AMATH 482 Homework 1

Jiatian Xie

January 26, 2020

Abstract

The goal of this assignment is to find a submarine in the Puget Sound using noisy acoustic sound. We already collected the set of data in 24 hours, but the data contains noise. In order to denoise the data, I firstly tried to find the center frequency generated by the submarine through averaging the spectrum. Then, I set up a 3D Gaussian filter to denoise the data and find the coordinates of x,y,z based on the denoised data. At last, I got the possible position of the submarine from the path.

1 Introduction and Overview

At first, I would like to introduce the question more precisely. When we are hunting for a submarine, we are hardly to detect them correctly because of the noise. Here, we obtained the acoustic data over a 24-hour period in half-hour increments. But, because of the moving submarine, we have to determine its path during the time period. Also, the collected data included 49 times(realizations) and in each time, there are three dimensions and each dimension contained 64 elements(coordinates).

To sum up, what we need to do can be divided into two steps:

- denoise the whole data
- find the path of the submarine

1.1 Denoise the Data

It is obvious that we need a filter to clean the data firstly. Therefore, we do need k_0, which is the center frequency of the spectrum.

1.1.1 Frequency Signature

There are several ways to determine the frequency signature(center frequency) and I did this by averaging the spectrum. Specifically, in each realization, I am able to rearrange the array to the 3D coordinates. Therefore, we can get the average frequency for each coordinates and find the center frequency, which is the maximum frequency.

1.1.2 Clean the Data Around the Frequency Signature

After find the center frequency, we need to filter the data. What I did was creating a 3D Gaussian filter and clean the noise in each realization.

1.2 Find the Path of the Submarine

Now, we have a set of clean data. In order to find the path, I found the coordinate of each realization and use plot3 to plot the path. Thus, the final position was the place where we can send our P-8 Poseidon subtracking aircraft.

2 Theoretical Background

As we learned from our textbook [1], **Fourier** introduced the concept of representing a given function $f(x)$ by a trigonometric series of sines and cosines:

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

Also, we learned **Fourier transform(FT)** and**inverse Fourier transform (IFT)** from the lecture.

$$\widehat{F'(x)} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ikx} f'(x) dx = ik\widehat{F(x)}$$
$$\widehat{f'(x)} = ik\widehat{f(x)}$$
$$\widehat{F^{(n)}(x)} = (ik)^n \widehat{f(x)}$$

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} \widehat{f(x)} dx = ik\widehat{F(x)}$$
$$\widehat{f'(x)} = ik\widehat{f(x)}$$
$$\widehat{F^{(n)}(x)} = (ik)^n \widehat{f(x)}$$

Then, I used a 1D **filter** that is given by Gaussian.

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

and $\tau > 0$ is the width of the filter, k_0 is the center.

3 Algorithm Implementation and Development

3.1 average spectrum

Firstly, I would like to show my steps to average the spectrum and find the center frequency

1. Load the data
2. Set the spatial domain and Fourier modes
3. Using linespace to get 64 spaces in [-L,L]. Also, we must rescale the frequencies by $2\pi/L$ since the FFT assumes 2π periodic signals.
4. Using `meshgrid` to get X,Y,Z and Kx,Ky,Kz
5. See algorithm 1 for how to average Un
6. Do the average `Uave = abs(fftshift(Uave))/49`

Now, we have the average frequency for each cooridnate. Using `isosurface(Kx,Ky,Kz,abs(Uave)/M,0.7)`, where M is max value of the absolute value in average frequency. Thus, I am able to see the frequency signature, which can be the highest(maximum) point in the graph.

3.2 denoise the data

Since we have determined the center frequency and the Gaussian filter for 1D, here, we can denoise the data in each dimension.

1. Set τ
2. Define the filter in each dimension as following, where K_0 equals to the center frequency we found previously: $g = exp(-tau * (K - K_0).^2)$
3. See algorithm 2 for how to find the coordinates in each realization through filtering the data

Therefore, we can get a 3*49 matrix that contains the coordinates of the max frequency in each time period.

