Coding Project 1: Detecting objects through frequency signatures

Justin Hexem

Abstract

In this project, we want to be able to find the location of a Kraken underneath the ice in the Climate Pledge Arena. To do this, we use the data obtained by a temporal seismometer to predict the path that the Kraken travels so we can then relay this information to the members of the home team in order to gain a competitive advantage. We will find the frequency that the signals are centered about by averaging the spectrum, filter the data at each time step, and finding the maximum values of the data to trace out the path of the Kraken. We will then project this three-dimensional path into the xy-plane to determine where on the ice the defencemen should be placed at any given time.

1 Introduction

The idea of radar detection dates back to the late 19th century when a German physicist named Heinrich Hertz was conducting some experiments. He discovered that radio waves were able to be bounced off of metal objects. This discovery led to a German inventor named Christian Hülsmeyer who designed a device in the early 20th century that could detect other ships when visibility was low in order to prevent collisions. The first modern radar systems was developed by eight nations independently in the years 1934-1939, and these nations used this technology extensively during World War II. Although radar technology did not contribute much to any new theoretical insights, it was a tremendous leap forward in the fields of engineering and technology. This technology was of great use to the allied powers of World War II and was one of the key factors that led to their victory in the war. Some even say that the invention of radar contributed to the allied victory

more than the atomic bomb did. After the war, radar helped astronomers map the surfaces of planets that are extremely far away, it helped doctors be able to directly view internal organs, and allowed air travel to be carried out safely.
Throughout this report, we will analyze how this technology works as well as how to apply it to tracking the movement of a Kraken underneath the Climate Pledge Arena. We will introduce the concept of the Fourier Series, the Fourier Transform, the Fast Fourier Transform algorithm, the technique of spectral averaging, and the technique of spectral filtering. We will then take a look at some code that uses these concepts to find the path of the Kraken from the data obtained from the temporal seismometer.

2 Theoretical Background

We will discuss the mathematics behind how radar detection works by introducing a few key topics. The following concepts will help us understand the code that calculates the path of the Kraken, and how the data is obtained by the temporal seismometer.

2.1 The Fourier Series

A mathematician named Joseph Fourier first came up with the idea that any function could be represented by a trigonometric series involving nothing but sines and cosines. For some function $f(x)$ this can be written as:

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

Fourier claimed that this was true even for discontinuous functions, which was a startling assertion. Let us assume that Eq. (1) is a uniformly convergent series. By multiplying both sides by $\cos mx$, integrating both sides, and simplifying, we obtain the following formulas for the coefficients a_n and b_n:

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx dx \quad n \geq 0 \tag{2}$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx dx \quad n > 0 \tag{3}$$

These coefficients are called the Fourier coefficients. We can also write the complex version of this expansion on the domain $x \in [-L, L]$:

$$f(x) = \sum_{-\infty}^{\infty} c_n e^{in\pi x/L} \quad x \in [-L, L] \tag{4}$$

with the Fourier coefficients expressed as

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x/L} dx. \tag{5}$$

Despite the fact that the Fourier series is complex, we observe that our function $f(x)$ is still a real-valued function. This Fourier series shows us how to express any real-valued function in terms of only sines and cosines.

2.2 The Fourier Transform

We will now see what happens when we take the limit as L approaches infinity. We see that with a finite L, that we can only have a finite number of Fourier coefficients c_n. However, once we take the limit as L goes to infinity, we see that c_n becomes a function instead of taking on discrete values. This function is called the Fourier transform of $f(x)$, denoted $F(k)$, and it can be written as:

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

If we examine what happens when we take the limit as L approaches infinity of Eq. (4), we see that we obtain an equation for our original function $f(x)$. This is called the inverse Fourier transform of $F(k)$, and it can be written as:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} F(k) dk. \tag{7}$$

We see that our Fourier transform converts our function $f(x)$ that is a function of x into a new function $F(k)$. One unique property of this transform is that taking derivatives becomes equivalent to multiplying by the constant term ik. This allows the Fourier transform to be extremely useful in solving differential equations that would be impossible to solve by other means.

