Coding Project 1: Detecting objects through frequency signatures

Jianhong Chen

January 2023

Abstract

There is Kraken under the ice in Climate Pledge Arena. However, we do not know path and location, but the position of a Kraken was measured by a set of date with noises by spatio temporal seismometers at 1 minute and 15 second intervals. By using Fourier analysis and filtering the data set in MATLAB, we can determine the position and path of Kraken in 3-D space by determining the spatial frequency. Finally, we can make a optimal position for defenceman.

1 Introduction

The traditional electromagnetic radiation experiment that the German scientist Heinrich Hertz performed in the late 1880s, which demonstrated that radio waves are reflected by metallic surfaces, is where radar (or radio detection and ranging) got its start. For example, while driving in really terrible weather, you may establish your check speed and lower it if required to prevent any traffic accidents or speeding penalties when you confirm the existence of active radar signals.

2 Theoretical Background

Since we utilize a noisy data collection, however. We show that the Kraken's course may be traced by extracting its frequency signature using the Fourier transform, using a filter that recovers this frequency signature to 'denoise' the data, then identifying the Kraken over time.

2.1 The Fourier Series

An infinite sum of sines and cosines is used to represent the expansion of a periodic function f(x) into a Fourier series. The orthogonality connections between the sine and cosine functions are used in the Fourier series. The formula for The Fourier Series is

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

For formulating the a_n, b_n, we can multiply a $cos(mx)$ on both side. Then, we can obtain that

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

Then, taking integral for both side. we can obtain that

$$\int_{-\pi}^{\pi} f(x) \cos(mx) dx = \frac{a_0}{2} \int_{-\pi}^{\pi} \cos(mx) dx + \sum_{n=1}^{\infty} (a_n \int_{-\pi}^{\pi} \cos(nx) \cos(mx) \, dx + b_n \int_{-\pi}^{\pi} \sin(nx) \cos(mx) \, dx)$$

By using the orthogonality property of trigonometric functions, we can obtain that

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx \, dx \qquad n \geq 0$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, dx \qquad n > 0$$

Noted that There is other versions of Fourier series for complex numbers (complex Version). The Fourier series on the domain $x \in [-L, L]$, we obtain that

$$f(x) = \sum_{-\infty}^{\infty} c_n \, e^{in\pi x/L} \qquad x \in [-L, L]$$

with the corresponding Fourier coefficients

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) \, e^{-in\pi x/L} \, dx$$

2.2 The Fourier Transform

For main idea of the Fourier Transform is doing integral over $x \in [-\infty, \infty]$,such that

$$F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ikx}\ f(x)\ dx$$

And the inverse Fourier transform is

$$f(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx}\ F(x)\ dx$$

2.3 The Fast Fourier Transform

As far as we know, The Fourier Transform is very useful. However, we face a main question that the operation speed is super slow. Since if we want to do a Fourier transform, we need n^2 operations. This imply that the Fourier transform has a time complexity O(N^2). Therefore, we can introduce 'Fast Fourier Transform(FFT)' for accelerating speed. The operation count for solving a system dropped to O(NlogN) that really speed up the operation.

2.4 Spectral Averaging

Two crucial facts—that the noise is white and that radar will continue to create additional signal data—could be failed by the filtering. To apply averaging, we need to examine the data from these white-noises. White-noise may be represented in two steps: adding a normally distributed random variable with zero means and unit variance to the Fourier components of the spectrum. The key concept is that white noise may eventually be averaged to zero across a large number of signals.

2.5 Spectral Filtering

Because signals and sounds are often intermingled in electronics and signal detection, the filtering process is quite common. This is so that we can see the target signal's reflection, but instead we see reflections from the target signal's intended target as well as reflections from geographical features (such as mountains, forests, etc.) and atmospheric phenomena (clouds, precipitation, etc.). A different signal source that has the appropriate frequency might potentially affect judgment. Therefore, before we examine the signals we receive, they must first be filtered. The radar detecting problem's emitted signal frequency is one example of a particular frequency from which information may be extracted using a technique called spectral filtering. Accurate detection, as well as locating competing signals in time and space, need 'denoising' and precise time-frequency analysis. This method's main goal is to enhance signal detection by 'denoising' data via spectral filtering. In general, using filtering may greatly improve the ability to identify signals in a noisy environment. The Gaussian filter is one of the simplest of the various kinds of filters that may be employed for this purpose. Which is

