples to lie on a linear space within the dimensions of the data, which is from -10 to 10 spatial units, or [-L, L]. The frequency dimensions – Kx, Ky, and Kz – were scaled from matrix indexing by $\frac{2\pi}{2L}$ to accommodate the FFT's requirement in MATLAB of periodicity over a 2π region and then shifted from MATLAB's frequency indexing into a [-L, L] range. A 3D mesh grid was created to represent both the frequency and spatial dimensions.

With the newly formatted data, the next step is to identify the center frequency of the

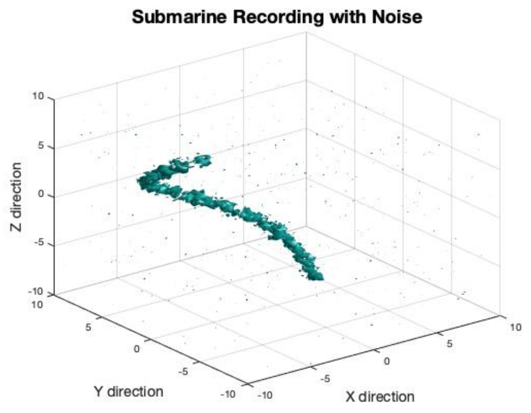


Figure 2: *Absolute value of the submarine's scaled acoustic signature in the spatial domain without noise filtering.*

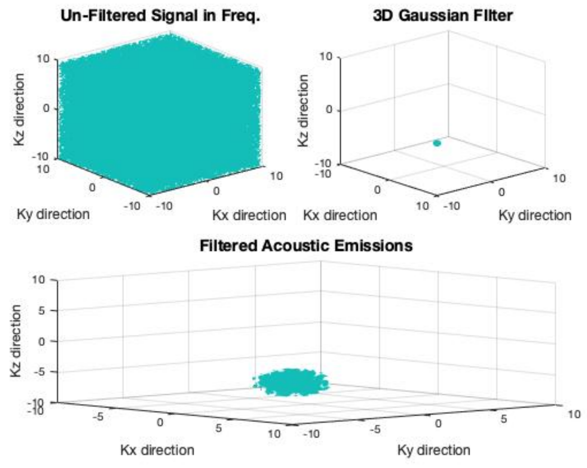
submarine's acoustic emissions. Applying an FFT to the signal allows for the analysis of each of the 49 timestamps projected into the 64x64x64 frequency space. While the emissions vary in the spatial dimension, the projections into frequency space for each sample remains constant as long as the sub is emitting at the same frequency. The fact that the center frequency will be aligned across samples and the assumption that the noise in the signal is a Gaussian distribution means the center frequency can be isolated by averaging. The absolute value of the projections of the data into the frequency space were averaged across the 49 samples, and the coordinates and amplitude of the maximum value in the resulting average frequency space were identified. When analyzing the signal, the absolute value of the frequencies and acoustic signal allows for the inclusion of any relevant complex data in the analysis. The coordinates were then transformed from matrix indices to the frequency values using the frequency mesh grid to reveal the center frequency.

In order to track the submarine through space and time, the newly acquired center frequency of the submarine's acoustic emissions was used to orient a Gaussian filter. The Gaussian filter was created by multiplying three Gaussian functions (10), one for each frequency dimension, that were each centered around that dimension's center frequency coordinate and scaled with a τ of 1. The value of τ was optimized through trial and error to produce the clearest coordinates for the sub's location. Multiplying the frequency domain for each sample by the filter greatly reduced noise in the signal and the frequency space representation of the signal was transformed back into the spatial domain with an iFFT. Using the newly filtered signal, the precise location of the submarine at each time point was identified by finding the indices of the maximum value of the acoustic signature for teach time point. Using the spatial domain mesh grid, these indices were converted to spatial coordinates and plotted to show the path of the submarine over the 24-hour period.

Computational Results

The results of averaging the signal across samples in the frequency space and then finding the maximum frequency showed the submarines acoustic emission frequency to be [-4.712, 3.142, -7.854] in the Kx, Ky, and Kz dimensions.

