Amath 482 Homework 1 Report
A Submarine Problem

Colin Kong

January 26th, 2021

Abstract: The project requires to use applications of Fourier Transforms, filtering and averaging in order to determine the frequency signature among 49 columns of data and then find the path of a submarine and its final location so that we can send an aircraft to the location.

Section I: Introduction and Overview

Introduction: The problem is about hunting for a submarine in the Puget Sound using noisy acoustic data. It is a new submarine technology that emits an unknown acoustic frequency that you need to detect. Using a broad spectrum recoding of acoustics, data is obtained over a 24-hour period in half-hour increments. Unfortunately, the submarine is moving, so its location and path need to be determined. Try to locate the submarine and find its trajectory using the acoustic signature.

Overview: In order to find the location of the submarine, first we need to start by removing the noisy signals from the original data by averaging the spectrum and finding out the center frequency generated by the submarine and this can be done by applying a Fourier Transform. Then, we need to apply a Gaussian filter to denoise the data and determine the path of the submarine which can be done by the inverse Fourier Transform. Finally, we can find the final position of the submarine and then send our P-8 Poseidon subtracking aircraft.

Section II: Theoretical Background

Fourier Series:

Suppose we are given a function f(x) for a limited range of x values. For simplicity, we will take $x \in [-\pi, \pi]$. The first thing to note is that all frequencies won't 'fit' into this interval. That is, we will only consider sin(kx) and cos(kx) with integer k since these are the only frequencies that lead to functions that completely repeat over the interval. This leaves infinitely many frequencies, given by k= 1,2,3,4,…, and we write f(x) as the infinite sum of these sines and cosines:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_n \cos(kx) + b_n \sin(kx)), \quad x \in [-\pi, \pi].$$

The constant term $\frac{a_0}{2}$ is for shifting the function up and down.

Fourier Transform:

Suppose we are given a function f(x) with x∈R. We define the Fourier transform of $f(x)$, written $\hat{f}(k)$ by the formula:

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

Furthermore, if you are given $\hat{f}(k)$ and want to recover $f(x)$, you can use the inverse Fourier transform:

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

Knowing that e^{ikx} is like sin(kx) and cos(kx), the value of k gives the frequencies of sine and cosine waves. There-fore, the Fourier transform takes a function of space (or time), x, and converts it to a function of frequencies, k.

Fast Fourier Transform:

The Fast Fourier transform, which we always call it FFT, performs the forward and backward Fourier transforms. Its advantage is the relatively low operation count of $O(N \log N)$ which shows its efficiency when dealing with large data.

One of the crucial features of FFT, which is also the key reason of its low operation count $O(N \log N)$, is the range x ∈ [−L,L] which discretizes into 2^n points, i.e. the number of points should be 2, 4, 8, 16, 32, 64, 128, 256…

Averaging:

When we start a radio detection, there will be large amount of noises around which it is hard for us to find the real signal. Thus, averaging over the realizations in frequency space can help to cancel out the noise from each realization and help to reach to the true signal. We will use Fast Fourier Transform on the data and average all the noisy data to 0. Then, we will take the its absolute value and divide by the maximum to find the normalized data and by taking the maximum value, we shall get the center frequency.

Filtering:

Filtering is to filter out the noise in order to better detect the signal. A filter is a mathematical function that we can apply:

$$F(k) = e^{-\tau(k-k_0)^2}$$

This is a Gaussian function with k standing for the frequency, τ determining the width of the filter and the constant k_0 determining the center of the filter. Using the filter, we will be able to denoise the data and determine the path of submarine by multiplying filter to the Fourier Transformed data and later we will be using the inverse Fourier Transform to set the results back to the time domain.

Section III: Algorithm implementation and Development

The project has four steps:

1. **Run the start code in MATLAB:** Load the subdata.mat, set up the environment for Fourier Trasnform, and brings a 3-D plot to represent the noise data

2. **Through averaging of the spectrum, determine the frequency signature (center frequency) generated by the submarine:** Use Fast Fourier Transform to transform all the data to the frequency domain, then use the absolute value of the data to be divided by the number of columns of the data which is 49, so that all the data has been averaged, plot the coordinates in the form of a matrix, determine the central frequency from the maximum value of the matrix

3. **Filter the data around the center frequency determined above in order to denoise the data and determine the path of the submarine. Use plot3 to plot the path of the submarine once you have it:** Create and apply a Gaussian filter based on the center frequency we discovered, multiply the Fourier Transformed data with the filter, take the inverse of the Fourier Transform to set the results back to the time domain and draw the 3-D plot with the coordinates, then we can determine the path of the submarine

4. **Where should you send your P-8 Poseidon subtracking aircraft? Give the x and y coordinates in a table to follow the submarine:** Find the final positions for the submarine and point out on the plot together with the path of the submarine, and these positions are the answers for sending the aircraft

