AMATH 482: HOME WORK 1
FINDING SUBMARINES

LANG QIN

Amazing Department, University of Washington, Seattle, WA
lq25@uw.edu

Abstract. The goal of this work is to locate a submarine in the Puget Sound using noisy acoustic data over time using Fast Fourier Transform and Gaussian Filter. This work consists of three sub-tasks in order to achieve the goal: 1) determine the frequency signature generated by the submarine; 2) denoise the acoustic data to determine the path of the submarine; and 3) determine the x, y coordinates of the submarine during the 24 hour period.

1. Introduction and Overview

A submarine equipped with new transmission technology is known to travel in the Puget Sound and its position is required. The resource available to track the submarine includes a broad spectrum recording of acoustics data obtained over 24 hours in half-hour increments. However, the frequency signature of this submarine is unknown because of the technology. Thus, this work must first determine the submarine's frequency signature before tracking down its location. In addition, denoise over the data is required because the acoustic ambient noise beneath the water makes the acoustics data noisy [5].

Following the above information, this work can be divided into three tasks:

(1) Through averaging of the Fourier transform and visual inspection, determine the frequency signature (center frequency) generated by the submarine.
(2) Design and implement a Filter to extract this frequency signature in order to denoise the data and determine the path of the submarine.
(3) Determine and plot the x, y coordinates of the submarine during the 24 hour period.

2. Theoretical Background

To achieve the goal, this work primarily relies on the following three points: 1) the Fourier Transform and Fast Fourier Transform (FFT), 2) Gaussian Filter, and 3) Time Averaging.

2.1. **The Fourier Transform and FFT.** We often have noisy signals/image/data taht needs to be cleared up or even compressed. On approach to this which is essentially the focus of the field of signal processing is to approximate the signal (data/image/etc.) as the sum of simpler components, i.e.

$$f(x) = \sum_{j=0}^{N} a_j \psi_j(x), \tag{1}$$

where $a_j \in \mathbb{R}$ coefficients and $\psi_j(x)$ are appropriate functions chosen [2].

Date: January 26, 2022.

2.1.1. *The Fourier Transform.* The Fourier Transform decomposes signal values over time into the frequency components by representing a function f as a sum of sine & cosine functions. Given $f : \mathbb{R} \to \mathbb{R}$, we define its Fourier Transform (FT) as

$$\mathcal{F}(f) \equiv \hat{f}(\xi) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) \exp(-ix\xi) du \tag{2}$$

and inverse FT for $g : \mathbb{R} \to \mathbb{C}$ as

$$\mathcal{F}^{-1}(g) \equiv \check{g}(u) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(\xi) \exp(ix\xi) d\xi \tag{3}$$

2.1.2. *Discrete Fourier Transform (DFT).* The discrete Fourier transform transforms a sequence of N complex numbers $\{\mathbf{x_n}\} := x_0, x_1, \ldots, x_{N-1}$ into another sequence of complex numbers, $\{\mathbf{X_k}\} := X_0, X_1, \ldots, X_{N-1}$, which is defined by

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \\ &= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right], \end{aligned} \tag{4}$$

where the last expression follows from the first one by Euler's formula [4].

2.1.3. *Fast Fourier Transform (FFT).* The Fast Fourier Transform (FFT) and its inverse (IFFT) are algorithms that compute $\hat{f}$ for a function $f(x)$ given at a set of equidistant points [3]. The FFT is a fast algorithm for computing the DFT. To compute the DFT of an N-point sequence using equation (4) would take $O(N^2)$ multiplies and adds. The FFT algorithm computes the DFT using $O(N \log(N))$ multiplies and adds [1].

2.2. **Gaussian Filter.** Filtering is implemented in signal/image/data processing to denoise the unclear (noisy) information. This work utilized the fact:

> *It is known that adding mean zero white noise to a signal is equivalent to adding mean zero white noise to its Fourier series coefficients.*

Therefore, this work utilized a common widely-used Gaussian Filter to remove undesired frequency of data. The Gaussian Filter takes the following form.

$$\mathcal{G}(k) = \exp(-\tau(k - k_0)^2) \tag{5}$$

The Gaussian filter attenuates the frequencies away from the center frequency.

