AMATH 482/582 HOMEWORK 1 - FINDING SUBMARINES

PHILIP WESTPHAL

Department of Applied Mathematics, University of Washington, Seattle, WA
philw16@uw.edu

Abstract. In this report, the use of the Fast Fourier Transform (FFT) algorithm to convert signal data to the frequency domain and remove noise to find a central frequency signature is explored. This methodology was applied to a three dimensional acoustic pressure signal given off by a submarine at specific time intervals. The obtained central frequency was used with a Gaussian filter to attenuate other signatures in the acoustic data and ultimately identify the physical location of the submarine in the problem space.

1. Introduction and Overview

The goal of this problem was to locate a submarine in the Puget Sound such that a tracking aricraft could be deployed to monitor the sub. A two dimensional dataset is provided, with each column representing a flattened $64x64x64$ grid of acoustic pressure data given off by the submarine at half-hour intervals for 24 hours. Our goal is to identify the central frequency signature of the submarine, create a filter using this signature, and ultimately determine the submarine location over time.

The main issue we face trying to identify the frequency signature of the submarine is that there is noise in the acoustic signal. Noise is caused by unpredictable variations in a signal between measurements that lead to random error in the measured values. Noise can be caused by many different sources such as vibration, static electricity, and external radiation signatures [1]. The signal in this problem contains mean zero white noise, which is random noise with equal power over the range of frequencies. Using the Discrete Fourier Transform (DFT) is one way to help address noise by the input signal to the frequency domain and removing low power frequencies or frequency ranges.

2. Theoretical Background

2.1. Fourier Series & Fourier Transform

The basic principle behind the Fourier Transform is that waveform signals in the real world can be always be represented as a sum of sinusoidal functions of differing frequencies [2]. This sum of signals is known as the Fourier Series for a function (or vector of functions) f operating on vector $\mathbf{x}$ of length n, given as

$$f(\mathbf{x}) = \sum_{-\infty}^{\infty} C_{\mathbf{k}} \exp(\mathrm{i}\mathbf{k}^T\mathbf{x}) \tag{1}$$

with coefficients $C_{\mathbf{k}}$ as

$$C_{\mathbf{k}} = (2\pi)^{-n} \int_0^{2\pi} f(\mathbf{x}) \exp(-\mathrm{i}\mathbf{k}^T\mathbf{x})\, d\mathbf{x} \tag{2}$$

Date: January 27, 2022.

However, we observe these equations are defined on an infinite domain, which is not reflective of real life signals. In practice, (and in solving this problem) a finite series of Fourier coefficients are calculated that measure the contributions of specific frequency signatures to represent an input signal on a compact, periodic interval $[0, 2L]$ or $[-L, L]$. These inputs can be a multidimensional function f as well as a discrete set of data points that make up a signal [3]. This is known as the Discrete Fourier Transform [4], with the forward transformation returning coefficients defined as

$$F(\mathbf{k}) = \sum_{n=0}^{N-1} f(\mathbf{x}) \exp(-2\pi \frac{\mathbf{i}\mathbf{k}n}{N}) \tag{3}$$

and the inverse transformation returning the inputs as

$$f(\mathbf{x}) = \frac{1}{N^n} \sum_{n=0}^{N-1} F(\mathbf{k}) \exp(2\pi \frac{\mathbf{i}\mathbf{k}n}{N}) \tag{4}$$

where N represents the number of discretation points in $\mathbf{x}$. In equations 3 and 4 we assume that the number of discretation points N is the same for all dimensions in $\mathbf{x}$ as this is the case in our problem. However, it should be noted that this is not required. The $\mathbf{k}$ values are the frequency wavenumbers of the signal given as $\frac{2\pi n}{2L}$ in each dimension, with n sequenced from $\frac{-N}{2}$ to $\frac{N}{2} - 1$ such that $\mathbf{k}$ is periodic on the closed interval $[0, 2L]$ or $[-L, L]$. In the case of our three dimensional problem, we define periodic $x, y, z \in [-L, L]$ with a discretation size N of 64 points. The frequency wavevectors in each dimension are thus

$$k_x, k_y, k_z = \frac{\pi}{L}[\frac{-N}{2}, \ldots, \frac{N}{2} - 1] \tag{5}$$

The Fast Fourier Transform (FFT) mentioned in this report is an algorithm that computes the DFT of a signal extremely well, with a computational efficiency of $O(N \log(N))$. One important note regarding the FFT algorithm is that the methodology requires N to be a power of two, but zero padding can be added to the input data to meet this criteria [3].