Section IV: Computational Results

1. **Averaging the noisy data and finding the center frequency**

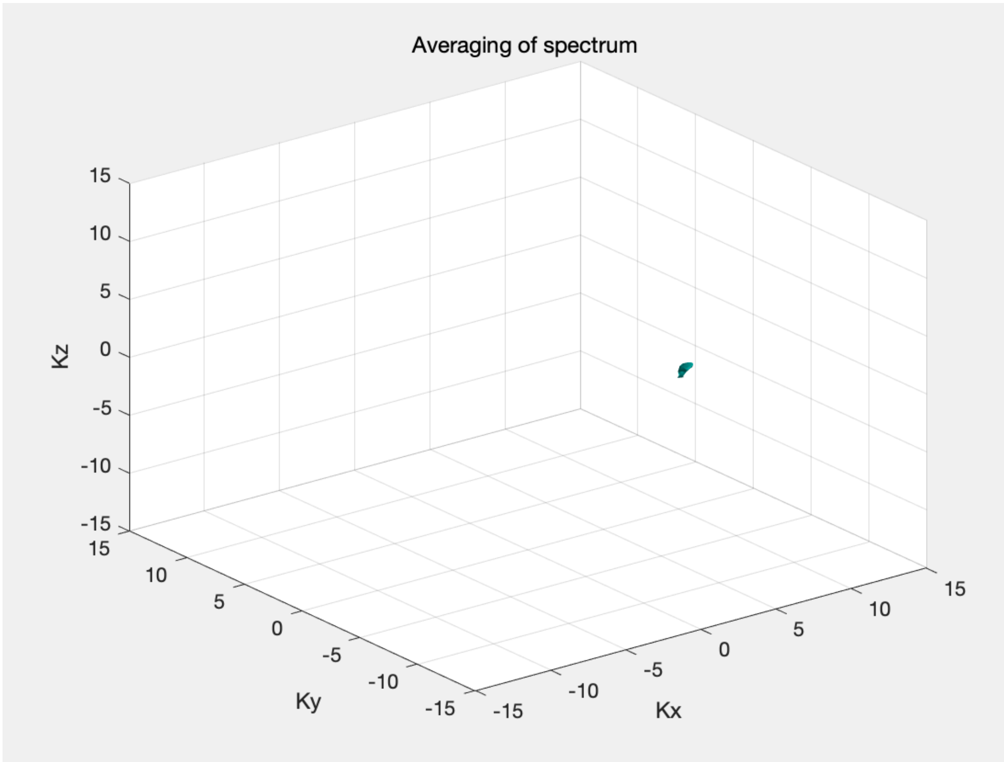

Figure 1: averaged data in the frequency domain

The coordinates of the center frequency in [x,y,z] form is [5.3407, -6.9115, 2.1991].

2. Apply the filter to denoise the data and find the path of the submarine

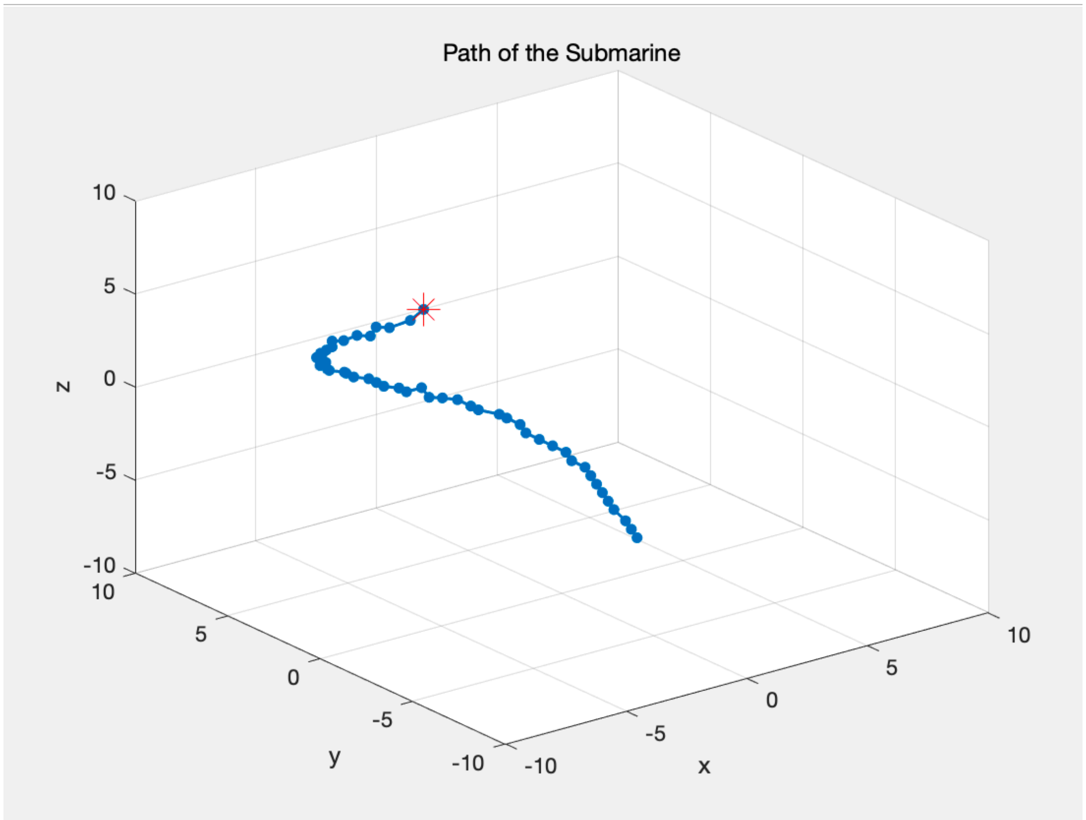

Figure 2: the path of the submarine

The coordinates of the final location to send the aircraft in [x,y,z] form is [-5.0000, 0.9375, 6.5625].