2.3 The Fast Fourier Transform

Being able to calculate the Fourier transform allows us to solve extremely difficult problems, but the formula for calculating it is difficult to execute due to the fact that an infinite integral is involved. This is why finding an algorithm that can quickly calculate the Fourier transform is important. Such an algorithm exists and is called the Fast Fourier Transform, and it allows us to solve a wide variety of problems extremely quickly, such as signal processing, image processing, and solving differential equations that model the world we live in. In MATLAB this function can be called by using the built-in function fft, and it can be used to compute the Fourier transform along any number of dimensions.

2.4 Spectral Averaging

If we have a given set of data that is extremely noisy, it can sometimes be difficult to detect what the underlying signal looks like. There are several ways to go about solving this issue, but one method in particular is called spectral averaging. This method makes use of the fact that the Fourier transform takes our original signal from the temporal domain into the frequency domain, which means that the pattern that we are trying to detect does not vary in time. The frequency of the signal we are looking for will remain constant. We take advantage of this by taking several different measurements in time of the same signal, and because they have the same frequency, we may take the fft of each signal and average all of our samples together. This reinforces the signal we are searching for, and averaging the noise will cause it to dissipate due to its random nature. This allows us to be able to see the signal we are looking for much more clearly and it removes a large amount of the noise from our data.

2.5 Spectral Filtering

When we find the location of the frequencies of our signal, we need some way to isolate those frequencies at each time step to determine the location of our signal. To do this, we must filter out all other signals that occur due to the random noise, and we must amplify the frequencies that we are looking for. The process in which this is done involves taking the Fourier transform of our function, multiplying a filter function to our frequency function to isolate the desired frequencies, and then applying the inverse Fourier transform to obtain data that has much less noise. One common function that can be used as a filter is the Gaussian because it is close to zero everywhere except for the center of its distribution. By translating this Gaussian function to target certain frequencies, we can drastically reduce the amount of random noise in our signal.

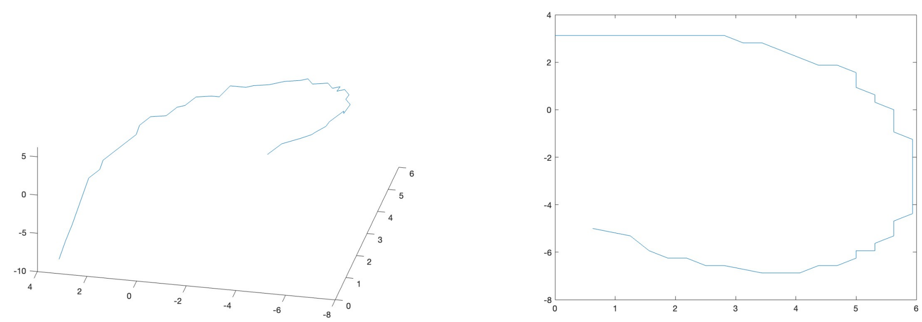
3 Numerical Methods

In order to calculate the path of the Kraken, we must first load the data obtained from our temporal seismometer into MATLAB. This data is extremely noisy, so the frequencies that we want are obscured by the noise. We need to use spectral averaging to average out the noise so that we can see the frequencies that we are targeting more clearly. To do this, we will first reshape our data from 49 large columns into 49 64 by 64 by 64 tensors. Then we will perform the fft algorithm along each of the three dimensions to obtain 49 frequency tensors. These 49 tensors are still extremely noisy, but we know that the frequency of the signal that we wish to find will be in the same location for each of the 49 tensors, so we will take the average of all of the tensors to reduce the noise enough to determine the center of our desired frequencies by taking the maximum value of this averaged tensor.