2.2. **Noise Filtering**

Given that mean zero white noise is present over the entirety of the signal, we expect that averaging the noise over all the time steps in the data will reduce the noise. In order to perform this averaging, we can make use of the fact that adding mean zero white noise to a signal is the same as adding mean zero white noise to the Fourier coefficients of the signal. Given a signal f and mean zero white noise ζ, this fact states the following, where $\mathcal{F}$ represents the Fourier Transform

$$\mathcal{F}(f + \zeta) = \mathcal{F}(f) + \mathcal{F}(\zeta)$$

Averaging all the Fourier Transforms over K samples, we have the following expression in which the white noise terms are cancelled out due to to having a mean of zero

$$\frac{1}{K} \sum_{k=1}^{K-1} \mathcal{F}(f + \zeta) = \frac{1}{K} \sum_{k=1}^{K-1} \mathcal{F}(f)^{(k)} + \cancelto{0}{\frac{1}{K} \sum_{k=1}^{K-1} \mathcal{F}(\zeta)^{(k)}}$$

While this methodology is not perfect in our case as it holds for $K \to \infty$ and we only have 49 time points, it will help attenuate the impact of the white noise on the signal. Using the de-noised and averaged data, we can find the location in the frequency space of maximum power, which represents the central frequency of the submarine. This location will correspond to values in our k_x, k_x, and k_z wavevectors, giving us the central frequency values.

As the central frequency was obtained from the averaged signal data, a method still needs to be designed to extract the signature from each individual time step to determine the submarine location. This can be accomplished by applying a three dimensional Gaussian filter centered at the central frequency to each time step. The Gaussian filter will attenuate noise and detail for values far from the central frequency, allowing us to isolate its location [5]. In three dimensions, our Gaussian filter is defined as

$$G(x) = \exp\left(\frac{-1}{2\sigma^2}((k_x - k_{cx})^2 + (k_y - k_{cy})^2 + (k_z - k_{cz})^2)\right) \quad (6)$$

where k_{cx}, k_{cy}, and k_{cz} are our central frequency signatures and σ is the standard deviation of the distribution.

3. Algorithm Implementation and Development

All code operations were done using the python programming package, and all Fourier operations were performed using the `numpy.fft` package [6]. Algorithm 1 gives an overview of the central frequency extraction and filter creation as described in section 2.2. The spatial grids for this problem were built using a length $L = 10$ and a spatial discretation of $N = 64$ points.

Algorithm 1 Noise Removal & Filter Creation

Define L, N
Create x, y, z positional grids
Create k_x, k_y, k_z grids with structure from 5
for time step in submarine data **do**
  Reshape data to $64x64x64$ ▷ Three dimensional pressure data
  Apply Fast Fourier Transform
  Apply Fourier Shift to Data
  Store Result
end for
Take Average of all FFT results ▷ Result is $64x64x64$ matrix
Take absolute value of average and find indices of max value
Extract k_{cx}, k_{cy}, k_{cz} from k_x, k_y, k_z using indices
Create Gaussian filter defined in 6

Using the Gaussian filter created in algorithm 1, the frequency signature of the submarine can be located in the data at each time step and used to determine the position of the submarine.

Algorithm 2 Locating Submarine Position

for time step in FFT transformed data **do**
  Apply Gaussian filter to data
  Apply inverse Fourier Shift to data
  Apply inverse Fourier Transform to filtered data
  Take absolute value and find indices of max value
  Extract submarine position from x, y, z using indices
  Store Result
end for

One note we make about algorithms 1 and 2 is that in both cases we apply a Fourier Shift to the data. This is due to the default functionality of `numpy.fft`, which places positive frequencies before negative frequencies. To get around this and work with ordered frequency data, we apply a Fourier Shift to the data after taking the FFT and an inverse Fourier shift before taking the inverse FFT.

4. Computational Results

Using the first part of algorithm 1 to remove the mean zero white noise from the averaged data, we find the characteristic frequency of the submarine to be the vector $\vec{k} = [5.34, -2.20, 6.91]$. The Gaussian filter was created using these values, with a spread $\sigma = 2$. Different values of σ were tested but increasing the value beyond $\sigma = 4$ was observed to leave noise in the signal, and the effect became more prominent as σ was increased further. There was not a noticeable difference between $\sigma = 2$ and $\sigma = 4$, but the lower value was chosen for the final filter.