Figure 3: *(top left) Unfiltered signal projected into frequency space from the first sample of 49. (top right) 3D Gaussian filter located at the center frequency of the submarine's acoustic emissions. (bottom) filtered signal in frequency space.*



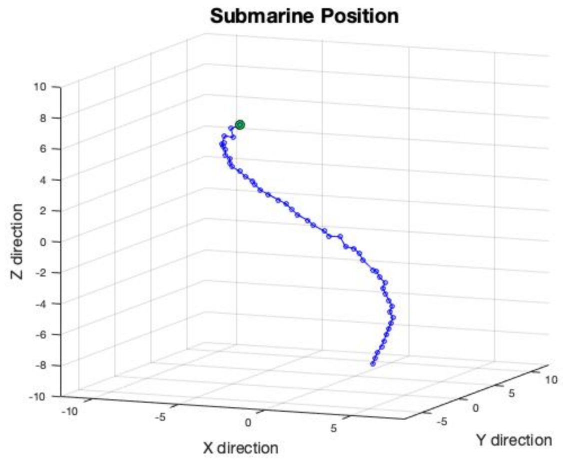
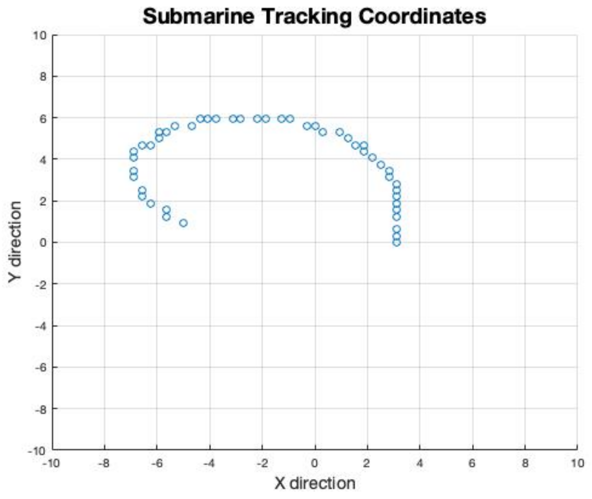


Figure 4: *Precise location of the submarine in 3D space at each of the 49 samples taken of the area around the sub. The green dot is the end point.*

Using information about the acoustic emissions, a Gaussian filter was applied around the center frequency (**figure 3**), and the submarine's position in 3D space at each time point was isolated (**figure 4**). Based on the resulting curve, the submarine can be seen surfacing in a helical pattern with the final position in green. From the 3D spatial coordinates, the precise X and Y coordinates of the submarine at each time point were extracted from the data and are shown in **Appendix A** for the navigation system of the P-8 Poseidon sub tracking aircraft.

Figure 5: *Coordinates for submarine tracking using a P-8 Poseidon Sub Tracker.*



Summary and Conclusions

This project demonstrates one of the many real-world applications of signal processing. Through processes like Fourier transforms, frequency space averaging, and applying Gaussian filters, the efforts successfully identified the submarine's acoustic emission frequency, filtered out signal noise, and located the submarine. The outlined procedure demonstrates the importance of data analysis not only to the mathematical field but to a diverse set of applications in industry and academia alike.

References

[1] "Fourier Transform (FT)," *Questions and Answers in MRI*. [Online]. Available: http://mriquestions.com/fourier-transform-ft.html. [Accessed: 27-Jan-2021].

Appendix A

Table 1

X and Y coordinates for tracking the submarine.

X Coordinates	Y Coordinates	X Coordinates	Y Coordinates
3.125	0	-2.8125	5.9375
3.125	0.3125	-3.125	5.9375
3.125	0.625	-3.75	5.9375
3.125	1.25	-4.0625	5.9375
3.125	1.5625	-4.375	5.9375
3.125	1.875	-4.6875	5.625
3.125	2.1875	-5.3125	5.625
3.125	2.5	-5.625	5.3125
3.125	2.8125	-5.9375	5.3125
2.8125	3.125	-5.9375	5
2.8125	3.4375	-6.25	4.6875
2.5	3.75	-6.5625	4.6875
2.1875	4.0625	-6.875	4.375
1.875	4.375	-6.875	4.0625
1.875	4.6875	-6.875	4.0625
1.5625	4.6875	-6.875	3.4375
1.25	5	-6.875	3.4375
0.9375	5.3125	-6.875	3.125
0.3125	5.3125	-6.5625	2.5
0	5.625	-6.5625	2.1875
-0.3125	5.625	-6.25	1.875
-0.9375	5.9375	-5.625	1.5625
-1.25	5.9375	-5.625	1.25
-1.875	5.9375	-5	0.9375
-2.1875	5.9375		

