Coding Project 1: Detecting objects through frequency signatures

Tyler Shakibai

Abstract

There is a monstrous Kraken lurking underneath the ice at Climate Pledge Arena. The Kraken plays a role in the hockey match by determining when and where players are allowed to land a hard check on an opposing player. In this project, we attempt to denoise the vibration measurements that correspond to the movement of the Kraken using methods that originate from radar detection. We utilize the Fourier transform, specifically the fast Fourier transform (FFT), to filter and analyze the noisy signals that vibrate through the ice and compose our measurements. After filtering and denoising, we are able to estimate the path of the Kraken to determine when a hard check is permitted.

1 Introduction

Radar detection dates back to the late 1800s with the experiments of German physicist Heinrich Hertz who showed that metallic objects could reflect radio waves. Radar detection later became a research area of great interest to many nations given its applications for communication, especially during the various armed conflicts of the 20th century. The first significant use of radar detection was for military purposes of locating targets in the sea or air. To date, radar is prevalent in many aspects of society ranging from communication, space exploration, air traffic control, ship navigation, and more.

The Fourier transformation was developed in 1822 by the French mathematician Jean Baptiste Joseph Fourier in his book *Théorie analytique de la chaleur*. The Fourier transform has become essential to radar detection due to its ability to process complex radio information as sinusoidal waveforms. The field of signal processing was revolutionized in 1965 when James Cooley and John Tukey developed the fast Fourier transform (FFT) algorithm, which is able to compute the discrete Fourier transform (DFT) more efficiently than previous implementations. The FFT is considered to be the greatest innovation in this domain and one of the most significant and influential algorithms in recent history.

This project will demonstrate how these developments can be utilized to process and denoise complex signal measurements. Specifically, we are interested in how these methods can be applied to the task of locating the Kraken underneath Climate Pledge Arena. We will discuss the mathematical background behind the Fourier series and the aforementioned Fourier transform. Subsequently, we will go into detail regarding the numerical scheme we will use to filter and process our data so that we can isolate the frequencies that correspond to the Kraken's location.

2 Theoretical Background

In this section, we will delve into the theoretical underpinnings of signal processing. This discussion will be focused primarily on derivations from Fourier Analysis as they relate to processing information into the frequency domain.

2.1 The Fourier Series

The main concept we will explore and examine in this work is the Fourier transform. The Fourier transform is an extension of the Fourier series which expands periodic functions into a sum of sinusoidal basis functions, sines and cosines, as is defined as

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

To derive this formulation, we assume there exists a periodic of function of interest $f(x)$ such that

$$f(x+P) = f(x) \qquad P > 0$$

which implies

$$f(x) = f(x+P) = f(x+2P) = f(x+3P)$$

We define P as the period of the function $f(x)$. Trigonometric functions such as $\sin x, \cos x, \tan x$ are some simple examples of periodic functions. We can then express $f(x)$ as an infinite sum of sine and cosine functions as follows

$$\begin{aligned} f(x) = \frac{a_0}{2} &+ a_1 \cos x + a_2 \cos 2x + a_3 \cos 3x + \ldots + a_n \cos nx \\ & b_1 \sin x + b_2 \sin 2x + b_3 \sin 3x + \ldots + b_n \sin nx \end{aligned} \tag{2}$$

Using summation notation, we obtain the previous series representation (1)

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

Using the orthogonality of $\sin x$ and $\cos x$ functions, we can find the coefficients of the Fourier series as

$$a_0 = \frac{1}{2L} \int_{-P}^{P} f(x) dx \quad a_n = \frac{1}{L} \int_{-P}^{P} f(x) \cos \frac{n\pi x}{P} dx \quad b_n = \frac{1}{L} \int_{-P}^{P} f(x) \sin \frac{n\pi x}{P} dx$$

The Fourier series and its related coefficients will serve as the foundation for the signal processing and analysis we will perform.