The filter was then applied to the three dimensional Fourier transformed submarine data at each time step (per algorithm 2) to isolate the location of the central frequency signature. The filtered data was then converted from the frequency domain back to the problem space. Figure 1 shows the absolute value of the filtered acoustic data in the x, y plane at the z location of the maximum power signal for various time steps. At each time step the frequency signature of the submarine can clearly be identified in the acoustic data, and the values at other locations are nearly nearly zero. This shows that the Gaussian filter was extremely effective at attenuating the noise in the data.

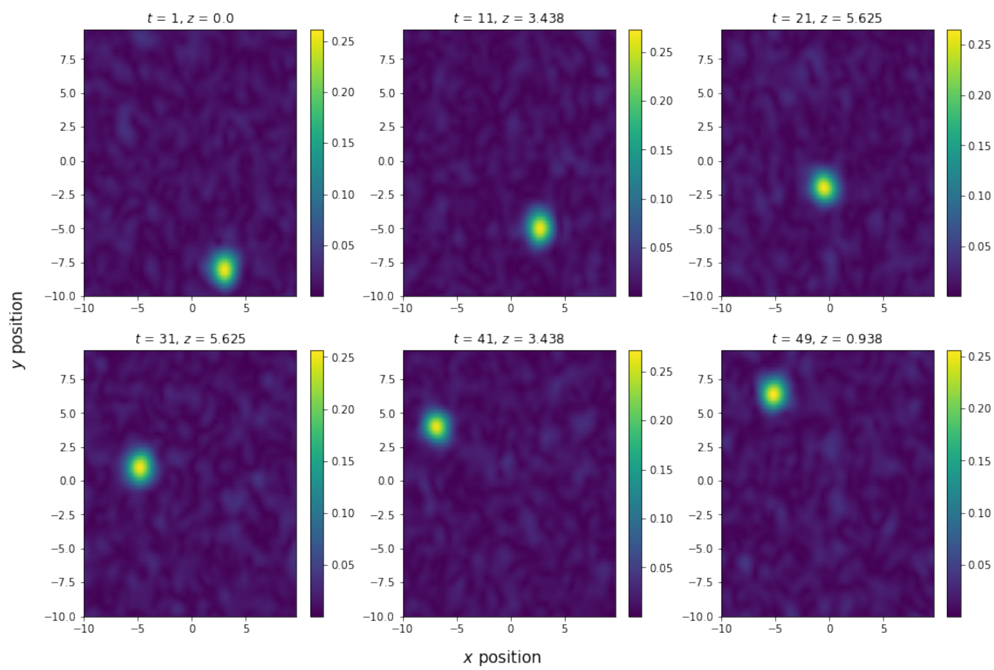
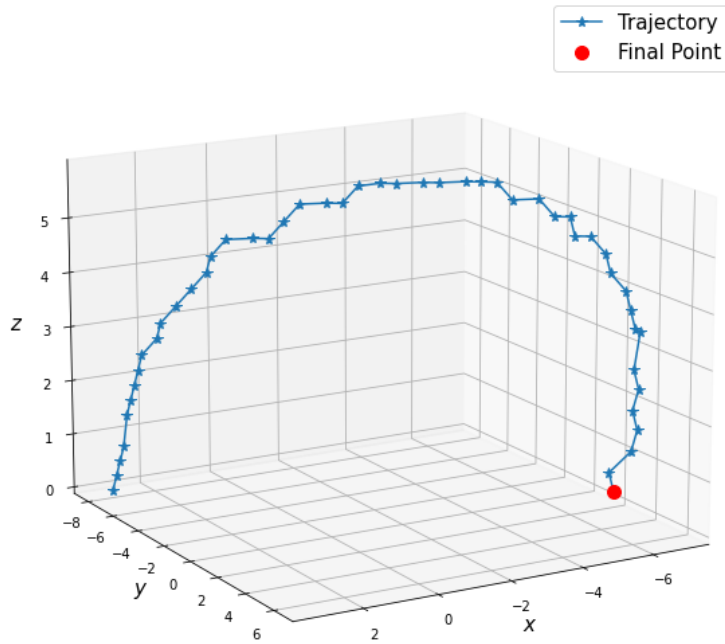


