AMATH 582 HW1: Save your dog!

Huasong Zhang
Department of Applied Mathematics
University of Washington
Seattle, WA 98195
huasongz@uw.edu

Abstract

This project investigates the 20 signals that we received in order to determine the position of marble inside dog's body. To solve this problem, we implemented Fast Fourier Transform in MATLAB. In that way, we were able to easily find the central frequency, and that's where the marble is. We were also able to find the movement projectory of the marble.

1 Introduction and Overview

In our daily life, localizing by using the reflection of signal has been widely used in medical fields and military fields. This problem is about a dog accidentally swallow a marble. The vet used ultrasound to detect the marble and received 20 signals with respect to time. In this problem, the signal we received and analyzed reflects the position and marble inside the dog, so that we can use these information to break and remove the marble, and save the dog.

2 Theoretical Background

The signals received were in time domain. In order to figure out the distribution of frequency and find where is the peak frequency and localize that back in x,y,z coordinate, we need to use Fourier Transform and its inverse to transfer the signal back and forth between time mode and frequency mode.

3 Algorithm Implementation and Development

Firstly, the signals were in spatial domain. We want to first discretized that and then transfer it to Fourier domain. We have periodic boundary conditions. Therefore, the value at last point will be the same as the first one. We only need to take the first n points. So we have the code:

```
L=15; % spatial domain
n=64; % Fourier modes
x2=linspace(-L,L,n+1); x=x2(1:n); y=x; z=x;
```

Then we rescale it to Fourier domain which goes from -2π to 2π. And use fftshift to shift the k back to the right order.

```
k=(2*pi/(2*L))*[0:(n/2-1) -n/2:-1]; ks=fftshift(k);
```

Then we used the meshgrid to build the matrix.

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

Now the axis are already set up and we can start working on signals. We first want to average to de-noise the signals. So we assume the noise has average 0. We build a zero matrix first and then add up all 20 signals in a loop. Since our signals are one dimensional and in spatial domain, we need to first reshape them to three dimensional and then use fftn to shift them to Fourier domain.

```
Uave = zeros(n,n,n);
for j = 1:20
    Utn(:,:,:) = fftn(reshape(Undata(j,:,:),n,n,n));
    Uave = Uave + Utn;
end
```

After adding up all 20 signals, we can fftshift the Uave to the right order and divide it by 20 to get the average signal. After doing this step, the noise should disappear and the major frequency remains.

```
Uave = fftshift(Uave)./20;
```

Then we find the maximum value of Uave and locate it, denoting the indexes as a,b, and c. Then, plug in a, b, and c in Kx, Ky, and Kz, we can find the exact location of the peak frequency in Fourier domain.

```
[val,idx] = max(Uave(:));
[a,b,c] = ind2sub(size(Uave),idx);
xx = Kx(a,b,c);
yy = Ky(a,b,c);
zz = Kz(a,b,c);
CF = sprintf('The central frequency in frequency domain is at %s %d %f.',xx,yy,zz);
```

After we find the central frequency, we want to de-noise by applying filter on the signal. So we can make a Gaussian function which acts on the peak frequency. The function for Gaussian filter is:

```
filter=exp(-0.2*(((Kx-xx).^2)+((Ky-yy).^2)+((Kz-zz).^2)));
```

And we want to track the movement of marble, so we need to save the x, y, and z coordinates throughout the entire domain. So I built three empty matrix in order to save data. In the for loop for time goes from 1 to 20, I first reshaped the 1D signal to 3D, transformed it. Since the filter is already fftshifted, I also need to fftshift the signal. Then I applied the filter on the signal by multiplying them together. Since we want to figure out the movement of marble in spacial domain, I did ifftn to shift the signal back and found the indexes of the location for peaks.

```
for i = 1:20
    dat(:,:,:) = reshape(Undata(i,:),n,n,n);
    Un = fftn(dat);
    Unt = fftshift(Un);
    Unft = filter.*Unt;
    Unf = ifftn(Unft);


    [val,idx] = max(Unf(:));
    [b,a,c] = ind2sub(size(Unf),idx);
    A(i) = a;
    B(i) = b;
    C(i) = c;
end
```

After getting vector A, B, and C, we can use plot3 to plot the movement of marble.

```
plot3(x(A),y(B),z(C))
xlabel('x')
ylabel('y')
zlabel('z')
title('Marble Movement Trajectory')
```

In order to breakup the marble, we need to locate the marble at the 20th location, which is

```
x20 = x(A(end));
y20 = y(B(end));
z20 = z(C(end));
L = sprintf('The 20th location of the marble is at %s %d %f.',x20,y20,z20);
disp(L)
```

4 Computational Results

So for part one we need to find where the central frequency is. If shows in the graph, it will look like this:(Figure 1) The ball-shaped thing on the right side which is larger than all the other pieces in this