2.2 The Fourier Transform

The Fourier transform is an extension of the Fourier series which allows for a signal to be converted from its original domain, such as space or time, into the frequency domain. We can analytically derive the Fourier transform from the series as follows. First recall Euler's formula, which is $e^{ix} = \cos x + i \sin x$ We can rewrite the Fourier series more compactly as

$$f(x) = \sum_{n=-\infty}^{\infty} C_n e^{\frac{in\pi x}{L}} \tag{3}$$

where the Fourier coefficients in frequency space defined as

$$C_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{\frac{-in\pi x}{L}} dx \tag{4}$$

Previously, we were bounded by L in our capacity to fit periodic signals. Taking the limit

$$\lim_{L\to\infty} C_n = 0$$

the domain is no longer restricted, meaning any number of sinusoids can be summed to fit our frequencies, making them homeomorphic to the set of real numbers. In other words, we no longer have a vector of coefficients, but rather a continuous function. Applying the limit to (4) we get

$$F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-ikx}dx \tag{5}$$

To understand how this function transforms the input, we will first look at the integrand. e^{-ikx} rotates $f(x)$ clockwise in a unit circle on the complex plane with frequency k. Integrating this term over x gives the center of mass of the function $f(x)e^{-ikx}$. The value of this center of mass remains small in magnitude for all frequencies unless it is equal to the frequency of $f(x)$, in which case this value moves away from zero and the frequency signature of the function can be detected.

Given this interpretation, it is necessary to have an operation that performs the opposite transformation, converting a signal from the frequency domain to some regular time or space domain. Thus, we define the inverse Fourier transform as

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(k)e^{ikx}dk \tag{6}$$

Together, these two integral transforms allow us to manipulate and analyze complex signals by moving them in and out of frequency space.

2.3 The Fast Fourier Transform

The fast Fourier transform is an algorithm that computes the discrete Fourier transform (DFT) of a sequence of information. It can alternatively be used to find the inverse discrete Fourier transform (IDFT). FFT converts a signal from its original space or time domain to a representation in the frequency domain, with the inverse Fourier transform performing the opposite conversion. The process of decomposing a signal into components of different frequencies is applicable in many fields but can be computationally intractable directly. Instead, the FFT algorithm utilizes sparsity to compute the transformation by factorizing the DFT matrix. This implementation reduces the complexity of computing the DFT from $O(N^2)$ to $O(N \log N)$ for data with size N. While there are many Python libraries that can perform FFT, we will SciPy's fast Fourier Transforms library, which can be called using

```
import scipy.fft
scipy.fft.fftn(f(x))
```

for some function f that we would like to compute the n-dimensional discrete Fourier transform of over some space x.

2.4 Spectral Averaging

We employ spectral averaging to help mitigate the effect of white noise on our signal measurements. In many real-world applications, radio detectors are not receiving just one signal, but rather many signals from varying locations in space and time. This measurement process is highly prone to white noise. Naturally, we are interested in ways to make our signal processing methods robust to such imprecision. We can take advantage of the fact that we are receiving multiple signals and simply take the average. This is useful because the noise is expected to be random, meaning sometimes the realizations, or observations, of our signal are higher or lower than in actuality. However, since these deviations are random in most cases, they will balance out over the entirety of our signal. Thus, averaging our signal damps the white noise and the actual signal with remain through the averaging, effectively amplifying the true signal we want to analyze.

2.5 Spectral Filtering

Spectral Filtering is the process of eliminating information that is not of interest in order to amplify what remains. This elimination is performed by observing the wavelengths in frequency space by taking the Fourier transform, and setting wave values to zero if they do not possess the correct frequency signature. For our application, this entails filtering signals that are determined to be noise as opposed to the vibrations that correspond to the Kraken's movements. Spectral filtering attenuates higher frequencies, which correspond to the source of the signal, as opposed to noise that may be present.

The simplest filtering approach is simply averaging the signal's frequencies to reduce the effect of noise. While this is useful, there are more effective filters that to process complex signal data. Another common filtering method to reduce noise is the Gaussian filter, which can be implemented in varying dimensions. The Gaussian filter isolates the frequency center from data to find the component of the signal that corresponds to the object of interest. The filtered frequency spectrum can then be inverse Fourier transformed to indicate the spatial or temporal component of the object.