Section V: Summary and Conclusion

After applying the method of averaging through Fourier Transform, we successfully denoise the noisy acoustic data and find the true signal of the submarine. We find the center frequency of [5.3407, -6.9115, 2.1991]. And also, we apply the Gaussian filter with setting the bandwidth τ to be 0.2. Thus, we may able to send the aircraft to the location of the submarine which is [-5.0000, 0.9375, 6.5625].

Appendix A: MATLAB Functions Used

1. **fftn(x)** – returns the multidimensional Fourier transform of an N-D array using a fast Fourier transform algorithm. The N-D transform is equivalent to computing the 1-D transform along each dimension of X. The output Y is the same size as X.
2. **reshape (A, [M,N])** – reshapes A into a given matrix.
3. **isosurface (X,Y,Z,V,isovalue)** – computes isosurface data from the volume data V at the isosurface value specified in isovalue
4. **[I1,I2,…,In] = ind2sub(sz,ind)** – returns n arrays I1, I2, …, in containing the equivalent multidimensional subscripts corresponding to the linear indices ind for a multidimensional array of size sz
5. **find(x)** – returns a vector containing the linear indices of each nonzero element in array x
6. **fftshift(x)** – rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.
7. **ifftn(x)** – returns the multidimensional discrete inverse Fourier transform of an N-D array using a fast Fourier transform algorithm. The N-D inverse transform is equivalent to computing the 1-D inverse transform along each dimension of Y. The output X is the same size as Y.
8. **plot3** – plot coordinates in 3D space

Appendix B: MATLAB Codes

```
clear all; close all; clc

load subdata.mat % Imports the data as the 262144x49 (space by time) matrix called subdata
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);

[X,Y,Z]=meshgrid(x,y,z);
[Kx,Ky,Kz]=meshgrid(ks,ks,ks);

for j=1:49
    Un(:,:,☺=reshape(subdata(:,j),n,n,n);
    M = max(abs(Un),[],’all’);
    close all, isosurface(X,Y,Z,abs(Un)/M,0.7)
    axis([-20 20 -20 20 -20 20]), grid on, drawnow
    pause(1)
```

```
end

% Average the spectrum with normalizing fourier transformed data
ave = zeros(1,n);
for j = 1:49
	Un(:,:,☺=reshape(subdata(:,j),n,n,n);
	Utn = fftn(Un);
	ave = ave + Utn;
end
ave = abs(fftshift(ave))/49;
figure(1);
close all, isosurface(Kx,Ky,Kz,ave./max(ave(☺),0.7);
axis([-15 15 -15 15 -15 15]), grid on, drawnow
title(“Averaging of spectrum “);
xlabel(“Kx”),ylabel(“Ky”),zlabel(“Kz”);

% Center Frequency
[x,y,z] = ind2sub([n,n,n], find(ave == max(abs(ave(☺))));
centFreq = [Kx(x,y,z), Ky(x,y,z), Kz(x,y,z)];
kx_0 = Kx(x,y,z); ky_0 = Ky(x,y,z); kz_0 = Kz(x,y,z);

% Gaussian filter
tau = 0.2;
filter = exp(-tau*((Kx-kx_0).^2+(Ky-ky_0).^2+(Kz-kz_0).^2));

% Determinethe path of the submarine.
Positions = zeros(49,3);
for j = 1:49
	Un_f(:,:,☺=reshape(subdata(:,j),n,n,n);
	Utn_f = fftn(Un_f);
	Unft = filter.*fftshift(Utn_f);
	Unf = ifftn(Unft);
	Max = max(abs(Unf(☺));
	[positionsX,positionsY,positionsZ] = ind2sub([n,n,n], find(abs(Unf)==Max));
	positions(j,1) = X(positionsX,positionsY,positionsZ);
	positions(j,2) = Y(positionsX,positionsY,positionsZ);
	positions(j,3) = Z(positionsX,positionsY,positionsZ);
end
figure(2);
plot3(positions(:,1),positions(:,2),positions(:,3),”.-“,”Linewidth”,1.5,”Markersize”, 15);
axis([-10 10 -10 10 -10 10]);
```

```
xlabel("x"),ylabel("y"),zlabel("z"), grid on;
title("Path of the Submarine");
hold on
% Final positions for P-8 Poseidon subtracking aircraft
plot3(positions(49,1),positions(49,2),positions(49,3),"r*","Markersize",15);
```

Reference:

1. Course notes by Professor Jason J. Bramburger
2. MathWorks Website