$$\mathcal{F}\ (k) = \exp(-\tau\ (k \text{ - } k_0)^2)$$

where k is wave-number and τ measures the bandwidth of the filter

3 Numerical Methods

I firstly generate a 64-point 3D axis array, a frequency array, and then rescale them to a 2π-periodic scale. Also, I create 3D grids for the spatial and frequency domains based on the arrays. Also, clearing our workshop is a necessary step.

```
% Clean workspace
 clear all; close all; clc
 load('Kraken.mat'); %load data matrix
 L = 10; % spatial domain
 n = 64; % Fourier modes
 x2 = linspace(-L,L,n+1);
 x = x2(1:n);
 y =x;
 z = x; %create 3D axis arrays with 64 points
 %create frequency array and rescale them to be 2pi periodic
```

```
k = (2*pi/(2*L))*[0:(n/2 - 1), -n/2:-1];
ks = fftshift(k); %shift values to order them correctly
%Create 3D grids for both spatial domain and frequency domain
[X,Y,Z] = meshgrid(x,y,z);
[Kx,Ky,Kz] = meshgrid(ks,ks,ks);
```

The signal is then averaged by adding up all of the realizations in the frequency space and dividing by the total number of realizations(49 realizations here). As a result, we can lessen the impact of white noise. Using the word "realizations" here to emphasize the fact that a new realization of the white-noise is created for each data frame. Also, noted that we use fftn as fast Fourier transform.

```
Unt_sum = zeros(n,n,n);
for j = 1:49
    Un(:, :, :) = reshape(Kraken(:, j), n, n, n);
    Unt_sum = Unt_sum + fftn(Un);
end
A1 = Unt_sum;
% Average the sum over the 49 realizations (i.e., A1/49) and save as A2
A2 = A1/49;
```

The next step is to identify the peak frequency in each of the x, y, and z axes. The element in the signal with the highest weight is the peak frequency of each direction. They may be used to build a suitable filter to extract the real signal.

```
Max = max(A2,[],'all');
LI = find(A2==Max);
ind = [LI];
% converting linear indices to subscripts
[kx0,ky0,kz0] = ind2sub([n,n,n],ind);
A3 = k(kx0);
A4 = k(ky0);
A5 = k(kz0);
```

The peak frequencies in each direction are now known, allowing us to begin building a Gaussian filter to filter the signal. We can pick $\tau = \frac{1}{L}$

```
ki = Kx(kx0,ky0,kz0);
kj = Ky(kx0,ky0,kz0);
kk = Kz(kx0,ky0,kz0);
tau = 1/L;
gFilter = exp(-tau*((Kx - ki).^2 + (Ky - kj).^2 +(Kz - kk).^2));
A6 = gFilter;
```

With the aforementioned result, we can apply a Fast Fourier Transform to determine the filtered signal's peak frequencies. The time of the peak frequencies may then be obtained using the inverse fast Fourier transform. We can finally determine the position of the Kraken by utilizing the time we have.

```
Kraken1 = zeros(3,49);
for j = 1:49
    Un = reshape(Kraken(:, j), n, n, n);
    Untf = fftn(Un).*gFilter;
    unf = ifftn(Untf);
    Max1 = max(unf,[],'all');
    LI1 = find(unf==Max1);
    ind = [LI1];
    [Kraken1_x,Kraken1_y,Kraken1_z] = ind2sub([n,n,n],ind);
    Kraken1(1,j) = X(Kraken1_x,Kraken1_y,Kraken1_z);
    Kraken1(2,j) = Y(Kraken1_x,Kraken1_y,Kraken1_z);
    Kraken1(3,j) = Z(Kraken1_x,Kraken1_y,Kraken1_z);
end
A7 = Kraken1(2,:); % x coordinates
A8 = Kraken1(1,:); % y coordinates
A9 = Kraken1(3,:); % z coordinates
```

4 Results

The position of the Kraken in 3-D space through time is shown in the following graph(figure 1). The Kraken's motions are shown by the blue lines. The x, y, and z values at each point on the blue line relate to where that point is in relation to other points in space. Since we are above the ice, this mean that we

