AMATH 582 Homework 1

Daniel Burnham

January 24, 2020

Abstract

The work presented here is motivated by material covered in AMATH 582 Computational Methods For Data Analysis regarding Fourier transform. The premise of the exercise is that a dog swallowed a marble and the veterinarian collected ultrasound measurements of the marble in the intestines. However, the data needs to be filtered and subsequently analyzed in order to determine where to focus an intense acoustic wave to break apart the marble. Averaging of the data transformed in the frequency domain using the Fourier transform will be used to identify the marble frequency signature. This signature will then be used to locate the marble at each point in time within the data by informing the location of a filter in the frequency domain. After this filtering the marble trajectory will be made clear, and the acoustic wave target can be acquired.

1 Introduction and Overview

The data analysis efforts will be divided into three distinct phases. First, the center frequency corresponding to the marble in the Fourier transformed data will be identified. Second, the center frequency will be used to filter the Fourier transformed data for each measurement instance. Third, this filtered data will be inverse transformed to yield the spatial location of the marble for each moment in time.

1.1 Marble Frequency Signature

The frequency signature of the marble will be isolated from the spatial data by first transforming the data into the frequency domain and then averaging the measurements together to diminish noise.

1.2 Data Filtering

The center frequency of the marble will be used to position a filter in the frequency domain to isolate the marble signature for each measurement. This filtered frequency spectrum can then be inverse transformed to indicate the spatial location of the marble.

1.3 Marble Location

The maximum value of the filtered and subsequently inverse transformed data for each measurement will indicate where the marble is located in space. The visualization of this location for each measurement instance will illustrate the trajectory of the marble in time and will allow for the acoustic wave to be targeted to the final marble position.

2 Theoretical Background

The main concept explored with this work is the Fourier transform. The Fourier transform is an extension of the Fourier series which represents a given function $f(x)$ as sums of cosines and sines.

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

This operation is especially useful in the context of signal analysis as it represents signal component frequencies for selective filtering.

To implement the Fourier transform computationally, the fft(x) function in MATLAB will be used. There are certain properties of the fft(x) function that need to be considered. First, the function is only "fast" when passed data that is 2^n discretized. Second, the algorithm shifts the data so that $x \in [0, L] \to [L, 0]$ and $x \in [L, 0] \to [0, L]$. Third, a 2π periodic domain is assumed.

3 Algorithm Implementation and Development

The main steps of the algorim are as follows:

1. Average frequency spectrum of each measurement instance to identify center frequency of marble.
2. Filter data in the frequency domain around marble frequency signature. Inverse transform data to find marble trajectory and final position.

Algorithm 1: Find center frequency

```
Import data from Testdata.mat
for j = 1 : 20 do
    Extract measurement j from Undata
    Apply Fourier transform
    Add to sum of transformed data
end for
Divide sum of transformed data by number of measurements
Find maximum value of the average transformed data
```

Algorithm 2: Filter Data

```
Create filter around marble frequency
for j = 1 : 20 do
    Apply filter to each measurement
    Apply inverse transform
    Find maximum value corresponding to the marble position at that time step.
end for
Visualize trajectory
Find final marble position
```

4 Computational Results

See Figure 1 for the result of averaging the Fourier transformed measurements. See Figure 2 for the determined marble trajectory. The final marble position was determined to be $(-5.6250, 4.2188, -6.0938)$ in the spatial domain.

5 Summary and Conclusions

The averaging of the transformed ultrasound measurements successfully diminished the noise in the signal and allowed for the marble frequency signature to be identified. Inverse transformation of the data filtered around this frequency resulted in a spiraling marble trajectory.

