Amath482 HW1

Name: Leqi Wang

Date: Jan21, 2021

Abstract: *Investigate the application of Fourier transform in 3D space by analyzing a signal emitted by a submarine over a 24-hour period in half-hour increments.*

1. Introduction and Overview:

In the first assignment, we investigated the application of Fourier transform in 3D space by analyzing a signal emitted by a submarine over a 24-hour period in half-hour increments. Given the data, we are asked to analyze center frequency by averaging the spectrum and filter the data around the center frequency to get the path of this submarine. We are also asked to simulate aircraft subtracting by calculating by obtaining the x-y coordinates of the submarine.

2. Theoretical Background

a) **Averaging:** According to lecture 3, white noise has mean frequency 0 and 0 variance, and the shape of the incoming signal will not change. After averaging through many realizations for each time t, the frequency of white noise will (ideally) be squeezed to zero and leave the variance, which is the signal we want to detect, in the frequency domain.

a) **Filtering:** According to lecture 2, if the frequency (or frequency range) of desired signal is known, then define a filter to cut out white noise in the frequency domain. If filter is applied correctly, desired signal will be also filtered out in time domain using inverse Fourier transform.

3. Algorithm Implementation and Development

a) **Q1:**

i. **Algorithm:** Add the frequency up for 49 realizations to cancel the white noise.

ii. **Development**: Define a variable to store the averaged data in frequency domain. In the loop, for each snapshot, reshape acoustic data sets, each contains the coordinates information in time domain. Transform *Un* into frequency domain using then store the sum in *Utnave*. Then calculate average and sort averaged data in right order. Use draw the graph in frequency dimension and use Mk (the maximum of averaged frequency) to scale the graph so that the desired signal is more evident.

b) **Q2:**

i. **Algorithm:** Apply filter in the frequency domain and transform back to time domain to visualize where(spatially) the desired frequency occurred in each time slot. I defined the spatial coordinates of the submarine to have maximum absolute value, so that I don't get a range of where the submarine could be.

ii. **Development:**

1. Define filter: From figure 1, the range of the center frequency is about from x[4.4, 5.5],y[-7.5,-6.5],z[1.4,2.6] (obtained from graph), therefore I define the filter as gaussian with same range in each direction: in x-y-z direction, centered at [5,-7, 2] with tau = 0.5 for each direction.
2. For each time slot [1:49], first transform time domain to frequency domain and apply filter. After inverse the filtered frequency back into time domain, record the maximum of the absolute value of *Unf*, extract the indices number of maxima in *Unf*, and convert it into subscripts. Then I use this index to extract the x-y-z value in [X, Y, Z] coordinate scale and store it in a location matrix, each column representing the x-y-z value.

c) **Q3:** Form a table using x-y value in location data set from Q2(table1).

4. Computational Results

a) Q1: The center frequency for the submarine plot:

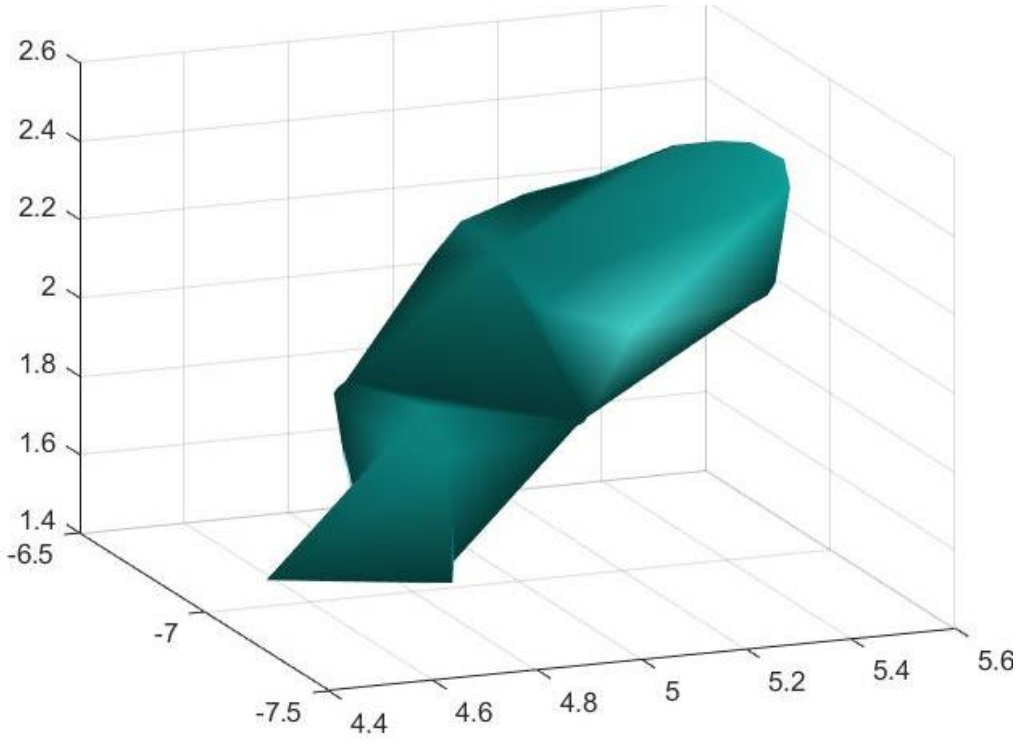

Figure 1: Center Frequency

b) **Q2: The track of submarine:**

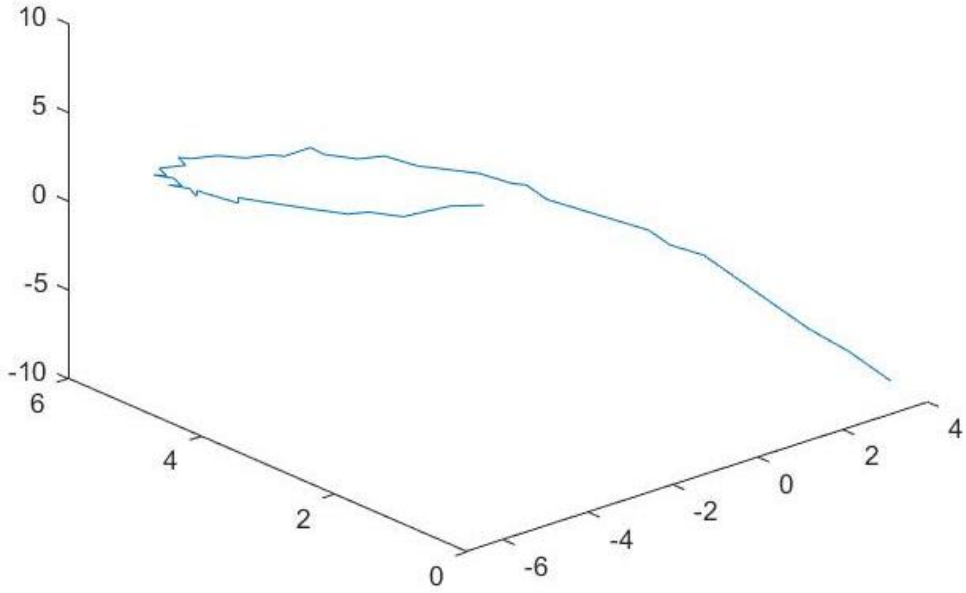

Figure 2: submarine track

c) **Q3: X-Y table:**

x	y
3.125	0
3.125	0.313
3.125	0.625
3.125	1.25
3.125	1.563
3.125	1.875
3.125	2.188
3.125	2.5
3.125	2.813
2.8125	3.125
2.8125	3.438
2.5	3.75
2.1875	4.063
1.875	4.375
1.875	4.688
1.5625	4.688
1.25	5
0.625	5.313
0.3125	5.313
0	5.625
-0.625	5.625
-0.938	5.938
-1.25	5.938

-1.875	5.938
-2.188	5.938
-2.813	5.938
-3.125	5.938
-3.438	5.938
-4.063	5.938
-4.375	5.938
-4.688	5.625
-5.313	5.625
-5.625	5.313
-5.938	5.313
-5.938	5
-6.25	4.688
-6.563	4.688
-6.563	4.375
-6.875	4.063
-6.875	4.063
-6.875	3.438
-6.875	3.438
-6.875	3.125
-6.563	2.5
-6.25	2.188
-6.25	1.875
-5.938	1.563
-5.313	1.25
-5	0.938

Table 1: x-y coordinates of submarine

5. **Summary and Conclusions:** It is possible to detect the position of an object through filtering and averaging the acoustic signal we receive.

6. **Appendix A. MATLAB functions used and brief implementation explanation**

   a) linspace(x1,x2,n): generates n points between x1 and x2.

   b) meshgrid(x,y,z) returns 3-D grid coordinates defined by the vectors x, y, and z.

   c) zeros(n, m): initialize an array of 0 with size n x m.

   d) reshape(subdata(:,j),n,n,n);

      i. subdata(: , j): extract spatial data in provided subdata's jth column with all the rows.

      ii. reshape(): reshape the subdata in time = j into a 64x64x64 matrix, with structure[x,y,z]

   e) fftn():Apply Fourier transform for multi-dimensional data.

   f) fftshift(): Sort the data after toggling between time and frequency domain.

   g) exp(-tau*(Kx - kx0).^2) .* exp(-tau*(Ky - ky0).^2) .* exp(-tau*(Kz - kz0).^2); define gaussian filter in 3-d (frequency)space.

h) max(abs(Unf),[],'all'): finds the maximum over all elements of Un.

i) ind2sub(size(Unf),find(abs(Unf)== maxCor)):

i. find(): find the indices in Unf where the absolute value of Unf equals maxCor, return an integer.

ii. [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. Here sz is a vector with n elements that specifies the size of each array dimension.

7. 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);

% averaging
Utnave = zeros(n,n,n);
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)
Utn = fftn(Un);
Utnave = Utnave + Utn;
end

Utnave = fftshift(abs(Utnave) / 49);
Mk = max(abs(Utnave),[],'all');
close all, isosurface(Kx,Ky,Kz,Utnave/Mk,0.7);
grid on, drawnow

%% Filtering
% The range of the center frequency is about from x[4.4,
% 5.5],y[-7.5,-6.5],z[1.4,2.6]
% Select the filter function:
```

```
tau = 0.5;
kx0 = 5;
ky0 = -7;
kz0 = 2;
filter = exp(-tau*(Kx - kx0).^2) .* exp(-tau*(Ky - ky0).^2) .*    exp(-tau*(Kz - kz0).^2);
location = zeros(49,3);

for j=1:49
Un(:,:,:)=reshape(subdata(:,j),n,n,n);
Utn = fftshift(fftn(Un));
Utnf = filter.*Utn; % Apply the filter to the signal in frequency space
Unf = ifftn(Utnf); % inverse fourier transform to time domian.
maxCor = max(abs(Unf),[],'all');
[xt,yt,zt] = ind2sub(size(Unf),find(abs(Unf)== maxCor));
location(j,:) = [X(xt, yt, zt), Y(xt, yt, zt), Z(xt, yt, zt)];
end
plot3(location(:, 1), location(:, 2), location(:, 3));

%% table of x and y coord
xy = table(location(:,1), location(:,2));
```