2.3. **Time Averaging.** Following the fact mentioned in previous section 2.2, the noise in the acoustics data can be modeled by a random variable with zero mean and finite standard deviation. Thus, by averaging the signals over numerous times, the noise should add up to zero. With this idea, since in total 49 time steps are sufficiently large, averaging the acoustics data over time should minimize the noise and leave the frequency signature of the submarine.

3. Algorithm Implementation and Development

The programming work of this project can be divided into four parts.

(1) Environment Preparation
(2) Frequency Signature
(3) Submarine Path
(4) x, y Coordinates

This programming part is developed on Google Colab. The `Python` libraries imported for this work include: `google.colab.drive`, `numpy`, `matplotlib.pyplot`, `plotly.graph_objects`, and `mpl_toolkits.mplot3d`.

3.1. **Environment Preparation.** Import data from `subdata.npy` and define the *Spatial* and *Frequency* domain. We use $L = 10$ as the length of spatial domain, and scale the wave numbers of k by $2\pi/L$ as the Fourier transform frequency domain because the FFT performs over a 2π periodic signal. According to the acoustics data consists of $64 \times 64 \times 64$ 3D space at 49 time steps, this work choose $N_{\text{grid}} = 64$ as the number grid points in x, y, z direction each.

3.2. **Frequency Signature.** This part computes the frequency signature of the unknown submarine by Time Averaging idea mentioned in previous section 2.3.

The algorithm for this part is as the following.

Algorithm 1: Determine the Frequency Signature

Input : Raw acoustics data of size 262144×49
Output: 3D frequency signature of the submarine kx_sig, ky_sig, kz_sig

$df\_ave \leftarrow$ empty $64 \times 64 \times 64$ 3D array `// storage for average values`
`// Sum up and average all time step data`
for $i \leftarrow 0$ **to** 49 **do**
  $curr \leftarrow$ reshaped 3D data at time i of size $64 \times 64 \times 64$
  $df\_ave \leftarrow df\_ave +$ absolute value of $curr$'s FFT
end for
$df\_ave \leftarrow df\_ave/49$ `// averaged frequency data`
`// Extract the frequency signature`
$kx\_sig, ky\_sig, kz\_sig \leftarrow$ the 3D coordinates of df_ave maximum at the frequency space

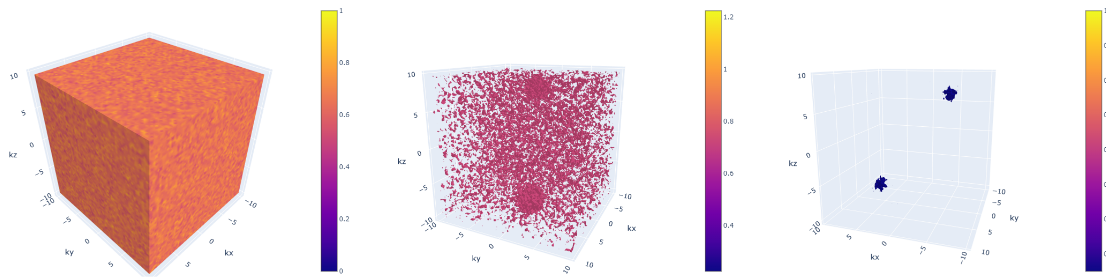
3.3. **Submarine Path.** This part denoises the raw acoustics data by Gaussian filter introduced in Section 2.2 and extracts the coordinates of the submarine over the given 24 hours.

The algorithm for this part is as the following.