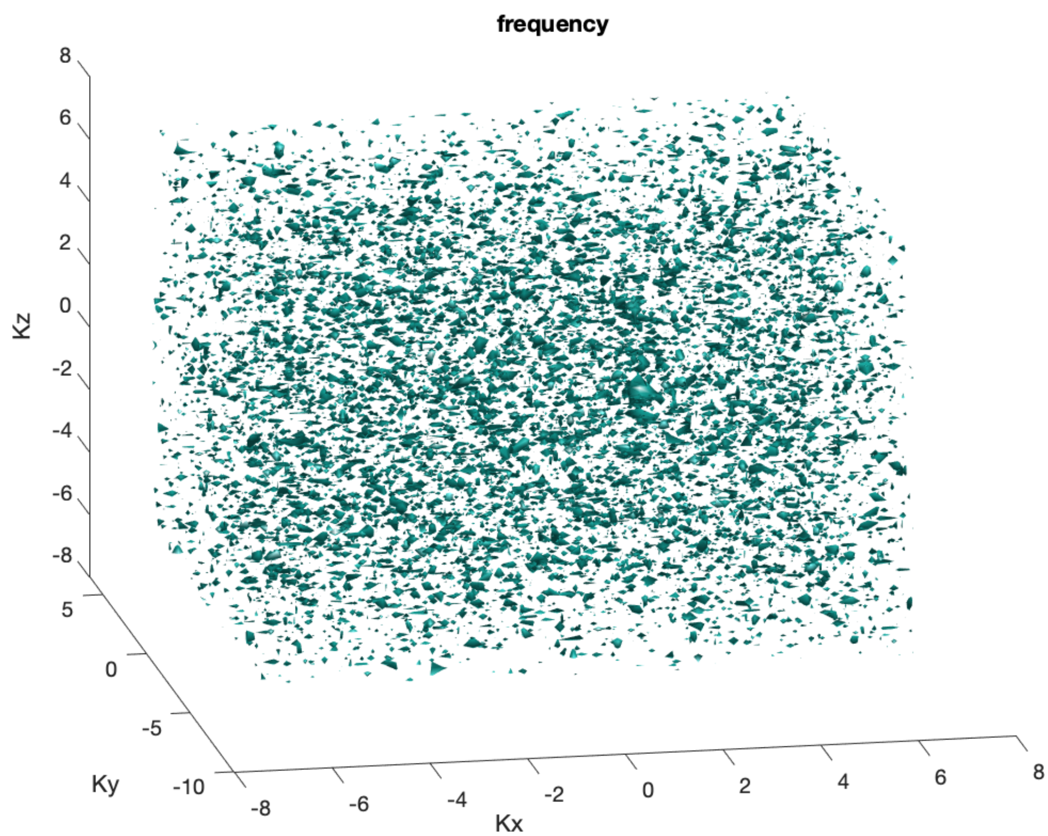

Figure 1: Signal in frequency domain

graph is the central frequency that we are looking for. The result is as following:

```
a = 28
b = 42
c = 33
xx = 1.8850
yy = −1.0472
zz = 0
```

And the last line of the code prints:

```
The central frequency in frequency domain is at 1.884956e+00 −1.047198e+00 0.000000.
```

