Assignment 1

A Submarine Problem

Author: Sofia Chen

Abstract: This assignment is to locate a new class of submarine in the Puget Sound. We want to find its trajectory using the acoustic signature and identify the acoustic admissions of this new class of submarine.

Section I. Introduction and Overview

Assume that we are 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 we need to detect. Using a broad-spectrum recoding of acoustics, data is obtained over a 24-hour period in half-hour increments. We need to determine the location and path of the submarine, since it is moving.

Section II. Theoretical Background

As we have the data about the signal information of the submarine, we can use the multidimensional Fourier transformation to look at the data in the frequency space. Through averaging of the spectrum, we can determine the center frequency generated by the submarine.

After that, we can filter the data around the center frequency determined above. With the filter, we denoise the data and then determine the path of the submarine.

Section III. Algorithm Implementation and Development

I used *X,Y,Z* to represent the spatial space and *Kx,Ky,Kz* to represent the frequency space. Both these two domains are 3-dimensionl. Surely, we must rescale the frequencies by $2\pi/L$ since the Fast Fourier Transform assumes 2π periodic signals.

And then I average the spectrum. I used the Fourier transform for higher dimensional command to transform the original signal into the frequency domain in each time realization. After all the 49 realizations, I cancel out the noise by averaging the spectrum. The max value of the average has the highest frequency of occurrence, meaning the associated K values in frequency domain represent the center frequency in the Fourier transform.

After we found the center frequency, we can filter around it to denoise the data and determine the path of the submarine. I choose to use the gaussian filter. I put the three center frequency values in to three gaussian equations separately, and then filter the original function by multiplying them together. Then, we shift the filtered function back to the spatial domain, and we can pull out the coordinates of the location of the submarine from the max of it during each realization. We also store the coordinates into

3 arrays in length of 49 during the process, which represents each dimension. After the 49 realizations, we can use the built-in 3D plot function in Matlab to plot the trajectory of the submarine.

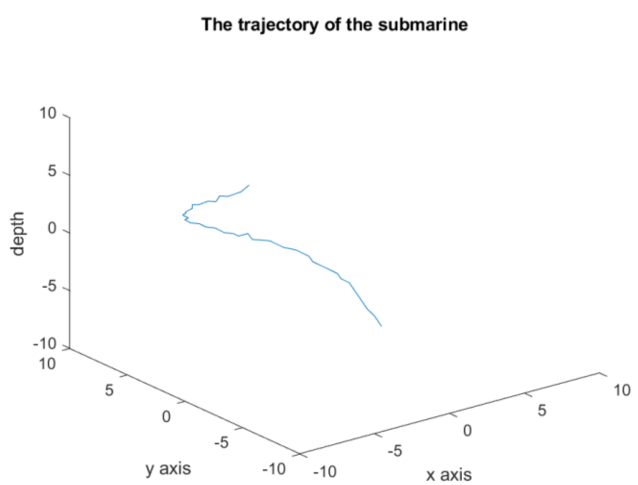
If we want to send our P-8 Poseidon subtracking aircraft to hit the submarine, we need to know the final location of it, which is made up by the last X and Y coordinate in the arrays mentioned above. It is (-5, 0.9375). Since the submarine can go underwater (z coordinate) but the airplane cannot, we only care the x and y coordinates.

Section IV. Computational Results

Center frequency:

Kx	Ky	Kz
5.3407	-6.9115	2.1991

The trajectory of the submarine:



The final location of the submarine:

X	Y
-5	0.9375

Section V. Summary and Conclusions

In this assignment, we utilize the Higher dimensional Fourier Transform, averaging and filtering the signal to get the information we want. In the case that we denoised the data and found the center frequency, we successfully locate the submarine and find its trajectory using the acoustic signature. However, we must rescale the frequencies since the Fast Fourier Transform assumes 2π periodic signals. We also need to remember to shift the function to the correct center when we transform between domains.

Appendix A. MATLAB functions used and brief implementation explanation

- *fftn - N-D fast Fourier transform*

This MATLAB function returns the multidimensional Fourier *transform of* an N-D array using a fast Fourier transform algorithm.

- *ifftn - Multidimensional inverse fast Fourier transform*

This MATLAB function returns the multidimensional discrete inverse Fourier transform of an N-D array using a fast Fourier transform algorithm.

- *fftshift - Shift zero-frequency component to center of spectrum*

This MATLAB function rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.

- *ifftshift - Inverse zero-frequency shift*

This MATLAB function rearranges a zero-frequency-shifted Fourier transform Y back to the original transform output.

- *ind2sub - Subscripts from linear index*

This MATLAB function returns the matrices I and J containing the equivalent row and column subscripts corresponding to each linear index in the matrix IND for a matrix of size siz.

- *plot3 - 3-D point or line plot*

This MATLAB function plots coordinates in 3-D space.

Appendix B. MATLAB codes

```
% Clean workspace
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);

ave = zeros(1,n);
for j=1:49
Un(:,:,:)=reshape(subdata(:,j),n,n,n);
Unt = fftn(Un);
ave = ave+Unt;
% M = max(abs(Un),[],'all');
% subplot(2,1,1)
% close all, isosurface(X,Y,Z,abs(Un)/M,0.7)
% axis([-20 20 -20 20 -20 20]), grid on, drawnow
end
ave = abs(fftshift(ave))/49; % Averaging over realizations
M = max(abs(ave),[],'all');
[Mx,My,Mz] = ind2sub(size(ave),find(abs(ave)==M)); % Find the cooridinates of
the max
cx = Kx(Mx,My,Mz); % Find the center frequency
cy = Ky(Mx,My,Mz);
cz = Kz(Mx,My,Mz);
```

```
tau = 0.2;
filterX = exp(-tau*(Kx-cx).^2); % Define the filter
filterY = exp(-tau*(Ky-cy).^2);
filterZ = exp(-tau*(Kz-cz).^2);
XX = zeros(49,1);
YY = zeros(49,1);
ZZ = zeros(49,1);
for j=1:49
Un(:,:,:)=reshape(subdata(:,j),n,n,n);
Unt = fftshift(fftn(Un));
unft = filterX.*filterY.*filterZ.*Unt; % Apply the filter to the signal in
frequency space
unf = ifftn(fftshift(unft)); % Shift it back to spatial space
M = max(abs(unf),[],'all');
[Mx,My,Mz] = ind2sub(size(unf),find(abs(unf)==M));
XX(j,1) = X(Mx,My,Mz);
YY(j,1) = Y(Mx,My,Mz);
ZZ(j,1) = Z(Mx,My,Mz);
end
plot3(XX,YY,ZZ) % Plot the trajectory of the submarine
axis([-10 10 -10 10 -10 10])
title('The trajectory of the submarine');
xlabel('x axis')
ylabel('y axis')
zlabel('depth')
```

Github repository: https://github.com/chenxr666/Amath482_Assignments.git