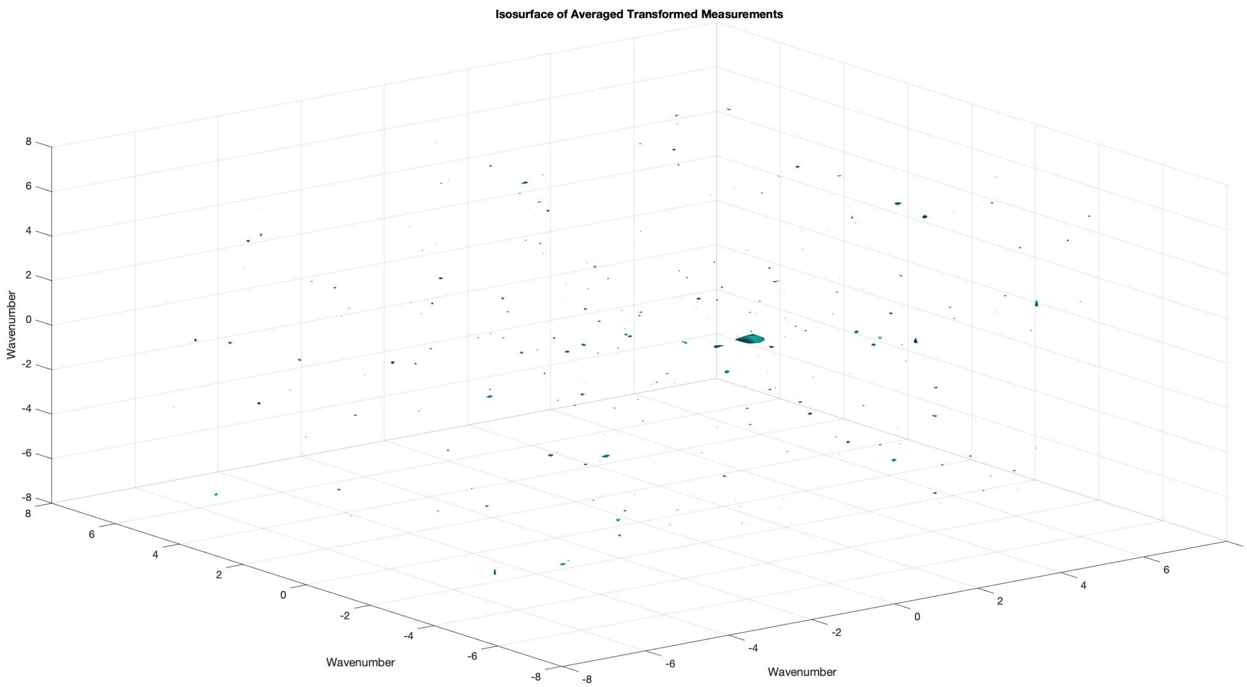

Figure 1: This is an isosurface plot (see Appendix A) of the marble central frequency after averaging measurements in the frequency domain. The frequency signature of the marble is $(1.8850, -1.0472, 0)$