Algorithm 2: Determine the Submarine's Path

Input : Raw acoustics data of size 262144×49; kx_sig, ky_sig, kz_sig
Output: List of 3D coordinates of the submarine over time

`/* Note: Python represents 3D array axes in order of x,z,y. One must flip coordinates into correct spatial coordination. */`
$df\_ave \leftarrow 0.2$ `// filter bandwidth`
`// Define the filter`
$filter \leftarrow$ 3D Gaussian filter centered at kx_sig, ky_sig, kz_sig
`// Denoise and Extract Coordinates`
$coordinates \leftarrow$ empty 49×3 2D array `// storage for coordinates at time steps`
for $i \leftarrow 0$ **to** 49 **do**
  $curr \leftarrow$ reshaped 3D data at time i of size $64 \times 64 \times 64$
  $curr\_noisy \leftarrow$ FFT of $curr$
  $curr\_clean \leftarrow$ inverse FFT of $curr\_noisy \times filter$
  $coordinates[i, :] \leftarrow$ the 3D coordinates of $curr\_clean$ maximum
end for



(A) Isovalue $\in [0, 1]$ (B) Isovalue $\in [0.6, 0.8]$ (C) Isovalue $= 0.725$

FIGURE 1. Isosurface plots of normalized averaged frequency data at corresponding isovalues.

3.4. **X,Y Coordinates.** Finally, this part extracts the x, y coordinates over time from the coordinates generated by Algorithm 2, and plots the trajectory of the submarine over time.

4. Computational Results

4.1. **Isosurface Plots of Normalized Averaged Frequency Data.** The isosurface plots of normalized averaged frequency data with various isovalues are shown as Figure 1. From Figure 1a, we observe that the isovalues of the data tend to be greater than 0.6. Figure 1b plots the isosurface of the data with isovalue at 0.725. This plot shows that the frequncy data points tend to accumulate around the the upper and lower corner. We then increase the isovalue to check if the higher isovalue frequency data share the similar property. From Figure 1c, we observe two groups of data that accumulates at the top and bottom of the space with isovalue range from 0.8 to 1. This confirms the idea that the frequency signature should be around the highest values.

4.2. **Frequency Signature of the Submarine.** The frequency signature of the submarine computed in Part 3.2 is

$$(kx\_sig, ky\_sig, kz\_sig) = (5.3407, 2.1991, -6.5973),$$

where kx_sig, ky_sig, kz_sig cooresponds to the frequency signature at $0, 1, 2$ axis of `Python` representation of 3D array.

Note: The `MATLAB` representation of the frequency signature should be $(5.3407, -6.5973, 2.1991)$.

4.3. **Submarine's Path.** The 3D path of the submarine is shown as Figure 2.

The exact position at each time step over the 24 hours are listed in Table 1.

4.4. **X,Y coordinates.** The plot of the x, y coordinates of the submarine during the 24 hour period is shown as Figure 3. The x, y coordinates can be obtained through Table 1.

5. Summary and Conclusions

Through the Fourier transform, time averaging, and the Gaussian filter, this work successfully extracts the 3D position information of the submarine from the noisy acoustics data over the 24-hour time period. The extracted 2D position data are useful for deploying a sub-tracking aircraft to keep an eye on the submarine in the future.

In addition, one may observe that the extracted path (both 3D and 2D) are not smooth. This implies that the method and algorithm implemented in this work are not perfect yet, i.e. the noise is not completely cleared and the frequency signature is not exact. The future work could be done to improve the accuracy of the submarine position.