Algorithm 1: Average Algorithm

Import data from `subdata.mat`
for $j = 1 : 49$ **do**
  Extract measurement j from `subdata`
  Reshape it to 64*64*64 array
  transform the data into frequency domain
  `Uave = Uave + Un;`
end for

Algorithm 2: Denoise the Data

Create an empty 3*64 matrix to store the path in each realization
for $j = 1 : 49$ **do**
  Reshape Un to 64*64*64 array
  Dot multiply gx,gy,gz with `fftshift(fftn(Un))` since we always denoise data in frequency domain
  IFFT Un in frequency domain into the time domain
  Find the coordinates of the maximum frequency and store it into the path matrix
end for

4 Computational Results

Thus, the frequency signature I got from averaging the spectrum was in Figure 1. The highest point of the figure should be the center frequency. Hence, I estimated the center frequency being (4.5,-7,3).

Next, see Figure 2 which is plotted by plot3 function and represented the path of the submarine in three dimension. Obviously, the submarine was getting down with time.

At last, see Figure 3 and Table 1. We were supposed to determine where to send the subtracking aircraft. In other words, we only care about the xy-plane. Thus, I only extracted the x-coordinates and y-coordinates and see what happened. In the last time realization, I found the coordinates of the submarine was (-5,0.9375). Therefore, we can send my P-8 Poseidon subtracking aircraft there, where is closest to the submarine.

nth Time period	X-coordinate	Y-coordinate
41	-6.875	3.75
42	-6.875	3.125
43	-6.875	2.8125
44	-6.5625	2.5
45	-6.25	2.1875
46	-6.25	1.875
47	-5.9375	1.5625
48	-5.3125	1.25
49	-5	0.9375

Table 1: The coordinates in xy-plane after 20 hours

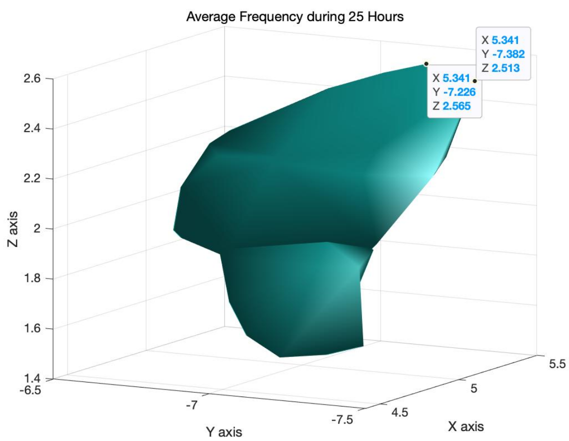

Figure 1: Here is the graph of the average frequency during 25 Hours.

5 Summary and Conclusions

The goal of this assignment is hunting for a submarine in the Puget Sound using noisy acoustic data. Basically, we need to get a clean data that represented 3D coordinates of the submarine. In order to denoise the data through filter, we must get the center frequency generated by the submarine. Then, we are able to clean the data and determine the path of the submarine. The final 2D coordinate I got was (-5,0.9375).

Actually, this assignment was a kind of difficult for me since I never tried 3D problem both in class or outside the class. However, I still enjoy solving this fun problem. The plot illustrated the path of a moving submarine: it moved from deep to shallow.