Appendix B – MATLAB Functions

load filename.mat – imports a .mat file into variable, used to load the submarine data.
linspace(start, end, int) – creates an array of variables between the start and end value with int data points, used to create the space and frequency dimensions with the number of Fourier modes taken by the data acquisition unit.
fftshift(K) – shifts the zero-frequency component of a frequency spectrum to the center of the array, used to shift the zero-frequency component to the center of our frequency dimensions after initializing.
[D1, D2, D3] = meshgrid(d1, d2, d3) – creates a 3D matrix of coordinates for each of the given the axes, used to create a coordinate matrix for each axis in the frequency and spatial dimensions.
reshape(data, n, n, n) – takes in a data array and reshapes it to a tensor with a dimension for each n where the number of elements in each dimension is the value of each n, used to reshape the subdata for each time point from a 26144x1 array into a 64x64x64 matrix corresponding to our frequency and space coordinate grids.
abs(x) – takes the absolute value of x, or if x is an array then it takes the absolute value of each element in x, used to consider the complex component of a data point at many points throughout the implementation.
fftn(x) – takes the n dimensional fast Fourier transform of the n-dimensional array x, used to calculate the FFT along each of the spatial dimensions in our 64x64x64 matrix,
ifftn(x) – takes the n dimensional inverse fast Fourier transform of the n-dimensional array x that contains frequencies, used to transform the signal back into the spatial dimension from the frequency dimension.
[maxEnergy, maxIndex] = max(x, [], 'all', 'linear') – finds the maximum element in x and returns the maximum value as well as the index of that maximum value in a linear array, used to find the center frequency and largest acoustic signal after filtering.
[d1, d2, d3] = ind2sub([n, n, n], index) – takes a linear index and returns the coordinates of that value in an n x n x n dimensional matrix, used to transform the linear index returned by the max function to indices in a coordinate grid.
isosurface(Dim1, Dim2, Dim3, Data, Value) – generates isosurface geometry to represent four dimensional data where Dim1-3 are 3D coordinate matrices that give locations for each value in Data. Used to represent our submarine data as a contoured surface because the complicated nature of the data makes it difficult to visualize in a traditional plot.
plot3(Cord1, Cord2, Cord3) – makes a plot similar to the traditional plot function except with 3 coordinates for plotting in 3 dimensions, used to plot the path of the submarine.