3 Numerical Methods

Now we will discuss an overview of the algorithm and the numerical methods employed to perform the signal processing and filtering. Firstly we load out data from the original source file and define our parameters based on the given domain and establish our spatial and spectral resolution. We will be using the interval $[-L, L]$ for our spatial domain, and 64 points for our resolution, or Fourier modes.

Next, we need to discretize our spatial domain using the parameters we just defined as a three-dimensional linear space in which the data will exist before it is transformed into the Fourier domain. Similarly, we discretize the frequency domain k using our Fourier modes on a 2π periodic space, and normalize for the interval $[-L, L]$. Together, these steps provide our spatial and spectral resolutions. We also want to shift the zero-frequency component to the center of the spectrum, using a ubiquitous function known as FFT shift. As a final step of our preprocessing, we then need to create two mesh grids, one for each of the discretized domains.

Now that our setup is complete we can begin with our iterative methods. For each realization in our data, the signal is reshaped into a 64 x 64 x 64 multidimensional array, or cube. Afterward, we want to sum our realizations in frequency space. To do this, we simply take the aforementioned FFT of our cubes of information, then iteratively take the sum over our dataset. We then want to average over all of our realizations which will be useful for damping white noise.

Our next goal is to isolate the peak frequencies in x-y-z space, that is, find the max in each direction of the normalized sum of frequencies in our original space. This information will provide us with the optimal center frequency for our filter. To find the center, we will need to unravel the index where such peak frequencies exist. We will then use the center frequency of the Kraken to position a Gaussian filter in the frequency domain to isolate the signature of the monster for each realization. We can then take the inverse Fourier transform of this filtered frequency spectrum to reveal the location of the Kraken in 3D space. We define our three-dimensional Gaussian filter as

$$F(k) = e^{-\tau(k_x - x_0)^2 + (k_y - y_0)^2 + (k_z - z_0)^2}$$

where $\tau = \frac{1}{L}$ determines the width of the filter and x_0, y_0, z_0 determine its center.

Lastly, to locate the Kraken, we use the isolated peak frequencies of the filtered signal to estimate the monster's location in the original domain. To do this, we iterate through the data again, and at each iteration, we reshape our signal cube and take the FFT, then we apply our appropriately centered filter to the reshaped signal cube in frequency space, then use the inverse fast Fourier transform to move the signal back into x-y-z space. Once again, we find the modulus in each direction, corresponding to the peak frequencies. Since these frequencies are correlated to the Kraken's movement, the indices where these peaks occur correspond to the Kraken's location. Once we unravel the accompanying index for each peak, we can plug them into the mesh grid for the original space discretization to return an array of the monster's position in three-dimensional space through time. Now we are ready to plot our results.

4 Results

The first and most revealing plot is that of the Kraken's path in three dimensions and through time underneath Climate Pledge Arena.

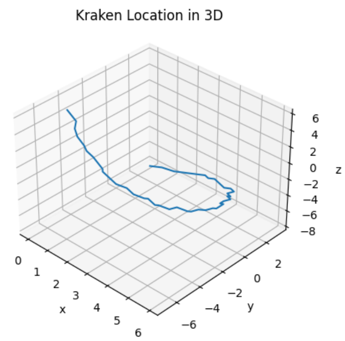

Figure 1: Three-dimensional plot of the Kraken's estimated movement through time.

Using the Gaussian filter, we are able to plot a smooth curve representing the monster's trajectory, which would otherwise be a jagged and noisy curve. Our signal processing reveals a winding path below the ice rink. We can project this path to a lower-dimensional space in order to only deal with the information that relates to the hockey game.

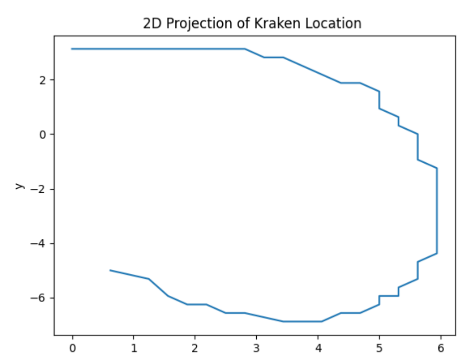

Figure 2: Two-dimensional projection of the Kraken's path onto the x-y plane through time.