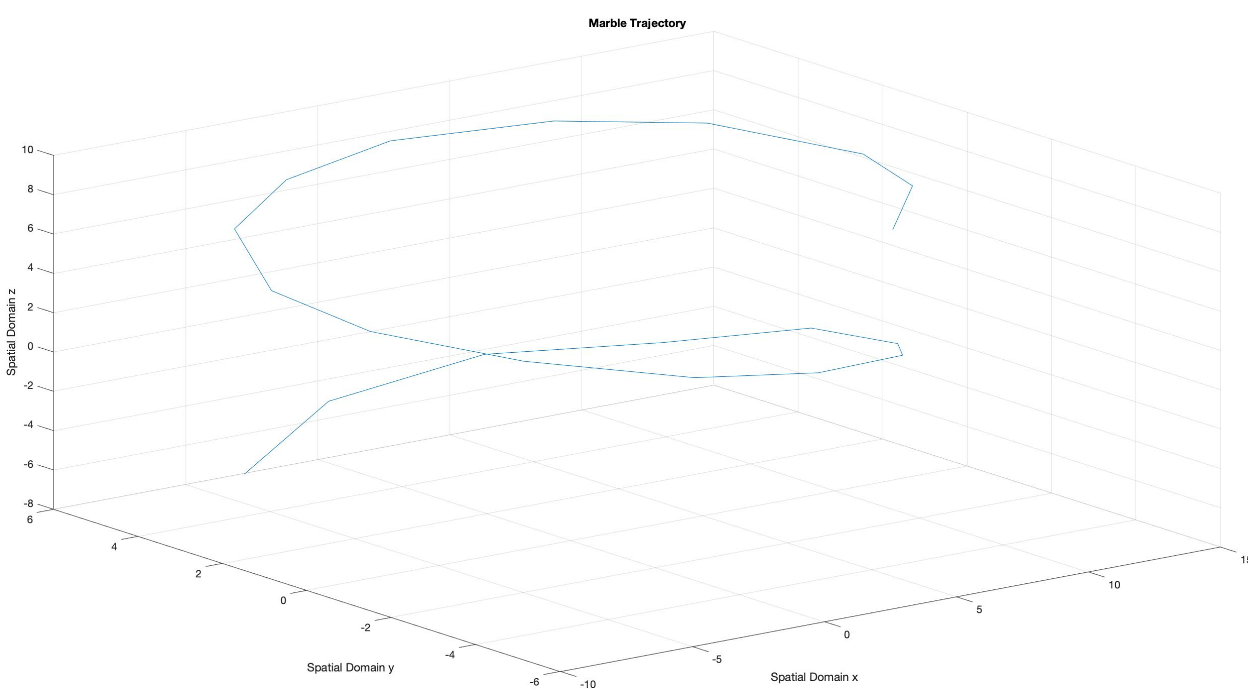

Figure 2: This is a 3D plot of the marble trajectory after filtering the data around the central frequency and using the inverse transform. The final marble position is $(-5.6250, 4.2188, -6.0938)$

Appendix A MATLAB Functions

- `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.

- `X = ifftn(Y)` 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.

- `[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.

- `fv = isosurface(X,Y,Z,V,isovalue)` computes isosurface data from the volume data V at the isosurface value specified in isovalue. That is, the isosurface connects points that have the specified value much the way contour lines connect points of equal elevation.

- `[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.

Appendix B MATLAB Code

```
clear all; close all; clc;
load Testdata
[r, c] = size(Undata);
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);
%% AVERAGE SIGNAL in 3D
Utsum = zeros(n,n,n);
for j=1:r
    Utsum = Utsum + fftn(reshape(Undata(j,:), n, n, n));
end
Utave = fftshift(Utsum)/r;
%isosurface of marble frequency signature after averaging
figure(5)
Utp = abs(Utave)./max(abs(Utave(:)));
isosurface(Kx,Ky,Kz,Utp,0.6)
title('Isosurface of Averaged Transformed Measurements')
xlabel('Wavenumber')
ylabel('Wavenumber')
zlabel('Wavenumber')
axis([-8 8 -8 8 -8 8]), grid on, drawnow
%% Find center frequency
[M, I] = max(abs(Utp(:)));%find maximum amplitude value in average signal matrix
[row,col,vert] = ind2sub(size(Utp),I);
x_freq = Kx(row, col, vert); y_freq = Ky(row, col, vert); z_freq = Kz(row, col, vert);
%% Apply Filter
tau = 0.2;
filter = exp(-tau*((Kx - x_freq).^2 + (Ky - y_freq).^2 + (Kz - z_freq).^2));
xp = zeros(1,r); yp = zeros(1,r); zp = zeros(1,r);
for j=1:r
    Utr(:,:,:)=fftn(reshape(Undata(j,:), n, n, n));
    unft = filter.*fftshift(Utr);
    unus=ifftshift(unft);
    unf=ifftn(unus);
    [M, I] = max(abs(unf(:)));%find maximum value corresponding to marble position
    [row,col,vert] = ind2sub(size(unf), I);
    x_pos = X(row, col, vert); y_pos = Y(row, col, vert); z_pos = Z(row, col, vert);
    xp(j) = x_pos; yp(j) = y_pos; zp(j) = z_pos;
end
plot3(xp, yp, zp), grid on;
title('Marble Trajectory')
xlabel('Spatial Domain x')
ylabel('Spatial Domain y')
zlabel('Spatial Domain z')
pos20 = [xp(20), yp(20), zp(20)]%final marble position
```

Listing 1: MATLAB code from external file used to generate the results presented here.