Coding Project 1: Detecting objects through frequency signatures

Hao Li
Department of Applied Mathematics
University of Washington
haoli25@uw.edu

Abstract

In the project, we are going to detecting objects through frequency signatures. In other words, we want to find the specific location of the object through out each time-step. More specific, using a spatiotemporal seismometer, we can collect the signals from the object at 1 minute and 15 second intervals. Thus, we are going to work these data in its frequency domain by applying the fast Fourier transform algorithm. After averaging the frequency, we will be able to find where it hits the maximum frequency so that we can apply the appropriate filter to reduce the noise and enhance our signal. By doing so, we can locate the object in the spatial domain more accurately. The results will be visualized by the figures in the later sections.

1 Introduction

For this project, we encountered a Kraken under the ice in Climate Pledge Arena. Players cannot feel the vibrations caused by the Kraken but a spatiotemporal seismomete does. Thus, the player can land a effective hard check if the players know the Kraken's position. Find the path of the Kraken thought out the time will be important to win the game.

2 Theoretical Background

In this section, we are going to discuss the theoretical background for solving the problem such as, Fourier Transform, Spectral Averaging, and Spectral Filtering.

2.1 The Fourier Series

Similar as the Taylor series, Fourier series are infinite series that represent periodic functions in terms of cosines and sines which form the basis of this expansion [1]. The original Fourier transform was invented by Fourier in the early 1800s to study the heat equation [3].

$$\frac{\partial u}{\partial t}(x,t) = c^2 \cdot \frac{\partial^2 u}{\partial x^2}(x,t)$$

where c^2 is called thermal diffusivity and we have the boundary condition: $u(0,t) = u(L,t) = 0$. Fourier found the solution is

$$u(x,t) = \sum a_n e^{c^2 n^2 \pi^2 t / L^2} \cdot \sin \frac{n\pi x}{L}$$

if $f(x) = u(x,0) = \sum a_n \sin n\pi x/L$. Note that a_n is the coefficient constant where $n \in \mathbb{N}$. Moreover, the full Fourier Series on the domain $x \in [-L, L]$ can be written as

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \tag{1}$$

$$= \sum_{-\infty}^{\infty} c_n e^{in\pi x/L} \tag{2}$$

by Euler's formula. More importantly, these $\sin$ and $\cos$ term form an orthonormal basis and if we take the inner product of $f(x)$ with the basis, it will bring the coefficients out because of the orthogonality.

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x)\, e^{-in\pi x/L}\, \mathrm{d}x \tag{3}$$

2.2 The Fourier Transform

As an extension of Fourier Series, the Fourier Transform is an integral transform defined over the whole real line. The Fourier Transform and its inverse on $x \in [-\infty, \infty]$ can be defined as the following expression.

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-ikx} f(x)\, \mathrm{d}x$$

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(k) e^{ikx}\, \mathrm{d}k$$

Note that if we consider k as the frequency and s as the spacial location, Fourier Transform then reveals that we can decompose a function of space into its constituent frequencies and this is key idea of solving this problem. By applying the Fourier Transform, we will be able to find the maximum frequency so that we can apply the appropriate filter to reduce the noise and enhance our signal. Then we can turn the filtered signal back to the spacial domain by apply the inverse Fourier Transform and that will allow us to find the accurate Kraken's location.

2.3 The Fast Fourier Transform

The general definition of Fourier Transform is an integral transform. However, a digital computer cannot work with a continuous-time signal and we need to take some discrete samples of $f(x)$. Thus, instead of integral, we can do a finite sum on a finite domain $x \in [-L, L]$. Unfortunately, such discretization does not really solve the real-world problems. Given an N points discretization, Discrete Fourier Transform has the computational complexity of $\mathcal{O}(N^2)$ for just one time-step. Hence, it is crucial to use the Fast Fourier Transform (FFT) which is introduced by Cooley and Tukey in the mid 1960s. The most important improvement is that FFT has the computational complexity of $\mathcal{O}(N \log N)$ [2]. By doing so, digital computers can quickly computer the Fourier Transform and its inverse with just some simple commands (e.g. `numpy.fft, numpy.ifft`) in Python.

2.4 Spectral Averaging

We have seen in the lecture that the white-noise can be modeled by adding the standard normal distribution to each Fourier mode. Hence, given a sequence of data in time, adding them up and take the average should wipe out the white-noise and extract a clean spectral signature [2].

2.5 Spectral Filtering

Spectral filtering is a method which allows us to extract information at specific frequencies and remove unwanted noises or frequencies from our signal. More specifically, here we are going to apply the Gaussian filter. Because of the decay property of normal distribution, any high-frequency noise that is far from the center will be quickly wiped out. The Gaussian filter's standard deviation regulates the width of the kernel and consequently the degree of smoothing applied to the signal. The mean controls the center of the filter. Hence, it is important to find the location of our peak frequencies before applying the filter otherwise, the filter will reduce our desired signal.