This projection gives the projection seen in figure 2. This curve gives the x-y locations where a hard check can be landed by hockey players throughout the game.

	x	y	z		x	y	z
1	0.0000	3.1250	-8.1250	26	5.9375	-2.8125	-0.6250
2	0.3125	3.1250	-7.8125	27	5.9375	-3.1250	-0.3125
3	0.6250	3.1250	-7.5000	28	5.9375	-3.4375	0.0000
4	1.2500	3.1250	-7.1875	29	5.9375	-4.0625	0.3125
5	1.5625	3.1250	-6.8750	30	5.9375	-4.3750	0.6250
6	1.8750	3.1250	-6.5625	31	5.6250	-4.6875	0.9375
7	2.1875	3.1250	-6.2500	32	5.6250	-5.3125	1.2500
8	2.5000	3.1250	-5.9375	33	5.3125	-5.6250	1.5625
9	2.8125	3.1250	-5.6250	34	5.3125	-5.9375	1.8750
10	3.1250	2.8125	-5.3125	35	5.0000	-5.9375	2.1875
11	3.4375	2.8125	-5.0000	36	5.0000	-6.2500	2.5000
12	3.7500	2.5000	-4.6875	37	4.6875	-6.5625	2.8125
13	4.0625	2.1875	-4.3750	38	4.3750	-6.5625	3.1250
14	4.3750	1.8750	-4.0625	39	4.0625	-6.8750	3.4375
15	4.6875	1.8750	-3.7500	40	3.7500	-6.8750	3.7500
16	5.0000	1.5625	-3.4375	41	3.4375	-6.8750	4.0625
17	5.0000	0.9375	-3.1250	42	3.4375	-6.8750	4.3750
18	5.3125	0.6250	-2.8125	43	2.8125	-6.5625	4.6875
19	5.3125	0.3125	-2.5000	44	2.5000	-6.5625	5.0000
20	5.6250	0.0000	-2.1875	45	2.1875	-6.2500	5.0000
21	5.6250	-0.6250	-1.8750	46	1.8750	-6.2500	5.6250
22	5.6250	-0.9375	-1.8750	47	1.5625	-5.9375	5.9375
23	5.9375	-1.2500	-1.2500	48	1.2500	-5.3125	5.9375
24	5.9375	-1.8750	-1.2500	49	0.6250	-5.0000	6.2500
25	5.9375	-2.1875	-0.9375				

Figure 3: Table of the Kraken's x-y-z locations for $t \in [0, 49]$

The table of the Kraken's location in three-dimensional space through time shows that the monster is moving in the positive z-direction. This information, along with the previous plots, informs that the Kraken is spiraling upwards underneath the ice rink as time progresses. We can also infer that the monster is moving clockwise around the path in the two-dimensional projection.

5 Conclusion

Using signal processing methods derived from Fourier Analysis, we were able to denoise the signals we measured that vibrated through the ice at Climate Pledge Arena, in Seattle, Washington. By moving our data into the Fourier domain, or frequency space, we were able to isolate the frequencies that corresponded to the Kraken moving underneath the ice rink. Then a Gaussian filter was applied the provide clarity to an otherwise noisy frequency signature for each realization. Using the inverse transform, the data was brought back into three-dimensional space so that we could observe the Kraken's path through time. The permissible locations on the ice for a player to perform a hard check follow the two-dimensional projection of the Kraken's movement onto the x-y plane.

While performing this object detection task, I learned how to manipulate data using the Fourier transform. I also learned how to appropriately center a filter in the Fourier domain to denoise frequency data. There do exist drawbacks to the method applied in this project, including the assumptions we made while processing the data. Namely, we assumed that there existed only one characteristic frequency or frequency signature of the Kraken, which might not be the case in other systems. In more complicated systems, the frequency signature might be dynamic and change through space and time.

Acknowledgment

I attended office hours with the course's TA, Katie, and heavily utilized the Numpy and Scipy Documentation. I also used Stack Overflow to make a table in Python using pandas.