Appendix C – MATLAB Code

```
%% Assignment 1
% William Ojemann
% AMATH 482
% January 10th, 2021
% Submitted - January 27th, 2021
close all; clear; clc;
%% Data Formatting - Provided Code
load subdata.mat % Imports the data as the 262144x49 (space by time) matrix
called subdata 5
L = 10; % spatial domain
n = 64; % Fourier modes
x2 = linspace(-L,L,n+1); x = x2(1:n); y =x; z = x;
k = (2*pi/(2*L))*[0:(n/2 - 1) -n/2:-1]; ks = fftshift(k);
```

```
% Creating 3D coordinate grids for space and frequency
[X,Y,Z]=meshgrid(x,y,z);
[Kx,Ky,Kz]=meshgrid(ks,ks,ks);
% Plotting raw data
figure
for j=1:49
    Un(:,:,:)=reshape(subdata(:,j),n,n,n);
    M = max(abs(Un),[],'all');
    isosurface(X,Y,Z,abs(Un)/M,0.7);
    axis([-10 10 -10 10 -10 10]), grid on
    drawnow;
    pause(0.01);
end
% Plot formatting
xlabel('X direction', 'FontSize', 14)
ylabel('Y direction', 'FontSize', 14)
zlabel('Z direction', 'FontSize', 14)
title('Submarine Recording with Noise', 'FontSize', 18)
%% Identifying Frequency Signature
avg_freqs = zeros(n,n,n); % initialize the 3D matrix
for j = 1:49
   avg_freqs = avg_freqs + fftn(reshape(subdata(:,j),n,n,n));
end
avg_freqs = abs(avg_freqs)/49; % creating the matrix of average frequencies

[maxE, maxIdx] = max((avg_freqs),[],'all','linear'); % finding index of max
frequency
[xi, yi, zi] = ind2sub([n, n, n], maxIdx); % converting index to 3D indices
% Converting matrix indices to frequency values
Kxi = Kx(xi,yi,zi);
Kyi = Ky(xi,yi,zi);
Kzi = Kz(xi,yi,zi);
center_freq = [Kxi,Kyi,Kzi];
%% Filtering
% Creating the Gaussian filter
tau = 1;
filterx = exp(-tau.*(Kx - Kxi).^2);
filtery = exp(-tau.*(Ky - Kyi).^2);
filterz = exp(-tau.*(Kz - Kzi).^2);
filter = filterx.*filtery.*filterz;
% Filtering one time stamp for visualization
j = 1;
Un(:,:,:)=reshape(subdata(:,j),n,n,n);
Ut = fftn(Un);
Uft = fftn(Un).*filter;
% Plotting filtering process
figure
view(3)
camlight
lighting gouraud
hold on
subplot(2,2,2)
isosurface(Kx,Ky,Kz,filter,.7);
xlabel('Kx direction', 'FontSize', 12)
ylabel('Ky direction', 'FontSize', 12)
zlabel('Kz direction', 'FontSize', 12)
title('3D Gaussian Filter', 'FontSize', 14)
axis([-10 10 -10 10 -10 10]), grid on
```

```
subplot(2,1,2)
isosurface(Kx,Ky,Kz,real(Uft),.7);
xlabel('Kx direction', 'FontSize', 12)
ylabel('Ky direction', 'FontSize', 12)
zlabel('Kz direction', 'FontSize', 12)
title('Filtered Acoustic Emissions', 'FontSize', 14)
axis([-10 10 -10 10 -10 10]), grid on
subplot(2,2,1)
isosurface(Kx,Ky,Kz,real(Ut),.7);
xlabel('Kx direction', 'FontSize', 12)
ylabel('Ky direction', 'FontSize', 12)
zlabel('Kz direction', 'FontSize', 12)
title('Un-Filtered Signal in Freq.', 'FontSize', 14)
axis([-10 10 -10 10 -10 10]), grid on
hold off
%% Finding The Maximum in Space
maxes = zeros(49,3);
for j=1:49
    Un(:,:,:)=reshape(subdata(:,j),n,n,n);
    Uft = fftn(Un).*filter; % frequency and shifted and filtered
    Unf = ifftn(Uft); % shifting back to space
    [~,temp_max_idx] = max(abs(Unf),[],'all','linear'); %index of maximum
frequency at time stamp
    [x, y, z] = ind2sub([n,n,n], temp_max_idx); %transforming to matrix
    maxes(j,1:3) = [X(x,y,z), Y(x,y,z), Z(x,y,z)]; % storing
end
% Plotting
figure
hold on
g = plot3(maxes(1:end,1),maxes(1:end,2),maxes(1:end,3),'-bo');
p = plot3(maxes(end,1),maxes(end,2),maxes(end,3),'ok');
set(p, 'markerfacecolor', 'g');
set(g, 'MarkerSize',4);
set(p, 'MarkerSize',8);
axis([-10 10 -10 10 -10 10]), grid on
xlabel('X direction', 'FontSize', 14)
ylabel('Y direction', 'FontSize', 14)
zlabel('Z direction', 'FontSize', 14)
title('Submarine Position', 'FontSize', 18)
hold off
%% Poseidon Sub Tracking
% Creating coordinates table
Xcoordinates = maxes(:,1);
Ycoordinates = maxes(:,2);
TrackingCoordinates = table(Xcoordinates,Ycoordinates);
% Plotting coordinates on grid
figure
scatter(Xcoordinates,Ycoordinates);
xlabel('X direction', 'FontSize', 14)
ylabel('Y direction', 'FontSize', 14)
title('Submarine Tracking Coordinates', 'FontSize', 18)
axis([-10 10 -10 10]), grid on
```