After finding the center of the frequencies for our signal, we know which parts of our signal to isolate and which parts appear due to the noise. Now we must use spectral filtering to isolate the center of the frequencies at each time step. This will remove most of the noise from our original signal, and will allow us to obtain a location in space for the Kraken at each of our time steps. To do this, we will use a Gaussian filter function centered about the maximum frequencies of our averaged tensor. We will multiply this Gaussian filter to the Fourier transform of each of our 49 time steps in order to isolate the frequency of the Kraken.

After this process, we will Fourier transform our data back into the time domain and our data will have most of the noise removed. Calculating the maximum value of our signal at each of the time steps will give the location of the Kraken at that time step. We can then plot the path of the Kraken by connecting this set of 49 coordinate points. To find out where on the ice a defenceman should be, we now need to project this three dimensional path into the two dimensional surface of the ice.

x	y	z	x	y	z	x	y	z	x	y	z	x	y	z
0	3.125	-8.125	3.4375	2.8125	-5	5.625	-0.625	-1.875	5.625	-4.6875	0.9375	3.4375	-6.875	4.0625
0.3125	3.125	-7.8125	3.75	2.5	-4.6875	5.625	-0.9375	-1.875	5.625	-5.3125	1.25	3.4375	-6.875	4.375
0.625	3.125	-7.5	4.0625	2.1875	-4.375	5.9375	-1.25	-1.25	5.3125	-5.625	1.5625	2.8125	-6.5625	4.6875
1.25	3.125	-7.1875	4.375	1.875	-4.0625	5.9375	-1.875	-1.25	5.3125	-5.9375	1.875	2.5	-6.5625	5
1.5625	3.125	-6.875	4.6875	1.875	-3.75	5.9375	-2.1875	-0.9375	5	-5.9375	2.1875	2.1875	-6.25	5
1.875	3.125	-6.5625	5	1.5625	-3.4375	5.9375	-2.8125	-0.625	5	-6.25	2.5	1.875	-6.25	5.625
2.1875	3.125	-6.25	5	0.9375	-3.125	5.9375	-3.125	-0.3125	4.6875	-6.5625	2.8125	1.5625	-5.9375	5.9375
2.5	3.125	-5.9375	5.3125	0.625	-2.8125	5.9375	-3.4375	0	4.375	-6.5625	3.125	1.25	-5.3125	5.9375
2.8125	3.125	-5.625	5.3125	0.3125	-2.5	5.9375	-4.0625	0.3125	4.0625	-6.875	3.4375	0.625	-5	6.25
3.125	2.8125	-5.3125	5.625	0	-2.1875	5.9375	-4.375	0.625	3.75	-6.875	3.75			



(a) Path of the Kraken

(b) Path of Kraken on Ice

Figure 1: Path of the Kraken from $t = 1$ to $t = 49$

4 Results

Figure 1 shows the plots of the Kraken's path in both three dimensions (1a) and this path projected into the xy plane (1b). The first plot was obtained by finding the maximum value of our signal at each time point and connecting them with a line. We see that this path almost resembles a helix in three dimensions. The second plot was generated by projecting the previous three dimensional curve onto the xy plane. This was done by only plotting the x and y coordinates, and this path gives the locations that the defenceman should be placed at that time during the game. The table above shows the coordinates of the Kraken at each time point.

5 Conclusion

By applying spectral averaging and spectral filtering, we were able to remove the noise from the data in order to view the Kraken's path underneath the arena. This project taught me many new things about how the mathematics behind radar technology works, and it showed me how useful the fft algorithm can be in helping solve these types of problems. It was interesting to see the difference in the amount of noise before applying the filter and afterwards. The threshold did not help much at all initially, but we barely needed the threshold at all after the data was filtered. This technique would encounter some problems if the frequency of our signal changes over time due to the fact that averaging the signal would cancel out parts of the signal we are searching for. Another possible drawback with this method would be detecting multiple objects at once due to the fact that our filter function would need to be more complicated.

Acknowledgment

I appreciated all the help that Professor Rahman and Michael Kupperman provided while I worked through this project. Thank you for taking the time to help me understand certain details of the problem that I had initially overlooked.