3 Numerical Methods

The basic algorithm can be implemented as follows:

i Loading the data and setting up the spatial and frequency domain where $x, y, z \in [-10, 10)$ with 64 equally spaced points.

ii For each time-step, reshape the signal into a $(64, 64, 64)$ 3D tensor and thus we have signals in x, y, z direction. Using `numpy.fft.fftn`, we can compute the 3-dimensional discrete Fourier Transform at each each time-step.

iii Summing them up and take the average, our signal should have less white-noise. Then using `numpy.argmax`, we will be able to find where is maximum frequency is each direction in the frequency domain.

iv Using `numpy.meshgrid`, we can create a 3D Gaussian filter centered at peak frequencies in each axis and the width of the filter is controlled the parameter τ. Later, we found when $\tau = 1/L$ gives us the best result.

v Now for each time-step, we apply our filter to our signal (in frequency domain) and using `numpy.fft.ifftn`, we can take the frequencies back to the spacial domain. Then we can find the maximum signal in each axis using the same function as we did in frequency domain.

vi Plot the location of Kraken for each time-step and that will generate the path of the Kraken.

4 Results

To have better visualization, we use `matplotlib.image.read` to read the image of Climate Pledge Arena and plot it with the Kraken's path. We assume that the ice surface is at the level $z = 10$ and Kraken is moving under the ice surface.

4.1 Kraken's Path in x-y plane

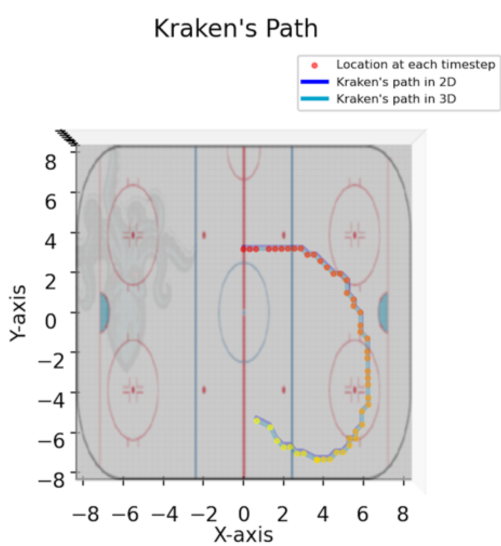


Figure 1: Kraken's path in 2D with the ice surface as the background

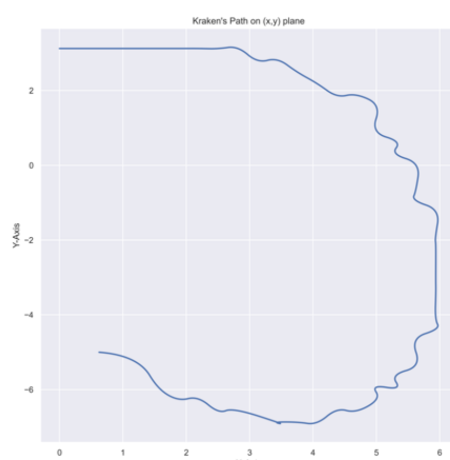
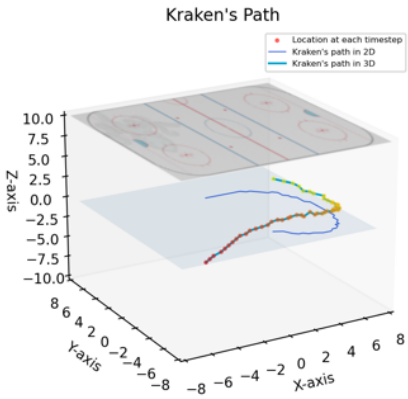
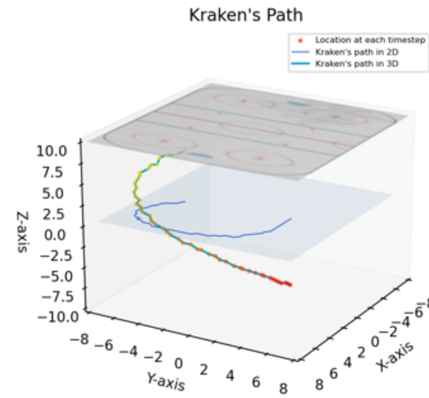


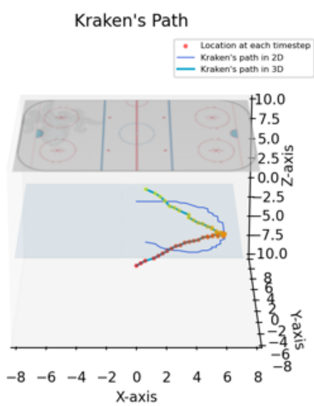
Figure 2: After zoom in the Kraken's path in 2D with more clean background