Figure 1. 2D Characteristic Frequency locations at various timesteps

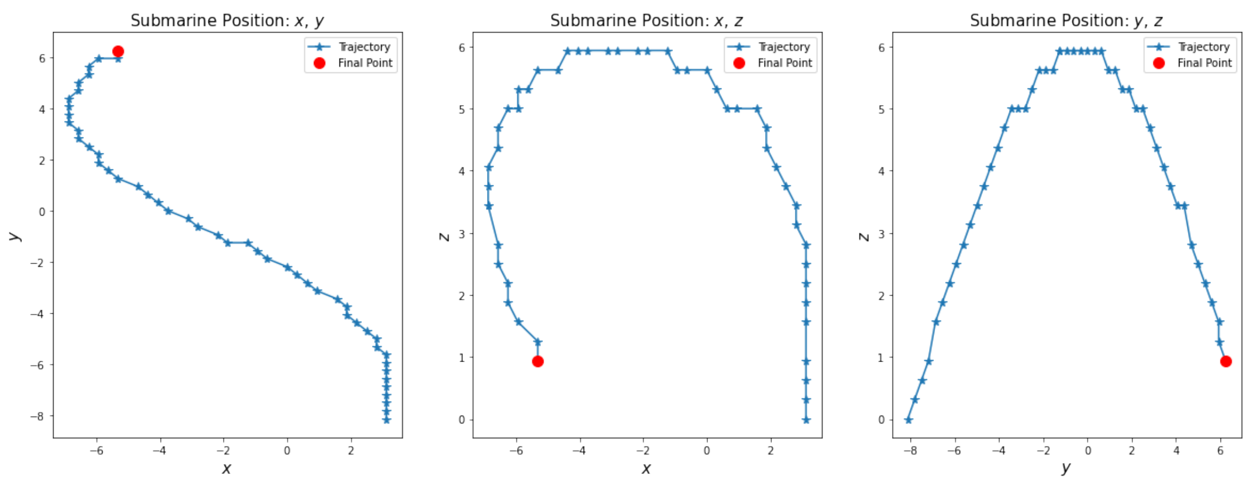
The location of the maximum values in the filtered acoustic data can be used with the x, y, and z grids defined in algorithm 1 to find position values at each time step. Table 1 shows the coordinate location of the submarine at three hour intervals. The final position of the submarine in x, y, z space was found to be $[-5.313, 6.25, 0.938]$.

time (hrs)	0	3	6	9	12	15	18	21	24
x	3.125	3.125	2.1875	0.3125	-2.1875	-4.6875	-6.5625	-6.5625	-5.3125
y	-8.125	-6.25	-4.375	-2.5	-0.9375	0.9375	2.8125	4.6875	6.25
z	0	2.1875	4.0625	5.3125	5.9375	5.625	4.6875	2.8125	0.9375

Table 1. Submarine x, y, and z Position at Three Hour Intervals



(A) Three Dimensional Submarine Trajectory



(B) Two Dimensional Planar Submarine Trajectories

FIGURE 2. Submarine Position Data over time

The full trajectory of the submarine can also be visualized over time. Figure 2a shows a three dimensional representation of the submarine position, while figure 2b shows two dimensional planar representations of its position. The combination of these trajectories could be used to deploy a submarine tracking aircraft to follow the submarine in the future.

5. Summary and Conclusions

In solving this problem, we were able to make use of the FFT algorithm to apply the Discrete Fourier Transform to real world data and obtain meaningful results. Using the FFT, we were able to remove noise from submarine acoustic data and extract a central frequency signature that was used to filter the signal and extract positional data from the results. The central frequency of the submarine was found to be $\vec{k} = [-5.34, -2.20, 6.91]$, and the submarine trajectory was visualized in figure 2. In future experiments, it would be interesting to see if the future trajectory of the submarine could be predicted using past data. However, this would likely require many more time steps and additional measurements beyond acoustic pressure to be available at each time step.

Acknowledgements

The author would like to thank Professor Bamdad Hosseini for covering the background concepts for the homework in lecture, as well as Katherine Owens for answering questions regarding reshaping of the input data in numpy. Collaboration with other classmates in Discord was also useful, and the author thanks those in the Discord chat who provided useful references and insights into potential methodologies for data visualization.

References

[1] "A pragmatic* introduction to signal processing," Jan 2022. [Online]. Available: https://terpconnect.umd.edu/~toh/spectrum/SignalsAndNoise.html
[2] P. Bevelacqua, "Introduction to the fourier transform." [Online]. Available: https://www.thefouriertransform.com/#introduction
[3] S. Brunton and J. N. Kutz, *Discrete Fourier transform (DFT) and fast Fourier transform (FFT)*.
[4] E. W. Weisstein, "Discrete fourier transform." [Online]. Available: https://mathworld.wolfram.com/DiscreteFourierTransform.html
[5] "Gaussian filtering," May 2010. [Online]. Available: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Gaussian%20Filtering_1up.pdf
[6] "Discrete fourier transform (numpy.fft)." [Online]. Available: https://numpy.org/doc/stable/reference/routines.fft.html