References

[1] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

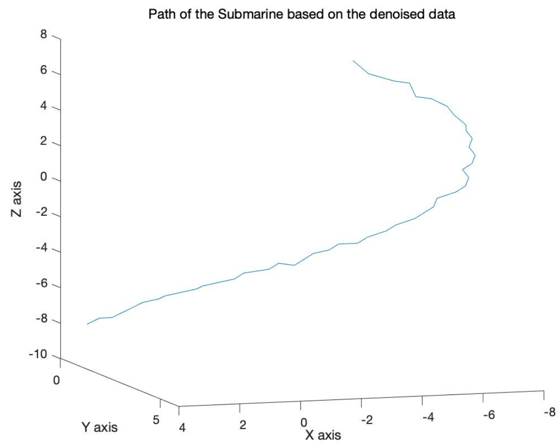

Figure 2: Here is the possible path of submarine in three dimension

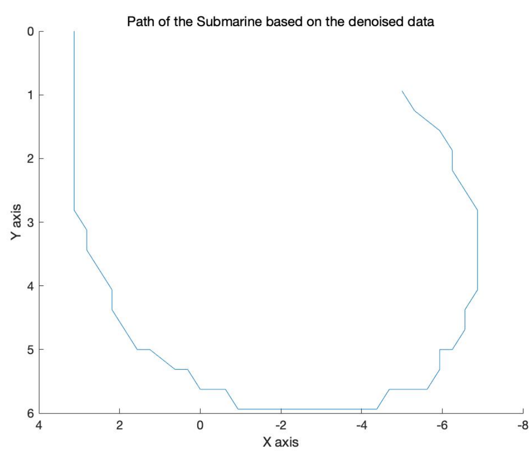

Figure 3: Here is a possible path of submarine in xy-plane

Appendix A MATLAB Functions

Add your important MATLAB functions here with a brief implementation explanation. This is how to make an **unordered** list:

- `y = linspace(x1,x2,n)` returns a row vector of `n` evenly spaced points between `x1` and `x2`.
- `[X,Y] = meshgrid(x,y)` returns 2-D grid coordinates based on the coordinates contained in the vectors `x` and `y`. X is a matrix where each row is a copy of `x`, and `Y` is a matrix where each column is a copy of `y`. The grid represented by the coordinates `X` and `Y` has `length(y)` rows and `length(x)` columns.
- `Un(:,:,:)=reshape(subdata(:,j),n,n,n)` returns a 3D coordinates that are reshaped from an array.
- `Y = 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.
- `Y = 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. rearranges a Fourier transform X by shifting the
- `Y = fftshift(X)` rearranges a Fourier transform X by shifting the zero-frequency component to the center of the array.
- `[M,I] = max(A)` returns the index into the operating dimension that corresponds to the maximum value of A for any of the previous syntaxes.

Appendix B MATLAB Code

```
clear all; close all; clc
load subdata.mat

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);

% Q1
Uave = zeros(n,n,n);

for j=1:49
    Un(:,:,:)=reshape(subdata(:,j),n,n,n);
    Un = fftn(Un);
    Uave = Uave + Un;

    % close all, isosurface(X,Y,Z,abs(Un)/M,0.7)
    % axis([-20 20 -20 20 -20 20]), grid on, drawnow
    % pause(1)
end

Uave = abs(fftshift(Uave))/49;
M = max(abs(Uave),[],'all');
close all,
figure(1)
isosurface(Kx,Ky,Kz,abs(Uave)/M,0.7)
grid on, drawnow
title('Average Frequency during 25 Hours')
xlabel('X axis'), ylabel('Y axis'), zlabel('Z axis')

%% Q2
tau = 0.5;
gx = exp(-tau*(Kx-4.5).^2);
gy = exp(-tau*(Ky+7).^2);
gz = exp(-tau*(Kz-3).^2);
path = zeros(3,49);
for j = 1:49
    Un(:,:,:)=reshape(subdata(:,j),n,n,n);
    Uf = gx.*gy.*gz.*fftshift(fftn(Un));
    Ut = ifftn(Uf);
    [mxv,idx] = max(abs(Ut(:)));
    path(1,j) = X(idx);
    path(2,j) = Y(idx);
    path(3,j) = Z(idx);
end
figure(2)
plot3(path(1,:),path(2,:),path(3,:));
title('Path of the Submarine based on the denoised data')
xlabel('X axis'), ylabel('Y axis'), zlabel('Z axis')

%% Q3
xy_coordinates = path(1:2,:);
final_p = xy_coordinates(:,end)
```


Listing 1: My code of solving this problem