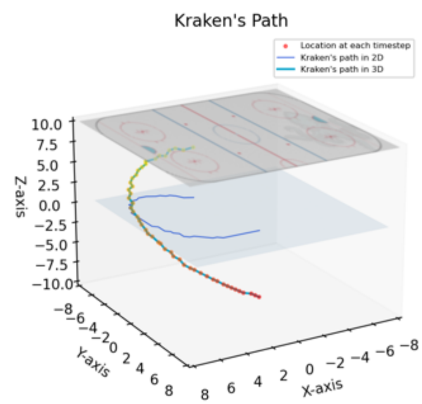
From the plot, we see Kraken is moving from top to bottom (orange→ yellow) like flipped 'C' shape. From Figure 1, we see most of the time, Kraken is in the 'attacking zone' (except the first 7 and last 6 time-steps). Figure 2 gives us a more clear visualization of Kraken's path on (x, y) plane. Hence, those locations will be optimal for a defenceman to be to make a hard check.

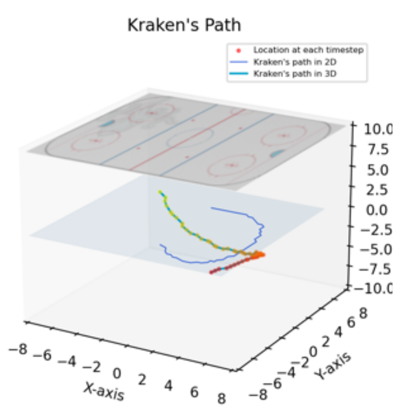
4.2 Kraken's Path in 3D

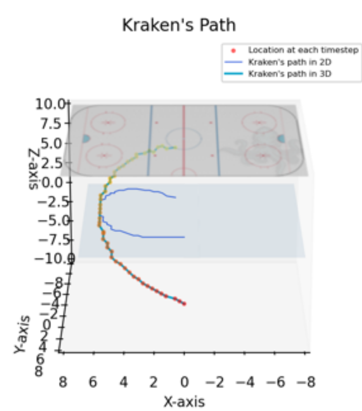












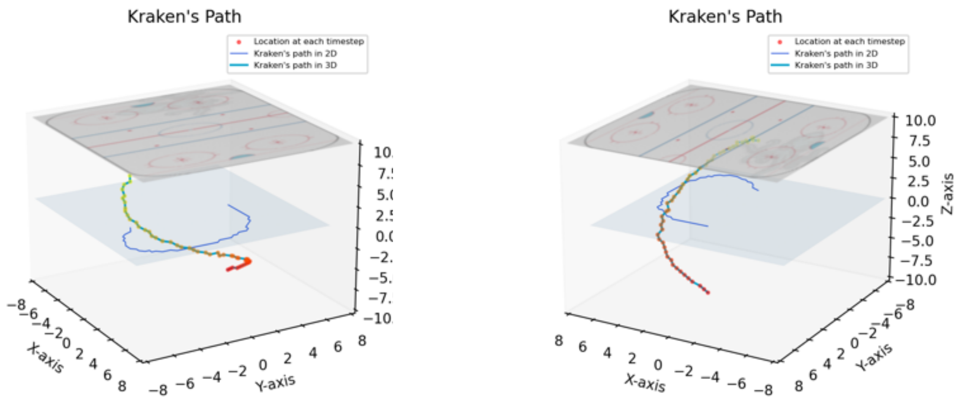


Figure 3: Kraken's path in 3D. Blue surface indicates the level $z = 0$. The left column figures are viewed at the angle from $(20, -120)$ to $(20, -30)$ and the right column figures are viewed at the angle from $(20, 30)$ to $(20, 120)$.

From the graphs, we can see Kraken's path is like parabola curve, moving from $z = -7.5$ to $z = 7.5$. Also we can see Kraken is not moving at a constant speed. Points are not all equally spaced.

5 Conclusion

From this project, we practiced some basic signal processing skills. Specifically, we are using spectral averaging and spectral filtering techniques to reduce the noises and enhance the signal of our data. The key idea behind these techniques is to convert the data from the spacial domain to the frequency domain so that we can locate these peak frequencies and apply our filter. However, Gaussian filter is not the only choice. In the future work, we might consider different filters and see what information they can extract from the data.

6 Acknowledgment

Thanks for the help from Katie's and Professor Amin's office hour.

References

[1] E. Kreyszig. *Advanced Engineering Mathematics, 10th Edition*. Wiley Global Education, 2010.

[2] J.N. Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Data-driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data. OUP Oxford, 2013.

[3] Benjamin Steinberg. *Representation Theory of Finite Groups*. Springer New York, 2012.