AMATH 482 Homework 1

Jerry Wen

January 24, 2021

Abstract

Here, we are hunting for a submarine by using frequency data that we collected in a noisy ocean. We collected data over a 24-hour period in half-hour increments and using algorithms that we learned in class, we will find out where the submarine is at all times and in the end find its final location.

1 Introduction and Overview

A submarine moves in a ocean and constantly sends out a signal that we collected. We are asked to average the spectrum to determine the frequency signature generated by the submarine so we can accurately denoise it. Afterwards, we filter around the center frequency that we determined and discover the path of the submarine. Finally, we will get the final x and y coordinates of the submarine.

1.1 Averaging The Spectrum

Because we have collected data over a 24-hour period in half-hour increments, we have a total of 49 instances of time. We'd want bring each instance into the frequency domain, and calculate the average. After getting the average, we would want to find the maximum value in the average and the corresponding index. This maximum corresponds to the center frequency. Then, we calculate the exact coordinates of this frequency.

1.1.1 Filtering The Signal

With the center frequency known, we filter our signal using the Gaussian. We are able to choose which tau to use and through some experimenting, we choose the one that makes the path the clearest.

1.1.2 Finding The Submarine

Now that know the path, we can just find the coordinates of the submarine by looking for where it is at the final instance of time.

2 Theoretical Background

As we learned from lecture, we would want to use the Gaussian to filter around a central frequency.

$$F(k) = e^{-\tau(k-k_0)^2} \tag{1}$$

However, we're in a 3-D space, therefore we need to change the equation to calculate off each point in x, y, and z.

$$F(kx, ky, kz) = e^{-\tau((kx-kx_0)^2+(ky-ky_0)^2+(kz-kz_0)^2)} \tag{2}$$

You can reference numbered equations, figures, tables, algorithms, and code like this: Equation **??**, etc.

3 Algorithm Implementation and Development

Here is the main idea on how to implement and develop the code.

1. We were given a beginning code sample that showed us how to load the data and how to start analyzing it.

2. From here, we take the average of the data in frequency space to find the center frequency.

3. After finding the center frequency we filter with the Gaussian by multiplying all of our data with the filter.

4. With the unfiltered signal, we can track the path of the submarine.

Algorithm 1: Submarine Algorithm

Import data from `subdata.mat`
Create the correct domain to work in with specific L and n. We also set y and z to be the same as we are looking at a 3D cube
Create the correct fourier domain
for $j = 1 : 49$ **do**
  Reshape the data into a n by n by n cube
  Transform it using FFT
  Add this to the total
end for
Get the average by dividing the total by 49
Find the max in the average
Find the coordinates of the max
Create the Gaussian Filter with the coordinates you made using the coordinates from the max by plugging them into the initial fourier domain
for $j = 1 : 49$ **do**
  Reshape the data into a n by n by n cube
  Apply the fourier transform and shift it
  Apply the filter to the frequency data
  Find the max to this data
  Find the coordinates of the max. This will be the location of the submarine for this instance of time
end for
Finally we should plot the entire path of the submarine

---

4 Computational Results

Here are our results of the path of the submarine:

t	X	Y	Z
1	33	43	7
2	34	43	8
3	35	43	9
4	37	43	11
5	38	43	11
6	39	43	12
7	49	43	13
8	41	43	14
9	42	42	15
10	43	42	16
11	44	42	17
12	45	41	18
13	46	40	19
14	47	40	20
15	47	39	21
16	48	37	22
17	49	37	23
18	50	36	23
19	50	34	25
20	51	33	26
21	51	32	27
22	52	30	27
23	52	29	29
24	52	27	30
25	52	25	30
26	52	24	31
27	52	23	32
28	52	21	33
29	52	20	34
30	52	19	35
31	51	18	37
32	51	16	37
33	50	15	39
34	50	14	39
35	49	14	40
36	48	13	41
37	48	12	42
38	47	11	43
39	46	11	44
40	46	11	45
41	44	11	46
42	43	11	47
43	42	11	47
44	41	12	49
45	40	12	49
46	39	13	51
47	38	15	51
48	37	16	53
49	36	17	54

Table 1: Coordinates of the Submarine at each time

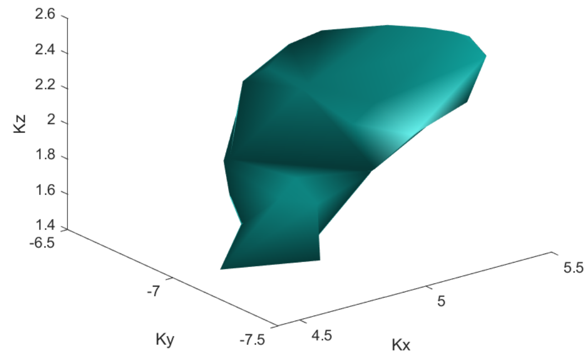

Figure 1: An image of the average in frequency space.