x	y	z
3.125	0.0	-7.8125
3.125	0.0	-7.8125
3.125	0.9375	-7.5
2.8125	0.9375	-6.875
3.125	1.25	-6.5625
3.125	2.1875	-6.5625
3.125	2.1875	-6.25
2.8125	2.1875	-5.625
2.8125	3.125	-5.625
2.8125	3.125	-5.3125
2.5	3.125	-4.6875
2.5	4.0625	-4.6875
2.5	4.0625	-4.6875
1.875	4.6875	-4.0625
1.875	4.6875	-4.0625
1.25	5.3125	-3.4375
1.25	5.3125	-3.4375
0.625	5.0	-2.8125
0.3125	5.0	-2.1875
0.0	5.625	-2.1875
-0.625	5.3125	-1.875
-0.9375	5.9375	-1.875
-1.25	5.9375	-1.25
-2.1875	6.25	-0.9375
-2.1875	6.25	-0.9375
-2.8125	5.9375	-0.3125
-2.8125	5.9375	-0.3125
-3.4375	5.625	0.0
-4.375	5.9375	0.3125
-4.375	5.9375	0.3125
-4.6875	5.9375	0.9375
-5.3125	5.625	1.5625
-5.3125	5.625	1.5625
-5.9375	5.3125	1.875
-5.9375	5.3125	2.1875
-6.5625	5.0	2.5
-6.5625	5.0	2.5
-6.5625	4.375	3.4375
-6.875	4.0625	3.125
-7.1875	4.0625	3.75
-6.875	3.4375	4.0625
-6.875	3.4375	4.0625
-6.5625	2.8125	4.375
-6.875	2.8125	5.0
-6.5625	2.1875	5.3125
-5.9375	1.875	5.625
-5.9375	1.875	5.625
-5.625	1.25	5.9375
-5.0	0.9375	6.25

TABLE 1. Position coordinates of the submarine over time. *Note*: z axis represents the depth of the submarine.

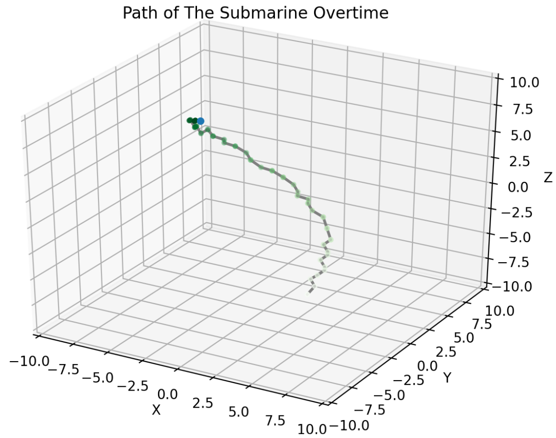


Figure 2. 3D path of the submarine over 24 hours. *Note*: z axis represents the depth of the submarine.

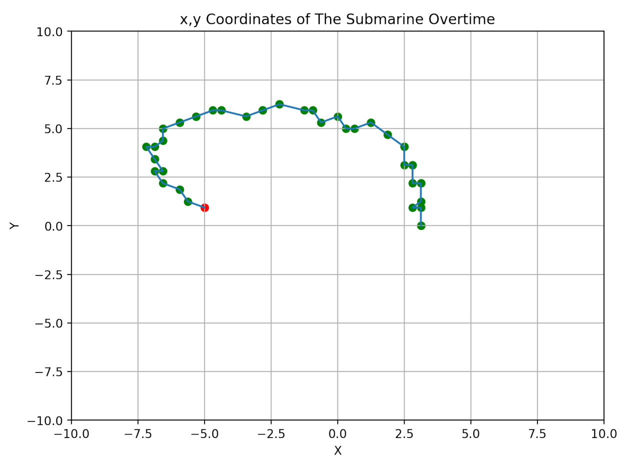


Figure 3. 2D path of the submarine over 24 hours. *Note:* The red dot is the position of the submarine at the end of the period.

Acknowledgements

The author is thankful to his girl friend Olivia Zhang and their pet cat Chonky for the emotional support overtime.

References

[1] P. Heckbert. Fourier transforms and the fast fourier transform (fft) algorithm, Jan 1998.
[2] B. Hosseini. Lecture 2 - introduction to fourier analysis. *University of Washington Computational Methods for Data Analysis*, Lecture Note, W 2022.
[3] B. Hosseini. Lecture 3 - signal processing with dft. *University of Washington Computational Methods for Data Analysis*, Lecture Note, W 2022.
[4] S. W. Smith. *The scientist and engineer's Guide to Digital Signal Processing*. California Technical Pub., 1997.
[5] G. M. Wenz. Acoustic ambient noise in the ocean: Spectra and sources. *The Journal of the Acoustical Society of America*, 34(12):1936–1956, 1962.