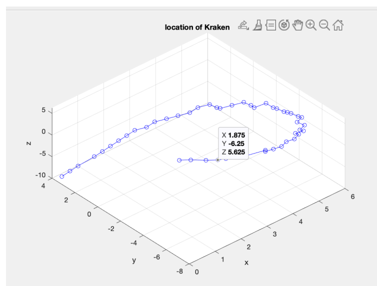

Figure 1: Location of Kraken

need to find the location in 2D. Therefore, The Kraken's 2D plane projection is seen in the following graph(figure 2). Since this graph is a projection from 3D space without the z value, we can observe the Kraken's motions in the 2-D plane The Kraken's x, y, and z coordinates are shown in the table below. Each

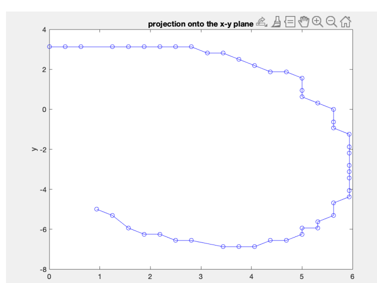

Figure 2: Projection on x-y plane

column represents the Kraken's (x, y, z) position at a certain moment. The table has 49 groupings of sites organized by various periods.

x	0.0000	0.3125	0.6250	1.2500	1.5625	1.8750	2.1875
y	3.1250	3.1250	3.1250	3.1250	3.1250	3.1250	3.1250
z	-8.1250	-7.8125	-7.5000	-7.1875	-6.8750	-6.5625	-6.2500
x	2.5000	2.8125	3.1250	3.4375	3.7500	4.0625	4.3750
y	3.1250	3.1250	2.8125	2.8125	2.5000	2.1875	1.8750
z	-5.9375	-5.6250	-5.3125	-5.0000	-4.6875	-4.3750	-4.0625
x	4.6875	5.0000	5.0000	5.3125	5.3125	5.6250	5.6250
y	1.8750	1.5625	1.2500	0.6250	0.3125	0.0000	-0.6250
z	-3.7500	-3.4375	-3.1250	-2.8125	-2.5000	-2.1875	-1.8750
x	5.9375	5.9375	5.9375	5.9375	5.9375	5.9375	5.9375
y	-0.9375	-1.2500	-1.8750	-2.1875	-2.8125	-3.1250	-3.4375
z	-1.8750	-1.2500	-1.2500	-0.9375	-0.6250	-0.3125	0.0000
x	5.9375	5.9375	5.6250	5.6250	5.3125	5.3125	5.0000
y	-4.0625	-4.3750	-4.6875	-5.3125	-5.6250	-5.9375	-5.9375
z	0.3125	0.6250	0.9375	1.2500	1.5625	1.8750	2.1875
x	5.0000	4.6875	4.3750	4.0625	3.7500	3.4375	3.4375
y	-6.2500	-6.5625	-6.5625	-6.8750	-6.8750	-6.8750	-6.8750
z	2.5000	2.8125	3.1250	3.4375	3.7500	4.0625	4.3750
x	2.8125	2.5000	2.1875	1.8750	1.5625	1.2500	0.9375
y	-6.8750	-6.5625	-6.2500	-6.2500	-5.9375	-5.3125	-5.0000
z	4.6875	5.0000	5.0000	5.6250	5.6250	5.9375	6.5625

Table 1: (x, y, z) position

5 Conclusion

I demonstrated the origins and uses of many methods in this study, including the Fourier series, Fourier transform, rapid Fourier transform, and others. Then I demonstrated how to analyze signals with noise using the aforementioned methods. In order to locate the genuine Kraken signal, we must thus average and filter the signals. The project's findings are shown visually in the graphs and table. I gained knowledge about the Fourier transform's value in signal processing by working on this project. Both the time domain and the frequency domain may be used to investigate a signal inside the Fourier transform and inverse Fourier transform. We constantly learn more about the emitting objects using this strategy. I also know how to denoise signals using averaging and filtering to achieve a clear analysis.

Acknowledgment

Thanks for our professor Amin Rahman providing a chance to learn this interesting project. Also, Thanks for our TA's helps.