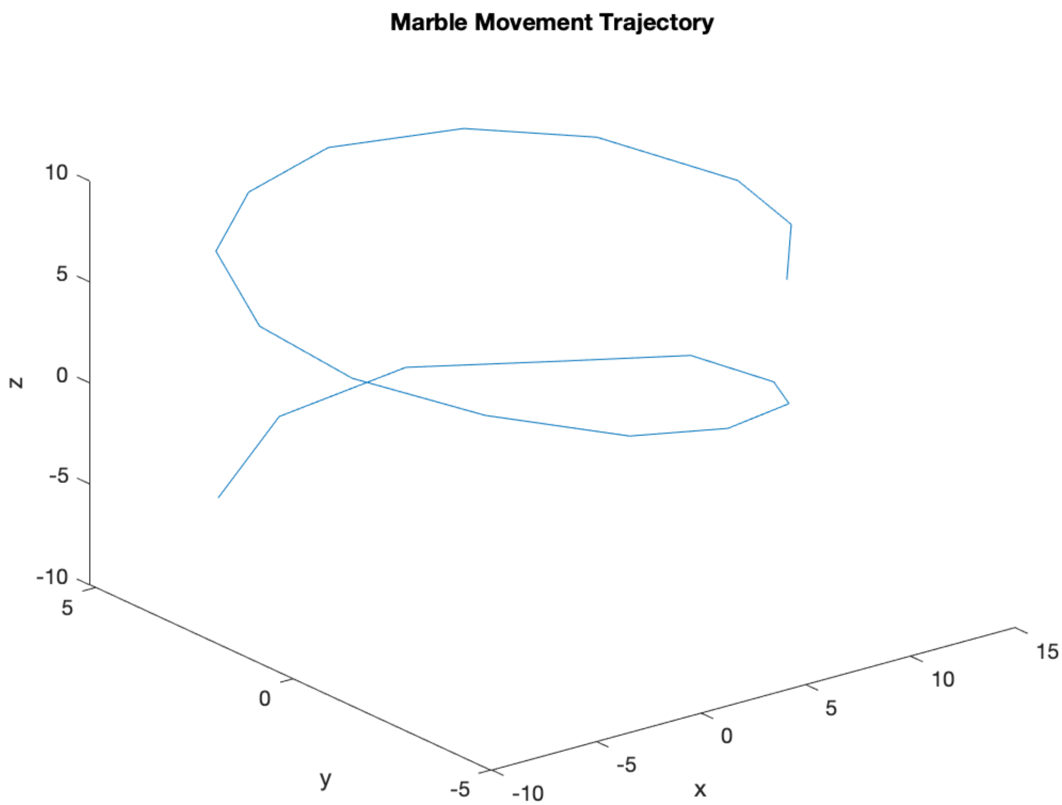

Figure 2: This graph shows the trajectory of how marble moves

After we find the location of the central frequency, we can apply a filter around it so that we can see the signal more clear. The vector of indexes that we found for x, y, and z is:

A = 43 51 55 52 46 36 26 17 12 12
18 27 37 47 53 54 50 42 31 21
B = 23 27 32 38 42 44 43 40 35 30
26 23 23 25 29 34 39 43 44 42
C = 54 54 53 53 52 51 50 49 47 46
44 42 40 38 35 33 29 27 24 20

And the graph of movement of marble is shown in Figure 2. The trajectory is spiral-shaped. The exact location for marble at 20th measurement is: (-5.625000e+00,4.218750e+00,-6.093750). Therefore, in order to breakup the marble, we should focus an intense acoustic wave at that point.

5 Summary and Conclusions

In this problem, since we are trying to breakup the marble inside dog's body, the key point is to find where the marble is. Since we cannot see it directly using our eyes, we can make use of the signal frequency. The place where we have high frequency is where the wave hits the marble; in that way, we can localize the marble. In this problem, we also used two ways to make our signal more clear. One way is to de-noise. However, this method is under the assumption that the mean of noise throughout time is zero. The other way to de-noise is to apply a filter on the signal. This method is more applicable when we know where is the central frequency. Both of the two methods have their advantages and disadvantages. We need to decide which one to use depend on different circumstances. In this problem, one thing that we need to pay more attention to is the domain. Sometimes, we are focusing on frequency and trying to find the central frequency; therefore, we should use the frequency domain. While, sometimes we want to locate where the marble is and in order to break the marble, then, time and location is more important to us. Therefore, we should work on the problem in spacial domain.

Appendix A MATLAB functions used and brief implementation explanation

fft: does Fourier Transform on signal
ifft: does inverse Fourier Transform
fftshift: since fft will switch the left side and right side of the signal and multiply the modes by -1, fftshift is puting the signal back in the right position and take the absolute of the amplitude.
fftn: does Fourier Transform in n dimension
ifftn: inverse Fourier Transform in n dimension

Appendix B MATLAB codes

```
clear all; close all; clc;
load Testdata
L=15; % 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);


Uave = zeros(n,n,n);
for j = 1:20
    Utn(:,:,:) = fftn(reshape(Undata(j,:,:),n,n,n));
    Uave = Uave + Utn;
end
Uave = fftshift(Uave)./20;
[val,idx] = max(Uave(:));
[a,b,c] = ind2sub(size(Uave),idx);
isosurface(Kx,Ky,Kz,abs(Uave)./max(abs(Uave(:))),0.4)
xx = Kx(a,b,c);
yy = Ky(a,b,c);
zz = Kz(a,b,c);
CF = sprintf('The central frequency in frequency domain is at %s %d %f.',xx,yy,zz);
disp(CF)


A = [];
B = [];
C = [];
filter=exp(-0.2*(((Kx-xx).^2)+((Ky-yy).^2)+((Kz-zz).^2)));
for i = 1:20
    dat(:,:,:) = reshape(Undata(i,:),n,n,n);
    Un = fftn(dat);
    Unt = fftshift(Un);
    Unft = filter.*Unt;
    Unf = ifftn(Unft);


    [val,idx] = max(Unf(:));
    [b,a,c] = ind2sub(size(Unf),idx);
    A(i) = a;
    B(i) = b;
```

```
    C(i) = c;
end
figure(2)
plot3(x(A),y(B),z(C))
x20 = x(A(end));
y20 = y(B(end));
z20 = z(C(end));
L = sprintf('The 20th location of the marble is at %s %d %f.',x20,y20,z20);
disp(L)
```