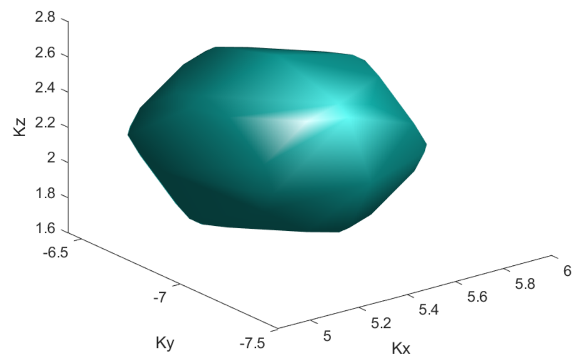

Figure 2: An image of the Gaussian filter we're going to use. This has a τ of 1.5. Described in (2)

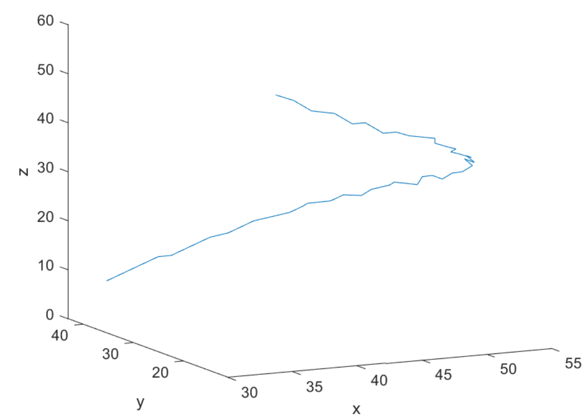

Figure 3: An image of the path of the submarine

5 Summary and Conclusions

We see that the submarine has stopped at (36, 17, 54) and therefore we should send in the subtracking aircraft at that location.

Through this project, we've seen the power of taking noisy data and by careful filtering, we are able to make sense out of it.

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.
- `[row,col] = ind2sub(sz,ind)` returns the arrays `row` and `col` containing the equivalent row and column subscripts corresponding to the linear indices ind for a matrix of size `sz`. Here `sz` is a vector with two elements, where `sz(1)` specifies the number of rows and `sz(2)` specifies the number of columns.
- `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 = fftshift(X)` rearranges a Fourier transform `X` by shifting the zero-frequency component to the center of the array.

Appendix B MATLAB Code

Add your MATLAB code here. This section will not be included in your page limit of six pages.

```
% Clean workspace
clear; close all; clc

load subdata.mat

L = 10;
n = 64;
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);

% average in frequency domains
Utnave = zeros(n,n,n);
for j = 1:49
    Un(:,:,:) = reshape(subdata(:,j),n,n,n);
    Utn = fftn(Un);
    Utnave = Utnave + Utn;

end
Utnave = fftshift(Utnave)/49;
M = max(abs(Utnave),[],'all');
% Plot central frequency
isosurface(Kx,Ky,Kz,abs(Utnave)/M,0.7);
xlabel('Kx')
ylabel('Ky')
zlabel('Kz')
%%
% Create the Gaussian Filter
[Max, Ind] = max(abs(Utnave(:)));
[Ix, Iy, Iz] = ind2sub(size(Utnave), Ind);
Kx0 = Kx(Ix, Iy, Iz);  %  5.34
Ky0 = Ky(Ix, Iy, Iz);  % -6.91
Kz0 = Kz(Ix, Iy, Iz);  %  2.19
tau = 1.5;  % Can change
filter = exp(-tau * ((Kx - Kx0).^2 + (Ky - Ky0).^2 + (Kz - Kz0).^2));
isosurface(Kx,Ky,Kz,filter,0.7);
xlabel('Kx')
ylabel('Ky')
zlabel('Kz')
%%
% Find and plot path of the submarine
P = zeros(3,49);
for j = 1:49
    Un(:,:,:)= reshape(subdata(:,j),n,n,n);
    Unt = fftshift(fftn(Un));
    Untf = filter .* Unt;
    Unf = ifftn(fftshift(Untf));
    [Max, Ind] = max(abs(Unf(:)));
    [Ixx, Iyy, Izz] = ind2sub(size(abs(Unf)), Ind);
    P(:,j) = [Ixx, Iyy, Izz];
end
plot3(P(1,:), P(2,:), P(3,:));
xlabel('x')
ylabel('y')
zlabel('z')
```


Listing 1: Code for this homework.