AMATH 482
Homework 1 Report
Fangzheng Sun
1239942

Abstract

This report treats with the given data with noise. I get the center frequency through averaging and use Gaussian filter to remove the noise, and finally apply the result to find the track of the marble in the dog.

Introduction

The dog Fluffy swallowed a marble. The ultrasound gives information of the positions of the marble in 20 measurements. By the internal fluid movement of the dog generates a lot of noise which makes it hard to directly determine the track of the marble. My job is to remove the noise in the data, find the proper track of the marble and break it up at the 20th measurement to save the dog's life.

Theoretical background

The following theories are included in this project:

1. Averaging. In a single snapshot, it is hard to determine the major frequency, but by computing the average of a number of snapshots, we can gradually get clearer views as the average noise in every points tend to zero.
2. Fast Fourier Transform and Inverse Fast Fourier Transform. FFT converts a signal from its original domain (time/space) to a representation in the frequency domain. And iFFT does the inverse job.

3. Gaussian Filter. Given the center frequency, Gaussian filter is used in the frequency domain to help denoise the given frequencies by multiplying with the data in the frequency domain and results in a denoised frequency domain. The Gaussian filter equation is:

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

   Where τ measures the bandwidth of the filter and k = k0 is desired signal.

Algorithm Implementation and Development

1. Creating the domain of frequency. Since FFT requires 2π periodic signals, I rescale the wavenumbers by 2π/L and rename it by "k". In the frequency domain, the k is shifted by FFT and renamed "ks". Additional to the two mesh grids [X, Y, Z] in space domain and [Kx, Ky, Kz] in frequency domain, I create a new mesh grid [KxU, KyU, KzU], which means "upshifted mesh grid in the space domain" for the major frequency.
2. FFT the data and averaging. In a "for" loop, FFT the data of the 20 measurements separately and add them up together in the frequency domain. By calculating the mean of 20 sets of data, I extract the max frequency and store its position as [ii, jj, kk].
3. Applying the Gaussian filter. In 3D, the Gaussian filter is created by multiplying the Gaussian function in each dimension:

$$F(k) = e^{-\tau((x-x_0)^2+(y-y_0)^2+(z-z_0)^2)}$$

   I FFT the noised data and multiply it with the filter in the frequency domain and iFFT it back to space domain (here I set τ to be 2, which is good bandwidth for the data in this project).As a result, all the noises in the given data are removed.
4. Applying the computational result.

Computational Results

Through the averaging process, I got the center frequency in the three dimension:

X: 1.8850

Y: -0.10472

Z: 0

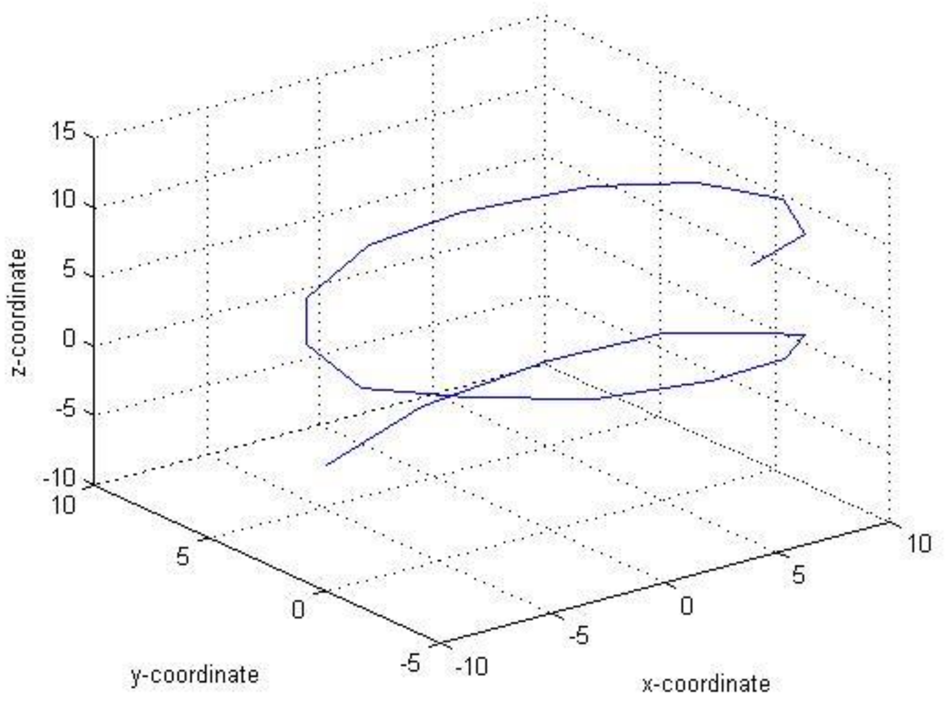

Figure 1: Spatial Position of the Marble

Finally we get the path of the marble through the 20 measurements. At the 20th measurement, the marble should be at (-5.625, 4.2188, -6.0938). And this is the coordinate of the position where an intense acoustic wave should be focused to break up the marble.

Summary and conclusions

In this project, I use the averaging theory, FFT and iFFT and Gaussian filter to remove the noise. Finally, I successfully denoise the data and find the path of the marble. After determining the position of the marble at 20th measurement, I can use acoustic wave to save the dog's life.

Appendix A: MATLAB functions

linspace – Generate linearly spaced vector

fftn – N-D Fast Fourier Transform

fftshift – Shift Zero frequency component to center of spectrum

ifftn – N-D Inverse Fast Fourier Transform

meshgrid – Create rectangular grid in 3-D space

reshape – reshape array to 3-D matrix

ind2sub – Subscripts from linear index

plot3 – 3-D plot

Appendix B: MATLAB code

```
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);
[KxU,KyU,KzU]=meshgrid(k,k,k); % unshifted meshgrid in space domian

UAve = zeros(n,n,n);
for j = 1:20
   Unt = fftn(reshape(Undata(j,:),n,n,n)); % FFT data in each
measurement
   UAve = UAve + Unt; % add up data in frequency domain
end
UAve = abs(UAve)/20; % get the mean
[v,I] = max(UAve(:)); % extract the maximum, which is the main
```

```
frequency
[ii,jj,kk] = ind2sub([n,n,n],I);% returns 3 subscript arrays equal to
I

CentreX = KxU(ii,jj,kk); % Center frequency
CentreY = KyU(ii,jj,kk);
CnetreZ = KzU(ii,jj,kk);

filter = exp(-2.*((KxU-KxU(ii,jj,kk)).^2+(KyU-KyU(ii,jj,kk)).^2
+(KzU-KzU(ii,jj,kk)).^2)); % 3-D Gaussian filter

for j = 1:20
    Unt = fftn(reshape(Undata(j,:),n,n,n)); % FFT data in each
measurement
    Untf = Unt.*filter; % multiply by the filter
    Unf = ifftn(Untf); % transform back to spatial domain
    [v2,I2] = max(Unf(:)); % extract the path with the max number
    [ii2,jj2,kk2] = ind2sub([n,n,n],I2); % returns 3 subscript arrays
equal to I2
    a(j) = X(ii2,jj2,kk2); % x-coordinate
    b(j) = Y(ii2,jj2,kk2); % y-coordinate
    c(j) = Z(ii2,jj2,kk2); % z-coordinate
end
plot3(a,b,c);grid on;
xlabel('x-coordinate'), ylabel('y-coordinate'), zlabel('z-
coordinate');
ans = [a(20), b(20), c(20)]; % final answer - marble position
```